

2023-02-23

WiFi-Based Human Activity Recognition Using Attention-Based BiLSTM

Amany Elkelany

TU Dublin, d21126975@mytudublin.ie

Robert Ross

TU Dublin, robert.ross@tudublin.ie

Susan McKeever

TU Dublin, susan.mckeever@tudublin.ie

Follow this and additional works at: <https://arrow.tudublin.ie/scschcomcon>

 Part of the [Artificial Intelligence and Robotics Commons](#)

Recommended Citation

Elkelany, A., Ross, R., McKeever, S. (2023). WiFi-Based Human Activity Recognition Using Attention-Based BiLSTM. In: Longo, L., O'Reilly, R. (eds) Artificial Intelligence and Cognitive Science. AICS 2022. Communications in Computer and Information Science, vol 1662. Springer, Cham. DOI: 10.1007/978-3-031-26438-2_10

This Conference Paper is brought to you for free and open access by the School of Computer Sciences at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, gerard.connolly@tudublin.ie.



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 4.0 License](#)
Funder: ADAPT Research Centre

Luca Longo
Ruairi O'Reilly (Eds.)


Communications in Computer and Information Science


1662


Artificial Intelligence and Cognitive Science

30th Irish Conference, AICS 2022
Munster, Ireland, December 8–9, 2022
Revised Selected Papers

Editorial Board Members

Joaquim Filipe , *Polytechnic Institute of Setúbal, Setúbal, Portugal*

Ashish Ghosh , *Indian Statistical Institute, Kolkata, India*

Raquel Oliveira Prates , *Federal University of Minas Gerais (UFMG),
Belo Horizonte, Brazil*

Lizhu Zhou, *Tsinghua University, Beijing, China*

Rationale

The CCIS series is devoted to the publication of proceedings of computer science conferences. Its aim is to efficiently disseminate original research results in informatics in printed and electronic form. While the focus is on publication of peer-reviewed full papers presenting mature work, inclusion of reviewed short papers reporting on work in progress is welcome, too. Besides globally relevant meetings with internationally representative program committees guaranteeing a strict peer-reviewing and paper selection process, conferences run by societies or of high regional or national relevance are also considered for publication.

Topics

The topical scope of CCIS spans the entire spectrum of informatics ranging from foundational topics in the theory of computing to information and communications science and technology and a broad variety of interdisciplinary application fields.

Information for Volume Editors and Authors

Publication in CCIS is free of charge. No royalties are paid, however, we offer registered conference participants temporary free access to the online version of the conference proceedings on SpringerLink (<http://link.springer.com>) by means of an http referrer from the conference website and/or a number of complimentary printed copies, as specified in the official acceptance email of the event.

CCIS proceedings can be published in time for distribution at conferences or as post-proceedings, and delivered in the form of printed books and/or electronically as USBs and/or e-content licenses for accessing proceedings at SpringerLink. Furthermore, CCIS proceedings are included in the CCIS electronic book series hosted in the SpringerLink digital library at <http://link.springer.com/bookseries/7899>. Conferences publishing in CCIS are allowed to use Online Conference Service (OCS) for managing the whole proceedings lifecycle (from submission and reviewing to preparing for publication) free of charge.

Publication process

The language of publication is exclusively English. Authors publishing in CCIS have to sign the Springer CCIS copyright transfer form, however, they are free to use their material published in CCIS for substantially changed, more elaborate subsequent publications elsewhere. For the preparation of the camera-ready papers/files, authors have to strictly adhere to the Springer CCIS Authors' Instructions and are strongly encouraged to use the CCIS LaTeX style files or templates.

Abstracting/Indexing

CCIS is abstracted/indexed in DBLP, Google Scholar, EI-Compendex, Mathematical Reviews, SCImago, Scopus. CCIS volumes are also submitted for the inclusion in ISI Proceedings.

How to start

To start the evaluation of your proposal for inclusion in the CCIS series, please send an e-mail to ccis@springer.com.

Luca Longo · Ruairi O'Reilly
Editors

Artificial Intelligence and Cognitive Science

30th Irish Conference, AICS 2022
Munster, Ireland, December 8–9, 2022
Revised Selected Papers

Editors

Luca Longo 
Technological University Dublin
Dublin, Ireland

Ruairi O'Reilly 
Munster Technological University
Cork, Ireland



ISSN 1865-0929

ISSN 1865-0937 (electronic)

Communications in Computer and Information Science

ISBN 978-3-031-26437-5

ISBN 978-3-031-26438-2 (eBook)

<https://doi.org/10.1007/978-3-031-26438-2>

© The Editor(s) (if applicable) and The Author(s) 2023. This book is an open access publication.

Open Access This book is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this book are included in the book's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

It is a great privilege to present the proceedings of the 30th Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2022). This book is a collection of the best contributions received in AICS 2022. With regular conferences dating back to 1988, the AICS conference is Ireland's premier forum for researchers active in the fields of Artificial Intelligence and Cognitive Science. AICS provides researchers in our community the opportunity to present their exciting advances in data analytics, information retrieval, machine learning, knowledge representation and extraction, logic and reasoning, computer vision and natural language processing.

This book presents recent developments in the context of theoretical models of Artificial Intelligence and practical, intelligent applications. From the content of these research contributions, it is evident that artificial intelligence is at the forefront of society today, with many novel theoretical contributions and practical applications.

AICS 2022 received a record of 102 articles from researchers, academics and doctoral scholars from a large number of Irish universities, national companies and international institutions. This book comprises a selection of the best 41 articles presented at the conference, selected through a strict, single-blind peer-review process. Each article received at least three reviews from scholars in academia and industry. Each reviewer held a PhD in Computer Science, Cognitive Science or a relevant cognate discipline. The general chairs of the conference performed the role of programme committee chairs, and carefully selected the top contributions by ranking articles across several criteria and evaluating the qualitative feedback given by the reviewers.

The event was supported by an organising committee consisting of members of Munster Technological University, Technological University Dublin and University College Cork. The event was hosted by the Department of Computer Science of Munster Technological University (MTU) in Cork, Ireland, on December 8–9, 2022.

We thank everyone who helped in the organising committee for the 30th Irish Conference of Artificial Intelligence and Cognitive Science (AICS 2022). A special thankyou goes to the local chair, Alison O'Shea, and the publicity chairs, Begüm Genç and Andrea Balogh. An appreciation is due to the sponsors that contributed to the event's success, including Qualcomm, Keelvar, Collins Aerospace, LERO, the Science Foundation Ireland Research Centre for Software and INSIGHT, the Science Foundation Ireland Research Centre for Data Analytics. Additionally, we would like to thank the staff from MTU's research office, Department of Computer Science and Department of Mathematics for their contributions and help – without which we would not have been able to organise this great meeting.

A special thank you goes to the researchers and practitioners who submitted their work and committed to attending the event and turning it into an opportunity to meet and share findings and new avenues of research.

December 2023

Luca Longo
Ruairi O'Reilly

Organization

Organizing Committee

General Chairs, Editors and Program Committee Chairs

Ruairi O'Reilly	Munster Technological University, Ireland
Luca Longo	Technological University Dublin, Ireland

Local Chair

Alison O Shea	Munster Technological University, Ireland
---------------	---

Publicity Chairs

Begüm Genç	University College Cork, Ireland
Andrea Balogh	University College Cork, Ireland

Program Committee

Haithem Afli	Munster Technological University, Ireland
Kashif Ahmad	Munster Technological University, Ireland
Elham Alghamdi	University College Dublin, Ireland
Christian Beder	Munster Technological University, Ireland
Marija Bezbradica	Dublin City University, Ireland
Ralf Bierig	Maynooth University, Ireland
Michaela Black	Ulster University, UK
Bojan Božić	Technological University Dublin, Ireland
Rob Brennan	University College Dublin, Ireland
Derek Bridge	University College Cork, Ireland
Ken Brown	University College Cork, Ireland
Paul Buitelaar	University of Galway, Ireland
Diego Carraro	Insight Centre for Data Analytics, Ireland
Ignacio Castineiras	Munster Technological University, Ireland
Simon Caton	University College Dublin, Ireland
Darryl Charles	University of Ulster, UK
Rem Collier	University College Dublin, Ireland
Louise Connell	Maynooth University, Ireland

Fintan Costello	University College Dublin, Ireland
Katie Crowley	University of Limerick, Ireland
Fred Cummins	University College Dublin, Ireland
Padraig Cunningham	University College Dublin, Ireland
Brian Davis	ADAPT Centre, Ireland
Declan Delaney	University College Dublin, Ireland
Sarah Jane Delany	Technological University Dublin, Ireland
Deirdre Desmond	Maynooth University, Ireland
Julia Dietlmeier	Insight Centre for Data Analytics, Ireland
Pierpaolo Dondio	Technological University Dublin, Ireland
Ruihai Dong	University College Dublin, Ireland
Ryan Donovan	Munster Technological University, Ireland
Seamus Dowling	Atlantic Technological University, Ireland
Ken Duffy	Maynooth University, Ireland
Ivana Dusparic	Trinity College Dublin, Ireland
Malachy Eaton	University of Limerick, Ireland
Suzanne Egan	Mary Immaculate College, Limerick, Ireland
Ciaran Eising	University of Limerick, Ireland
Guillaume Escamocher	University College Cork, Ireland
Muftah Fraifer	University of Limerick, Ireland
John Gilligan	Technological University Dublin, Ireland
Frank Glavin	University of Galway, Ireland
Derek Greene	University College Dublin, Ireland
Josephine Griffith	University of Galway, Ireland
Diarmuid Grimes	Munster Technological University, Ireland
Eoin Grua	University of Limerick, Ireland
Lifeng Han	University of Manchester, UK
Conor Hayes	University of Galway, Ireland
Julio Noe Hernandez Torres	Trinity College Dublin, Ireland
Elizabeth Hunter	Technological University Dublin, Ireland
Georgiana Ifrim	University College Dublin, Ireland
Gareth Jones	Dublin City University, Ireland
Mark Keane	University College Dublin, Ireland
Meghana Kshirsagar	University of Limerick, Ireland
Aonghus Lawlor	University College Dublin, Ireland
Suzanne Little	Dublin City University, Ireland
Conor Lynch	Munster Technological University, Ireland
Brian Mac Namee	University College Dublin, Ireland
Tiziana Margaria	University of Limerick, Ireland
Liam Marnane	University College Cork, Ireland
John P. McCrae	University of Galway, Ireland
James McDermott	University of Galway, Ireland

Susan McKeever	Technological University Dublin, Ireland
Lucy McKenna	Trinity College Dublin, Ireland
Paul Mc Kevitt	Ulster University, Ireland
Anila Mjeda	Munster Technological University, Ireland
Brian Murphy	Munster Technological University, Ireland
Brendan Murphy	University College Dublin, Ireland
Enrique Naredo	University of Limerick, Ireland
John Nelson	University of Limerick, Ireland
Hung Ngo	Technological University Dublin, Ireland
Pilib Ó Broin	University of Galway, Ireland
Niall O'Mahony	Munster Technological University, Ireland
Mark O'Sullivan	University College Cork, Ireland
Diarmuid O'Donoghue	Maynooth University, Ireland
Colm O'Riordan	University of Galway, Ireland
Barry O'Sullivan	University College Cork, Ireland
Arjun Pakrashi	University College Dublin, Ireland
Harshvardhan J. Pandit	ADAPT Centre, Ireland
James Vincent Patten	University of Limerick, Ireland
Steve Prestwich	Insight Centre for Data Analytics, Ireland
Gregory Provan	University College Cork, Ireland
Luis Quesada	University College Cork, Ireland
Debbie Rankin	Ulster University, UK
Bujar Raufi	Technological University Dublin, Ireland
Daniel Riordan	Munster Technological University, Ireland
Lucas Rizzo	Technological University Dublin, Ireland
Eoin Rogers	Technological University Dublin, Ireland
Robert Ross	Technological University Dublin, Ireland
Ellen Rushe	University College Dublin, Ireland
Takfarinas Saber	University of Galway, Ireland
Bianca Schoen Phelan	Technological University Dublin, Ireland
Michael Schukat	University of Galway, Ireland
Ted Scully	Munster Technological University, Ireland
Alan Smeaton	Dublin City University, Ireland
Barry Smyth	University College Dublin, Ireland
Alexey Tarasov	2K Games Dublin, Ireland
Anh Duong Trinh	Technological University Dublin, Ireland
James Usher	Technological University Dublin, Ireland
Alex Vakaloudis	Munster Technological University, Ireland
Pepijn Van de Ven	University of Limerick, Ireland
Paul Walsh	Accenture, Ireland
Nic Wilson	University College Cork, Ireland
Xhemal Zenuni	SEE University in Tetovo, Macedonia

Contents

Machine Learning, Deep Learning and Applications

Inter and Intra Signal Variance in Feature Extraction and Classification of Affective State	3
<i>Zachary Dair, Samantha Dockray, and Ruairi O'Reilly</i>	
A Self-attention Guided Multi-scale Gradient GAN for Diversified X-ray Image Synthesis	18
<i>Muhammad Muneeb Saad, Mubashir Husain Rehmani, and Ruairi O'Reilly</i>	
Spot the Fake Lungs: Generating Synthetic Medical Images Using Neural Diffusion Models	32
<i>Hazrat Ali, Shafaq Murad, and Zubair Shah</i>	
Multi-Graph Convolutional Neural Network for Breast Cancer Multi-task Classification	40
<i>Mohamed Ibrahim, Shagufta Henna, and Gary Cullen</i>	
A Transformer Architecture for Online Gesture Recognition of Mathematical Expressions	55
<i>Mirco Ramo and Guénolé C. M. Silvestre</i>	
Analysis of Attention Mechanisms in Box-Embedding Systems	68
<i>Jeffrey Sardina, Callie Sardina, John D. Kelleher, and Declan O'Sullivan</i>	
A Machine Learning Approach to Industry Classification in Financial Markets	81
<i>Rian Dolphin, Barry Smyth, and Ruihai Dong</i>	
A Machine Learning Approach for Modeling and Analyzing of Driver Performance in Simulated Racing	95
<i>Fazilat Hojaji, Adam J. Toth, and Mark J. Campbell</i>	
Rapid Quantification of NaDCC for Water Purification Tablets in Commercial Production Using ATR-FTIR Spectroscopy Based on Machine Learning Techniques	106
<i>Hamzeh Asadi, Tom O'Mahony, Julie Lambert, and Kenneth N. Brown</i>	

WiFi-Based Human Activity Recognition Using Attention-Based BiLSTM 121
Amany Elkelany, Robert Ross, and Susan McKeever

A Data-Driven Analysis of Formula 1 Car Races Outcome 134
Ankur Patil, Nishtha Jain, Rahul Agrahari, Murhaf Hossari, Fabrizio Orlandi, and Soumyabrata Dev

Brain Tumor Synthetic Data Generation with Adaptive StyleGANs 147
Usama Tariq, Rizwan Qureshi, Anas Zafar, Danyal Aftab, Jia Wu, Tanvir Alam, Zubair Shah, and Hazrat Ali

Responsible and Trustworthy Artificial Intelligence

Challenges Associated with the Adoption of Artificial Intelligence in Medical Device Software 163
Karla Aniela Cepeda Zapata, Tomás Ward, Róisín Loughran, and Fergal McCaffery

An Intelligent Empowering Agent (IEA) to Provide Easily Understood and Trusted Health Information Appropriate to the User Needs 175
Marco Alfano, John Kellett, Biagio Lenzitti, and Markus Helfert

Comparison and Analysis of 3 Key AI Documents: EU’s Proposed AI Act, Assessment List for Trustworthy AI (ALTAI), and ISO/IEC 42001 AI Management System 189
Delaram Golpayegani, Harshvardhan J. Pandit, and Dave Lewis

AI and ML in School Level Computing Education: Who, What and Where? . . . 201
Joyce Mahon, Brett A. Becker, and Brian Mac Namee

Author Gender Identification Considering Gender Bias 214
Manuela Nayantara Jeyaraj and Sarah Jane Delany

Identity Term Sampling for Measuring Gender Bias in Training Data 226
Nasim Sobhani and Sarah Jane Delany

How Augmented Reality Beauty Filters Can Affect Self-perception 239
Clara Isakowitsch

Privacy-Enhanced ZKP-Inspired Framework for Balanced Federated Learning 251
Stefano Marzo, Royston Pinto, Lucy McKenna, and Rob Brennan

Automatic Vehicle Ego Body Extraction for Reducing False Detections in Automated Driving Applications	264
<i>Ciarán Hogan and Ganesh Sistu</i>	

Natural Language Processing and Recommender Systems

Recommendation Uncertainty in Implicit Feedback Recommender Systems	279
<i>Victor Coscrato and Derek Bridge</i>	
Graph-Based Diffusion Method for Top-N Recommendation	292
<i>Yifei Zhou and Conor Hayes</i>	
CouRGe: Counterfactual Reviews Generator for Sentiment Analysis	305
<i>Diego Carraro and Kenneth N. Brown</i>	
Meme Sentiment Analysis Enhanced with Multimodal Spatial Encoding and Face Embedding	318
<i>Muzhaffar Hazman, Susan McKeever, and Josephine Griffith</i>	
Intelligent Image Compression Using Traffic Scene Analysis	332
<i>David Bowden and Diarmuid Grimes</i>	
Aerial Object Detection for Water-Based Search & Rescue	344
<i>Eoghan Mulcahy, Pepijn Van de Ven, and John Nelson</i>	
Cryptocurrency Volatility Index: An Efficient Way to Predict the Future CVI ...	355
<i>An Pham Ngoc Nguyen, Martin Crane, and Marija Bezbradica</i>	
Exploring Abstractive vs. Extractive Summarisation Techniques for Sports News	368
<i>Ahmed Jouda</i>	
Automatic Linking of Podcast Segments to Topically Related Webpages	381
<i>Carla McKeon, Claudio Rocha, and Gareth J. F. Jones</i>	

Knowledge Representation, Reasoning, Optimisation and Intelligent Applications

A Large Neighborhood Search Approach for the Data Centre Machine Reassignment Problem	397
<i>Filipe Souza, Diarmuid Grimes, and Barry O'Sullivan</i>	

Bayesian Optimization with Multi-objective Acquisition Function
for Bilevel Problems 409
Vedat Dogan and Steven Prestwich

Explaining the Effects of Preprocessing on Constraint Satisfaction Search 423
Richard J. Wallace

Variable-Relationship Guided LNS for the Car Sequencing Problem 437
Filipe Souza, Diarmuid Grimes, and Barry O’Sullivan

Unimodal and Multimodal Representation Training for Relation Extraction 450
*Ciaran Cooney, Rachel Heyburn, Liam Madigan, Mairead O’Cuinn,
Chloe Thompson, and Joana Cavadas*

Run-Time Norms Synthesis in Dynamic Environments with Changing
Objectives 462
Maha Riad, Saeedeh Ghanadbashi, and Fatemeh Golpayegani

Computational Phenotyping of Decision-Making over Voice Interfaces 475
*Lili Zhang, Ruben Mukherjee, Piyush Wadhai, Willie Muehlhausen,
and Tomas Ward*

Latent Space Cartography for Geometrically Enriched Latent Spaces 488
Niall O’ Mahony, Anshul Awasthi, Joseph Walsh, and Daniel Riordan

Personalised Filter Bias with Google and DuckDuckGo: An Exploratory
Study 502
Awais Akbar, Simon Caton, and Ralf Bierig

Entity Resolution for Multiple Sources with Extended Approach 514
Phuc Pham Huu, Dongyun Nie, and Michael Scriney

Safe Lane-Changing in CAVs Using External Safety Supervisors: A Review ... 527
Lalu Prasad Lenka and Mélanie Bouroche

Author Index 539

Machine Learning, Deep Learning and Applications



Inter and Intra Signal Variance in Feature Extraction and Classification of Affective State

Zachary Dair¹(✉) , Samantha Dockray² , and Ruairi O'Reilly¹

¹ Munster Technological University, Cork, Ireland
zachary.dair@mycit.ie, ruairi.oreilly@mtu.ie

² University College Cork, Cork, Ireland
s.dockray@ucc.ie

Abstract. Psychophysiology investigates the causal relationship of physiological changes resulting from psychological states. There are significant challenges with machine learning-based momentary assessments of physiology due to varying data collection methods, physiological differences, data availability and the requirement for expertly annotated data. Advances in wearable technology have significantly increased the scale, sensitivity and accuracy of devices for recording physiological signals, enabling large-scale unobtrusive physiological data gathering. This work contributes an empirical evaluation of signal variances acquired from wearables and their associated impact on the classification of affective states by (i) assessing differences occurring in features representative of affective states extracted from electrocardiograms and photoplethysmography, (ii) investigating the disparity in feature importance between signals to determine signal-specific features, and (iii) investigating the disparity in feature importance between affective states to determine affect-specific features. Results demonstrate that the degree of feature variance between ECG and PPG in a dataset is reflected in the classification performance of that dataset. Additionally, beats-per-minute, inter-beat-interval and breathing rate are identified as common best-performing features across both signals. Finally feature variance per-affective state identifies hard-to-distinguish affective states requiring one-versus-rest or additional features to enable accurate classification.

Keywords: Machine learning · Classification · Psychophysiology · Electrocardiogram · Photoplethysmography · Affective states

1 Introduction

A significant goal of Affective Computing is to improve human-to-computer interaction by providing a system with a level of emotional intelligence that

This publication has emanated from research supported in part by a Grant from Science Foundation Ireland under Grant number 18/CRT/6222.

© The Author(s) 2023

L. Longo and R. O'Reilly (Eds.): AICS 2022, CCIS 1662, pp. 3–17, 2023.

https://doi.org/10.1007/978-3-031-26438-2_1

aids natural communications and is capable of including emotional components [27]. This has commonly been approached by deriving emotional states from speech, facial expressions, gestures and body posture analysis. However, utilising physiological signals to communicate psychological information is a recent exploration in the domain, likely due to the increased accessibility of signals from wearables.

A physiological signal represents an individual’s biological processes derived from core aspects of human biology. These signals can enable diagnostics, for instance, analysing heart rate (HR) to detect arrhythmia [29]. Psychological analysis can also be enabled as mental states originating from unconscious effort typically present a noticeable physiological change in the relevant human system [16]. The combined analysis enables a richer understanding of individuals in terms of their mental and physical health [8].

Psychological states are complex processes comprised of several components, including feelings, cognitive reactions, behaviour and thoughts [1]. Mapping psychological states to individual experience provides valuable information regarding well-being, health (physical and mental), social contexts, experiences and emotions [7].

Electrocardiograms (ECG) are physiological signals that measure the electrical activity of the heart. Typically recorded in a clinical setting using multiple electrodes attached to the individual. Photoplethysmography (PPG) is a physiological signal used to measure heart activity through variations in the blood volume of the skin, using a light-emitting-diode and photodetector. Wearable devices predominately utilise PPG to monitor heart activity. However, recently advanced wearables have included ECG capabilities for a limited number of commercial off-the-shelf (COTS) devices.

Data variances occur when recording ECG and PPG due to differing sensor placement and signal granularity [8, 22]. A lower sampling frequency is commonly used in PPG compared to ECG to reduce battery consumption in COTS devices. Such variances are under-recognised in the field of psychophysiology.

This work investigates the impact of signal variances occurring in ECG and PPG signals acquired from wearable devices for classifying affective states by addressing the following research aims: (i) To assess differences in features representative of affective states on a per signal basis, (ii) To investigate the disparity in precedence ordering of feature importance per signal, and (iii) To investigate the disparity precedence ordering of feature importance per affective state.

These aims inform the development of machine learning (ML) pipelines for classifying affective states. Utilising feature variance per signal to identify abnormal signal activity or similar affective states which are causing reduced classification accuracy. In conjunction, feature importance is utilised to provide insights into feature selection, aiding performance in tailored signal- or affect-specific approaches.

2 Related Work

2.1 Heart-Related Physiological Signals

The prevalence of heart-related data in wearable devices stems from a desire to monitor health through arrhythmia detection and HR as a measure of fitness [29]. As the heart is controlled involuntarily through the autonomic nervous system (ANS), it facilitates identifying relationships between involuntary physiological changes in heart activity and psychological states such as emotions or behaviour. Multiple psychophysiological theories aim to explain this relationship, such as Polyvagal Theory [30], which proposes that the ANS provides the neurophysiological substrates for adaptive behavioural strategies [28].

Heart activity is complex to capture. In medicine, the gold standard utilises a 12-lead ECG, resulting in comprehensive data recorded from multiple electrodes on the human body. However, in ambulatory research and daily life, this method is not feasible. Typically research-grade (RG) equipment uses several electrodes, commonly 3-lead ECG, and occasionally includes PPG as an additional measure. COTS devices tend to rely solely on PPG to monitor heart activity. However, with recent advances, top-of-the-range smart-watches (Apple Watch 4–9, Galaxy Active 2, Fitbit) include a 1-lead ECG, which is promising for portable ECG analysis [25].

Additional physiological signals such as electrodermal activity (EDA), respiration, skin temperature, electromyogram (EMG), and electrooculogram (EOG) have demonstrated potential for affective state detection [8, 20]; however, due to additional sensor requirements they are excluded from this work.

Numerous studies of affective states conduct custom data collection, providing precise control over the psychological domain explored. Varied stimuli have aided the elicitation of psychological states, for example, images, movie clips, music, and dedicated tasks to elucidate stress, such as the Trier Social Stress Test [2, 33]. As denoted in Table 1, several open-access or on-request datasets containing ECG and PPG are available. The distinct lack of emotionally labelled ECG signals from COTS devices is likely due to the recent inclusion of ECG monitoring capabilities [25].

2.2 Affective ECG Analysis

ECG signals contain noise introduced by motion artefacts, biological differences and sensor de-attachment. Signal processing techniques such as Butterworth Bandpass, Notch filters and Empirical Mode Decomposition (EMD) are utilised to reduce the signal noise levels [1]. Subsequently, features suitable for affective state classification can be extracted from the pre-processed signals.

An overview of features derivable from ECG and PPG is denoted in Table 2, grouped by extraction method. Performant ECG-based approaches typically utilise handcrafted features, particularly time-based HRV features, such as R-R intervals (RR) which are the intervals between heartbeats, successive dif-

Table 1. Datasets containing affectively labelled ECG, PPG or both

Dataset	ECG	PPG	Participants	Label
CASE [34]	✓ (1000 Hz)	✓ (1000 Hz)	30	Aro./Val.
WESAD [33]	✓ (700 Hz)	✓ (64 Hz)	15	B, S, A, M
DREAMER [15]	✓ (256 Hz)	x	23	Aro./Val./Dom.
SWELL [18]	✓ (2048 Hz)	x	25	S, Aro./Val./Dom.
DEAP [17]	x	✓ (256 Hz)	32	Aro./Val

B: Baseline, S: Stress, A: Amusement, M: Meditation Aro: Arousal, Val: Valence, Dom: Dominance

ferences (SD) and frequency-based features, such as relative, peak and absolute power of various frequency bands. Automated feature extraction is less frequently adopted, with only three of the reviewed approaches utilising deep learning or signal-processing feature extraction methods.

Recent approaches have favoured deep learning methodologies [31], achieving significant accuracies on multi-class classifications. However, older studies focusing on linear and quadratic discriminant analysis (LDA, QDA) [1, 5, 26] and support vector machines (SVM) [12] remain highly relevant, achieving high accuracy for their respective classifications. Combinations of ML classifiers forming ensembles have demonstrated potential for binary classifications in emotion detection [6]. In comparison to other studies, [31] achieved the highest accuracy for multiple emotion detection from ECG data utilising a CNN and reported setting the new state of the art for ECG emotion detection. Despite the high performance of deep learning approaches in the literature, this work focuses on classifiers using handcrafted features.

2.3 Affective PPG Analysis

PPG analysis provided by COTS devices has typically focused on tracking medical conditions, physical activity, and stress. The detrimental effects of stress on human health are a significant motivator for physiological analysis and preventative healthcare research [3]. However, instances of PPG have demonstrated similar noise levels to ECG, with the addition of skin tone and environmental light effects impacting signal quality, requiring signal cleaning techniques.

There is no consensus on the most frequently used features from the reviewed PPG-based approaches, see Table 2. The most performant approach [11] leverages handcrafted non-linear entropy features, followed by [24] using an autoencoder method for automatic feature extraction. Importantly, both handcrafted and automatically extracted features aid in achieving a high classification accuracy above 90% [10, 11, 14, 24].

Table 2. Handcrafted, automated and statistical features utilised for affective state classification. Note the divergence between features used by PPG and ECG.

	ECG								PPG								Combined	
	[31]	[1]	[13]	[5]	[26]	[12]	[6]	[32]	[11]	[24]	[10]	[14]	[21]	[19]	[32]		[4]	[9]
HRV Time:																		
HR	x	x	x	✓	x	✓	x	✓	x	x	x	✓	x	x	✓		x	x
IBI	x	x	x	x	x	x	x	✓	x	x	x	x	x	x	✓		✓	x
RR	x	x	x	✓	✓	✓	✓	✓	x	x	x	x	x	x	✓		x	✓
SD	x	x	✓	✓	✓	✓	✓	✓	x	x	x	x	x	x	✓		x	✓
P-QRS-T	x	x	✓	x	x	x	✓	✓	x	x	x	x	x	x	✓		x	x
HRV Frequency:																		
Low Freq	x	x	✓	✓	✓	✓	✓	✓	x	x	x	✓	x	x	✓		x	✓
High Freq	x	x	x	✓	✓	✓	✓	✓	x	x	x	✓	x	x	✓		x	✓
Freq. Ratios	x	x	x	✓	✓	✓	✓	✓	x	x	x	✓	x	x	✓		✓	✓
Non-Linear:																		
Poin	x	x	x	x	✓	✓	✓	x	x	x	✓	x	x	x	x		x	x
Entr	x	x	x	x	✓	✓	x	✓	✓	x	x	x	x	x	✓		✓	✓
Deep Learning:																		
Model	✓	x	x	x	x	x	x	x	x	x	x	x	✓	✓	x		x	x
AE	x	x	x	x	x	x	x	x	x	✓	x	x	x	x	x		x	x
Additional Features:																		
BR	x	x	x	x	x	✓	x	x	x	x	x	x	x	x	x		x	✓
Sig. Amp	x	x	x	x	x	x	x	✓	x	x	x	x	x	x	✓		x	✓
EMD	x	✓	x	x	x	x	✓	x	x	x	x	x	x	x	x		✓	x
Classifier	CNN	LDA	KNN	LDA	QDC	SVM ^b	Ens	SVM	PNN	SVM	SVM	DNN	CNN	CNN	SVM	SVM ^b	FNN	
No. Classes	4	2	2	3	4	2	4	2	14 ^a	4 ^a	2	5	2	2	2	2	2	
Accuracy	95%	89%	85%	85%	84%	82%	80%	69%	100%	99%	96%	91%	83%	76%	65%	93%	70%	
Datasets	[33]	Priv	Priv	Priv	Priv	Priv	Priv	Priv	[17]	[33]	[17]	Priv	[33]	[17]	Priv	Priv	Priv	

HRV: Heart Rate Variability, IBI: Inter-beat Interval, RR: R-R Intervals, SD: Successive Differences, Freq: Frequency, Poin: Poincare, Entr: Entropy, AE: Auto-Encoder, BR: Breathing Rate, Ens: Ensemble, Sig. Amp: Signal Amplitude, EMD: Empirical Mode Decomposition, LDA: Linear Discriminant Analysis, (C/D/F)NN: Convolutional/Deep/Feed-Forward Neural Network, QDC: Quadratic Discriminant Classifier, SVM: Support Vector Machine, KNN: K-Nearest Neighbours, ^aOne Vs Rest, ^bLeast-Squares SVM

These affective state classifications are conducted by variations of neural networks [11, 14, 19, 21] and SVMs [10, 24, 32], which demonstrates great potential for both binary and multi-class affective state detection using PPG solely. Notably, these approaches leverage extensive signal processing to reduce signal noise and contribute to the high performances achieved.

3 Methodology

The proposed methodology provides an approach for investigating ECG and PPG variances and the subsequent impact on affective state classification. The baseline performance of affective state classification is achieved using multiple ML classifiers per signal. The inter-signal performance variances are investigated by analysing the disparity in features between temporally aligned ECG and PPG, where the degree of feature variance is an indicator of signal quality. Inter-affective state feature variance is analysed using statistical measures to

provide insights into the distribution and similarity of affective states. Feature importance is employed to identify commonalities among the best-performing features across both signals and evaluate each feature’s utility for affect-specific approaches. Finally, a one-versus-rest (OVR) classification is adopted to improve performance when classifying similar affective states.

3.1 Datasets

For the purposes of this work, the focus was narrowed to RG physiological signals due to a lack of publicly available data for COTS devices. “The Dataset of Continuous Affect Annotations and Physical Signals for Emotion Analysis” (CASE) [34] and “The Wearable Stress and Affect Detection Dataset” (WESAD) [33], see Table 1, were utilised in this work. The datasets were selected due to their inclusion of temporally aligned ECG and PPG with psychological annotations. Additionally, these signals were recorded using RG devices in a laboratory environment. CASE incorporates Arousal and Valence annotations, achieved by collecting joystick movement resulting from emotionally stimulating video clips. WESAD focuses on stress detection with limited affective states: a baseline state elicited from “neutral reading”, amusement caused by comedic video clips, a Trier Social Stress Test [2] to provoke stress, and a meditation stage aimed at “de-exciting” the individual following the amusement and stress stages.

3.2 Pre-processing

ECG and PPG signals recorded per subject within these datasets span the duration of the experiment resulting in approx 91/40 min for WESAD/CASE. Each signal is pre-processed into 10-second windows to facilitate analysis, accomplished using a sliding window technique with a 1-second overlap. A 10-second duration was selected due to efficient performance demonstrated in [31]; additionally, this duration enables low latency as classification occurs every 10-seconds and contains adequate data for feature computation.

A Butterworth-Bandpass filter is used to reduce signal noise, facilitating the extraction of selected features while maintaining a degree of “rawness” in the signal. This filter was adopted as it is frequently adopted in the literature and more closely aligns with COTS devices and their reduced computational power.

Once filtered and windowed, the data is aligned with the psychological annotations. For WESAD, annotations were numeric values sampled 700 Hz. Each value from 0–4 is associated with the psychological states: Transient, Baseline, Stress, Amusement and Meditation. Annotations 5–7 and Transient data are omitted as per the author’s instructions [33]. Certain windows may include multiple emotive annotations; hence to identify the most pertinent emotion, the mean of all annotation values per window is calculated and rounded to the nearest annotation (1–4) using Euclidean distance. Alternative approaches [6] omit these windows and the neighbouring segments to prevent overlap.

A similar procedure is required for CASE; the raw annotation data is provided as values on an x and y-axis representing Arousal and Valence [34], these

values are normalised to a range of 0.5 to 9.5, and subsequently converted to discrete representations, resulting in low (0.5–5) and high (5.01–9.5) Arousal and Valence for each window.

Both signals provide capabilities to derive a wide array of handcrafted features useful for identifying affective information. This work utilises a python toolkit HeartPy [35] to enable extraction of HRV features from each window of data, summarised in accompanying table of Fig. 4.

3.3 HRV Feature Variance

The feature variance approach proposed is to statistically evaluate any disparity occurring in derived features from ECG and PPG under multiple conditions. Inter-signal variance is evaluated by computing the absolute difference between an ECG-derived feature and its PPG counterpart from temporally aligned signals. This is assessed using the same window of heart activity and provides a granular analysis to aid in identifying noisy, erratic or abnormal signal activity, causing unreliable computations of features. This variance is depicted by a significant absolute difference of a feature between the two signals.

Analysing the inter-affective state variance in features enables the identification of the degree of change between states, as investigated in [10,12]. The proposed methodology computes the minimum, maximum, mean and standard deviation of each feature value per affective state. Additionally outliers are identified, these are observations found in the upper and lower quartiles. This method identifies states which are complex to distinguish due to a similar feature distribution, such as meditation and relaxation. An OVR approach is adopted to convert a multi-class problem into multiple binary classifications. Using OVR, a classifier aims to identify an affective state individually from the remaining states, which increases the degree of distinction between classes.

3.4 HRV Feature Importance

This work adopts a game theory approach for feature importance known as “Shapley Additive exPlanations” [23]. This method computes SHAP values representing the degree of change on the classifier output caused by each individual feature, the magnitude of change and number of samples affected indicate the impact factor of a given feature.

Feature importance has enabled the identification of signal-specific features in [32]. However, their approach utilised different features for ECG and PPG, as such, an intra-signal comparison could not be conducted, which would provide insights into the commonality of features between ECG and PPG, motivating the intra-signal feature importance analysis provided in this work.

Feature importance can also provide insights into the variance of features per affective state, valuable for the creation of tailored emotion-specific approaches. In [9], a neural network is used for classification, and the most important features were identified from the first layer’s weights. These features were then evaluated to identify a statistical difference between affective states.

3.5 ML Based Classification of Affective State

A range of classifiers was selected to provide a holistic view of the classification performance using different architectures and the suitability of ECG and PPG for automated affective state detection.

Each classifier conducts a per-signal classification on each dataset, where 20% of the data acts as a hold-out test set, which is unseen data used to evaluate the final classifier. To ensure generalisability, five-fold cross-validation is utilised, transforming the remaining 80% of data into “folds”, enabling a per-fold classification. Subsequently, comparing the per-fold and average performance across the five folds enables the identification of the most robust and performant classifier. Finally, the most performant classifier is trained on the entire training set and evaluated against the hold-out set to assess expected performance in real-life classifications.

4 Results and Discussion

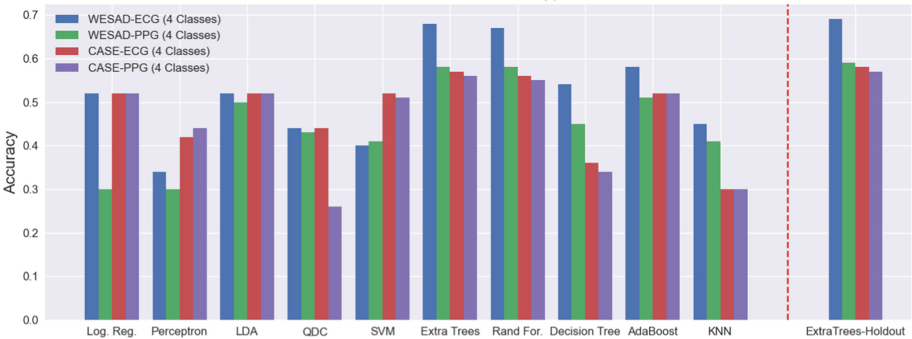


Fig. 1. Mean cross-validation accuracy classifying affective states for model selection, and the performance on the holdout test set from the best performing classifier. The performance variance in ExtraTrees classifier between ECG and PPG reflects the degree of feature variance identified per dataset.

4.1 HRV Feature Variance

The wearables’ sample rate disparity (See Table 1) is evident in the inter-signal feature variance results depicted in Fig. 2. The reduced sample rates in WESAD result in slightly decreased granularity of ECG data and significantly in PPG data compared to CASE. A higher fluctuation in feature variance occurs in WESAD in terms of magnitude and frequency, stemming from the high sample rate disparity.

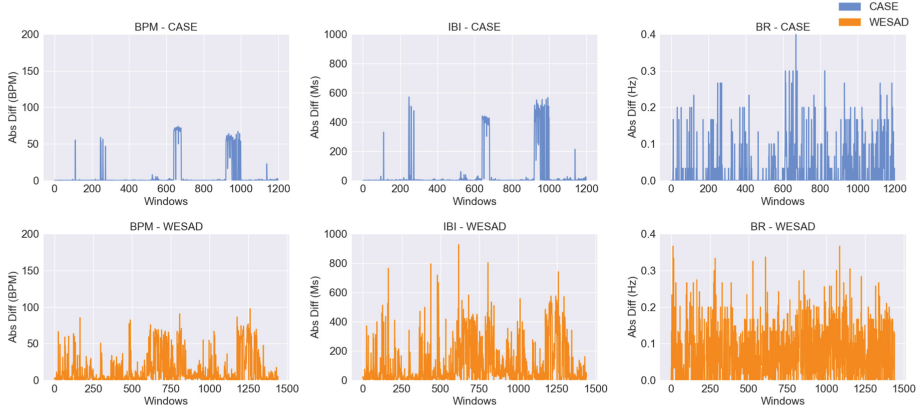


Fig. 2. Absolute difference between ECG and PPG features: BPM, IBI and BR, for CASE and WESAD. High variance demonstrates unreliable feature computation in one of the signals due to signal noise or sensor differences.

In CASE, beats per-minute (BPM) and inter-beat interval (IBI) contain a small variance with substantial spikes relative to the average. These variances occur in isolated data segments and are likely caused by electrode disconnection, movement, or subject-specific factors, visible in Fig. 2 at approximately window numbers 250, 550 and 900. Such occurrences may benefit from additional signal processing to reduce noise and improve feature computation accuracy.

Interestingly, the breathing rate (BR) feature exhibits a high deviation between signals in both datasets. This deviation indicates that at least one of the signals is unreliably computing BR, likely due to the wrist and finger placement of the PPG sensors.

A low degree of inter-affective state feature variance was identified between WESAD baseline, amusement, and meditation states for all features indicating these states are difficult to distinguish as depicted in Fig. 3. Statistically similar features negatively impact automated classification, as the classifier struggles to differentiate between the classes. This impact is demonstrated by the reduced performance in multi-class classifications (58%–69%) as compared to the OVR performance depicted by the ROC curves (ROC Area: 0.70–0.95) in Fig. 5. This performance increase validates the utility of OVR classifications when classifying affective states that are difficult to differentiate due to statistical similarities.

4.2 HRV Feature Importance

Analysing the SHAP values per feature indicates that BPM, IBI, and BR have the most significant impact on classification for both signals, as demonstrated in Fig. 4. The remaining features exhibit inconsistent influence between the signals. Most notably, standard deviation 1 divided by standard deviation 2 ($SD1/SD2$) and room-mean-square of successive differences (RMSSD) exhibit higher impact

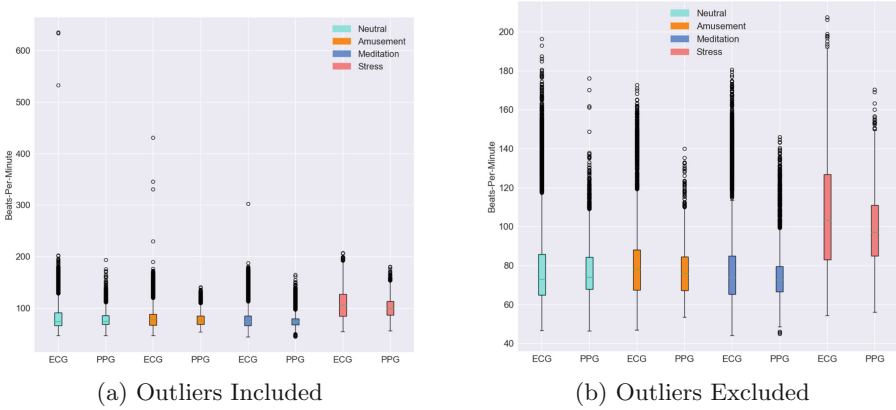


Fig. 3. Inter-signal and inter-affective state variance for BPM in WESAD, including and excluding outliers. Note in (a) the presence of outliers with a BPM of over 300 occurring in ECG indicating abnormal signal activity. Additionally, in (a, b), a visible overlap in neutral, amusement, and meditation occur, demonstrating the degree of similarity in these states.

in PPG as opposed to ECG. This demonstrates the need for assessing feature importance on a per-signal basis to identify which features are most informative for use in tailored signal-specific classification approaches.

Certain features demonstrate varying impacts across affective states, indicating the presence of affect-specific features. For example, BPM and IBI exhibit high impacts on the class “stress”, indicating their suitability for stress detection approaches. Assessing feature importance per-affective state provides an informative analysis of feature utility for affect-specific approaches.

The high feature importance of BPM for “stress” is due to statistical distinction to the other affective states in the inter-affective state feature variance, as depicted in Fig. 3. This demonstrates the benefit of assessing inter-affective state feature variance and feature importance to gain insights to aid the creation of affect-specific approaches.

4.3 Automated Affective State Classification Variance

Finally, the selected classifier is trained on the initial 80% of data and classifies the hold-out set to assess expected performance in real-life classifications. The ExtraTrees classifier (ET) was selected as the most performant classifier from the model selection, where it was trained on 80% of the training data and evaluated on the remaining 20%. Notably, ET exhibits an increased performance when evaluated on the hold-out set as it was trained on all available training data. The full model comparison and ET hold-out performance is depicted in Fig. 1. Interestingly, the classifier performance variance between ECG and PPG is similar to the degree of the inter-signal feature variance identified per dataset.

Feature	Abbrev.
Beats Per Minute	BPM
Interbeat Interval	IBI
Std dev. of RR Intervals	SDNN
Std dev. of successive diff.	SDSD
RMSE of successive diff.	RMSSD
Proportion of diff. < 20ms	pNN20
Proportion of diff. < 50ms	pNN50
Median absolute dev. of RR intervals	MAD
Estimated breathing rate	BR
Poincare analysis	SD1, SD2, S, SD1/SD2

Dev: Deviation, Diff: Difference

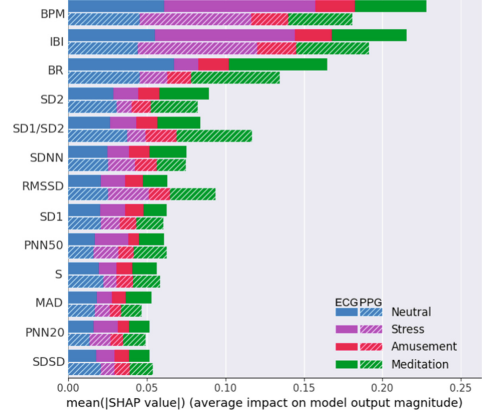


Fig. 4. SHAP Additive exPlanations (SHAP) Feature Importance from ExtraTrees classifying WESAD signals.

In contrast with the state-of-the-art [11,31], the performance achieved is lower for ECG and PPG; however, this work focuses on the analysis and understanding of variances between the signals for affective analysis rather than achieving high classification accuracy. Analysing the ROC curves from ET demonstrates the true and false positive rates per signal for each affective state, see Fig. 5. On average, ECG demonstrates increased capabilities for affective classification by achieving a higher ROC area than PPG, varying with a range of 0.02–0.11. The increased performance via OVR demonstrates the benefit of identifying and overcoming the effects of similar affective states to achieve greater classification performance.

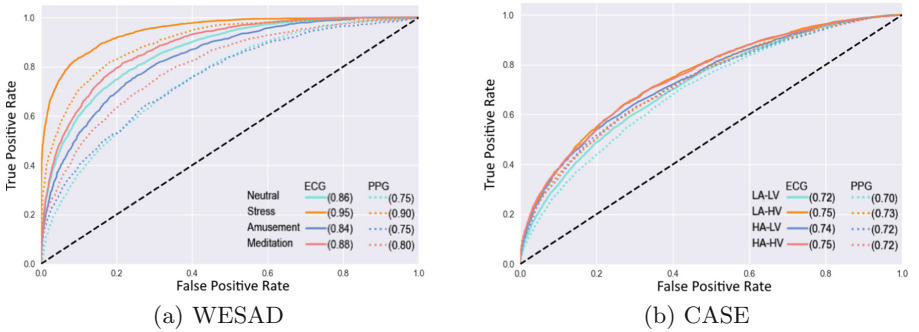


Fig. 5. ROC Curves from ExtraTrees representing the OVR classification variance between ECG and PPG

5 Conclusions

The inter-signal classification performance disparity mirrors the degree of feature variance between signals from both datasets. Specifically, WESAD exhibited a high feature variance, which explains the higher disparity in classification accuracy and ROC area per signal. Conversely, a lower inter-signal feature variance and a lower disparity in the performance measures occurred for CASE. This demonstrates the utility of inter-signal feature variance in identifying inconsistent computations of features stemming from sensor differences or abnormal signal activity, which negatively impact classification performance. These occurrences are likely to be more frequent in the ambulatory analysis due to motion artefacts and uncontrolled usage of wearables.

Furthermore, inter-affective state feature variance enables the identification of affective states that contain a similar distribution of features, which causes classification confusion. To counter this, the similar states are aggregated into an OVR classification problem, leading to increased performance, demonstrated by the ROC area per affective state.

Feature importance identifies BPM, IBI, and BR as the most impactful features for affective classification across ECG and PPG. Notably, the remaining features exhibit inconsistent impacts, specifically SD1/SD2 and RMSSD, which demonstrate a greater impact in PPG, warranting the exploration of signal-specific features. Analysing statistical measures to understand the inter-affective state feature variance indicates that certain features provide a greater degree of affect-specific information beneficial for tailored applications.

This work contributes an empirical analysis of data variances in ECG and PPG acquired using wearables and the impact on affective state classification. Therefore, enabling practitioners to make informed decisions when creating ML pipelines for affective state classification. The code-base will be made open access on Github (<https://github.com/ZacDair/Emo-Phys-Eval>), enabling automated feature variance analysis from each of these perspectives in a combined manner, regardless of data acquisition methods. While this approach analyses handcrafted features, it can also be utilised with automatically extracted features.

Future work will expand the analysis by utilising additional datasets to provide greater insights into the variances stemming from data collection devices, affective states, and population differences. In addition, an extended analysis will be conducted using additional features and methods to further inform the development of ML pipelines for affective state detection.

References

1. Agraftioti, F., et al.: ECG pattern analysis for emotion detection. *IEEE Trans. Affect. Comput.* **3**(1), 102–115 (2012). <https://doi.org/10.1109/T-AFFC.2011.28>
2. Birkett, M.A.: The trier social stress test protocol for inducing psychological stress. *J. Vis. Exp.* (2011). <https://doi.org/10.3791/3238>
3. Can, Y.S., Chalabianloo, N., Ekiz, D., Ersoy, C.: Continuous stress detection using wearable sensors in real life: algorithmic programming contest case study. *Sensors* **19**(8), 1849 (2019). <https://doi.org/10.3390/s19081849>

4. Cheema, A., Singh, M.: Psychological stress detection using phonocardiography signal: an empirical mode decomposition approach. *Biomed. Signal Process. Control* **49**, 493–505 (2019). <https://doi.org/10.1016/j.bspc.2018.12.028>
5. Cinaz, B., Arnrich, B., La Marca, R., Tröster, G.: Monitoring of mental workload levels during an everyday life office-work scenario. *Pers. Ubiquit. Comput.* **17** (2013). <https://doi.org/10.1007/s00779-011-0466-1>
6. Dissanayake, T., Rajapaksha, Y., Ragel, R., Nawinne, I.: An ensemble learning approach for electrocardiogram sensor based human emotion recognition. *Sensors* **19**(20) (2019). <https://doi.org/10.3390/s19204495>
7. Dockray, S., O'Neill, S., Jump, O.: Measuring the psychobiological correlates of daily experience in adolescents. *J. Res. Adolesc.* **29**(3), 595–612 (2019). <https://doi.org/10.1111/jora.12473>
8. Dzedzickis, A., Kaklauskas, A., Bucinskas, V.: Human emotion recognition: review of sensors and methods. *Sensors* **20**(3) (2020). <https://doi.org/10.3390/s20030592>
9. Filippini, C., et al.: Automated affective computing based on bio-signals analysis and deep learning approach. *Sensors* **22**(5) (2022). <https://doi.org/10.3390/s22051789>
10. Goshvarpour, A., Goshvarpour, A.: Poincaré's section analysis for PPG-based automatic emotion recognition. *Chaos Solitons Fractals* **114**, 400–407 (2018). <https://doi.org/10.1016/j.chaos.2018.07.035>
11. Goshvarpour, A., Goshvarpour, A.: Evaluation of novel entropy-based complex wavelet sub-bands measures of PPG in an emotion recognition system. *J. Med. Biol. Eng.* **40**(3), 451–461 (2020). <https://doi.org/10.1007/s40846-020-00526-7>
12. Hsu, Y.L., Wang, J.S., Chiang, W.C., Hung, C.H.: Automatic ECG-based emotion recognition in music listening. *IEEE Trans. Affect. Comput.* **11**(1), 85–99 (2020). <https://doi.org/10.1109/TAFFC.2017.2781732>
13. Jing, C., Liu, G., Hao, M.: The research on emotion recognition from ECG signal. In: 2009 International Conference on Information Technology and Computer Science, vol. 1, pp. 497–500 (2009). <https://doi.org/10.1109/ITCS.2009.108>
14. Kalra, P., Sharma, V.: Mental stress assessment using PPG signal a deep neural network approach. *IETE J. Res.* 1–7 (2020). <https://doi.org/10.1080/03772063.2020.1844068>
15. Katsigiannis, S., Ramzan, N.: Dreamer: a database for emotion recognition through EEG and ECG signals from wireless low-cost off-the-shelf devices. *IEEE J. Biomed. Health Inform.* (2018). <https://doi.org/10.1109/JBHI.2017.2688239>
16. Kim, J., André, E.: Emotion recognition based on physiological changes in music listening. *IEEE PAMI* **30**(12), 2067–2083 (2008). <https://doi.org/10.1109/TPAMI.2008.26>
17. Koelstra, S., et al.: Deap: a database for emotion analysis; using physiological signals. *IEEE Trans. Affect. Comput.* **3**(1), 18–31 (2012). <https://doi.org/10.1109/T-AFFC.2011.15>
18. Koldijk, S., Sappelli, M., Verberne, S., Neerincx, M.A., Kraaij, W.: The SWELL knowledge work dataset for stress and user modeling research. In: ICMI, pp. 291–298. ACM (2014). <https://doi.org/10.1145/2663204.2663257>
19. Lee, M.S., Lee, Y.K., Pae, D.S., Lim, M.T., Kim, D.W., Kang, T.K.: Fast emotion recognition based on single pulse PPG signal with convolutional neural network. *Appl. Sci.* **9**(16) (2019). <https://doi.org/10.3390/app9163355>
20. Lin, S., et al.: A review of emotion recognition using physiological signals. *Sensors* **18**(7), 2074 (2018). <https://doi.org/10.3390/s18072074>

21. Lisowska, A., Wilk, S., Peleg, M.: Catching patient's attention at the right time to help them undergo behavioural change: stress classification experiment from blood volume pulse. In: Tucker, A., Henriques Abreu, P., Cardoso, J., Pereira Rodrigues, P., Riaño, D. (eds.) AIME 2021. LNCS (LNAI), vol. 12721, pp. 72–82. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77211-6_8
22. Mahdiani, S., et al.: Is 50 Hz high enough ECG sampling frequency for accurate HRV analysis? In: EMBC, pp. 5948–5951 (2015). <https://doi.org/10.1109/EMBC.2015.7319746>
23. Molnar, C.: Interpretable Machine Learning. Lulu.com (2022). <https://christophm.github.io/interpretable-ml-book/>
24. Mukherjee, N., et al.: Real-time mental stress detection technique using neural networks towards a wearable health monitor. Meas. Sci. Technol. **33**(4), 044003 (2022). <https://doi.org/10.1088/1361-6501/ac3aae>
25. Nabeel, S., et al.: A comparison of manual electrocardiographic interval and waveform analysis in lead I of 12-lead ECG and apple watch ECG: a validation study. Cardiovasc. Digit. Health J. (2020). <https://doi.org/10.1016/j.cvdhj.2020.07.002>
26. Nardelli, M., Valenza, G., Greco, A., Lanata, A., Scilingo, P.: Recognizing emotions induced by affective sounds through heart rate variability. IEEE Trans. Affect. Comput. **6**(4), 385–394 (2015). <https://doi.org/10.1109/TAFFC.2015.2432810>
27. Picard, R.W.: Affective computing: challenges. Int. J. Hum Comput Stud. **59**(1), 55–64 (2003). [https://doi.org/10.1016/S1071-5819\(03\)00052-1](https://doi.org/10.1016/S1071-5819(03)00052-1)
28. Porges, S.W.: The polyvagal theory: new insights into adaptive reactions of the autonomic nervous system. Cleveland Clin. J. Med. **76**(Suppl. 2), S86–S90 (2009). <https://doi.org/10.3949/ccjm.76.s2.17>
29. da José, S., Luz, E., et al.: ECG-based heartbeat classification for arrhythmia detection: a survey. CMPB **127**, 144–164 (2016). <https://doi.org/10.1016/j.cmpb.2015.12.008>
30. Porges, S.W., et al.: Vagal tone and the physiological regulation of emotion. Monogr. Soc. Res. Child Dev. **59**(2–3), 167–186 (1994)
31. Sarkar, P., Etemad, A.: Self-supervised ECG representation learning for emotion recognition. IEEE Trans. Affect. Comput. (2021). <https://doi.org/10.1109/TAFFC.2020.3014842>
32. Sayed Ismail, S.N.M., Ab. Aziz, N.A., Ibrahim, S.Z.: A comparison of emotion recognition system using electrocardiogram (ECG) and photoplethysmogram (PPG). J. King Saud Univ. - Comput. Inf. Sci. **34**(6, Part B), 3539–3558 (2022). <https://doi.org/10.1016/j.jksuci.2022.04.012>
33. Schmidt, P., et al.: Introducing wesad, a multimodal dataset for wearable stress and affect detection. In: ICMI 20. ACM (2018). <https://doi.org/10.1145/3242969.3242985>
34. Sharma, K., et al.: A dataset of continuous affect annotations and physiological signals for emotion analysis (2018). <https://doi.org/10.48550/ARXIV.1812.02782>
35. van Gent, P., et al.: HeartPy: a novel heart rate algorithm for the analysis of noisy signals. Transp. Res. F: Traffic Psychol. Behav. **66**, 368–378 (2019). <https://doi.org/10.1016/j.trf.2019.09.015>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





A Self-attention Guided Multi-scale Gradient GAN for Diversified X-ray Image Synthesis

Muhammad Muneeb Saad^(✉) , Mubashir Husain Rehmani ,
and Ruairi O'Reilly

Munster Technological University Cork, Cork, Ireland
`muhammad.saad@mycit.ie`, `{mubashir.rehmani,ruairi.oreilly}@mtu.ie`

Abstract. Imbalanced image datasets are commonly available in the domain of biomedical image analysis. Biomedical images contain diversified features that are significant in predicting targeted diseases. Generative Adversarial Networks (GANs) are utilized to address the data limitation problem via the generation of synthetic images. Training challenges such as mode collapse, non-convergence, and instability degrade a GAN's performance in synthesizing diversified and high-quality images. In this work, MSG-SAGAN, an attention-guided multi-scale gradient GAN architecture is proposed to model the relationship between long-range dependencies of biomedical image features and improves the training performance using a flow of multi-scale gradients at multiple resolutions in the layers of generator and discriminator models. The intent is to reduce the impact of mode collapse and stabilize the training of GAN using an attention mechanism with multi-scale gradient learning for diversified X-ray image synthesis. Multi-scale Structural Similarity Index Measure (MS-SSIM) and Frechet Inception Distance (FID) are used to identify the occurrence of mode collapse and evaluate the diversity of synthetic images generated. The proposed architecture is compared with the multi-scale gradient GAN (MSG-GAN) to assess the diversity of generated synthetic images. Results indicate that the MSG-SAGAN outperforms MSG-GAN in synthesizing diversified images as evidenced by the MS-SSIM and FID scores.

Keywords: GANs · Self-attention · Multi-scale gradients · Mode collapse · Diversity · X-ray images · Synthesis · MS-SSIM · FID

1 Introduction

Generative adversarial networks (GANs) are generative models used for image synthesis in the computer vision domain [1]. GANs are composed of generator and discriminator models. The generator takes a random vector input and

This work is supported by the Munster Technological University's Risam Scholarship Award.

generates a noisy image. This image is passed to the discriminator model. The discriminator model classifies the generated images from the real images and provides gradient feedback to the generator. The generator model updates its learning of the feature distribution of real images through feedback provided by the discriminator. GANs work with adversarial training where the generator and the discriminator try to improve their performance based on each other's feedback [2].

GANs face difficulty in synthesizing images with complex and diverse features. This problem arises due to technical challenges that occur during the training of GANs. Training challenges include mode collapse, non-convergence, and instability [3]. Mode collapse refers to the generation of identical synthetic images by the generator regardless of diverse real images while the non-convergence and instability problem imbalanced the training due to the vanishing gradient problem. These problems limit the utility of GANs for image datasets with a diverse range of salient image features [4]. In general, GANs are designed with convolutional neural networks (CNNs) that fail to capture image features such as texture, geometry, position, and color of the objects. One of the reasons could be that the CNNs mostly utilize convolutional features in modeling the dependencies over diverse image regions [5].

In the domain of biomedical imaging, the diverse features of biomedical images are important to consider in disease recognition or computer-based diagnosis tasks [6]. These diverse features contain significant information about the disease being diagnosed and analyzed. GANs have been utilized for biomedical image synthesis. Several imaging modalities such as X-rays, Computed Tomography (CT), Magnetic Resonance (MR), Ultrasound, and Positron Emission Tomography (PET) have utilized GANs to generate synthetic samples [7]. The generation of diversified synthetic images is a significant barrier for GANs that limits their utility in the biomedical imaging domain.

X-ray images are widely utilized to diagnose diseases in the human body. X-ray images contain a wide spectrum of disease features that help physicians to monitor diseases more accurately [8]. Publicly available X-ray image datasets are limited and imbalanced [9]. Image synthesis is a potential means of augmenting and balancing these X-ray images. In image synthesis, synthetic images are produced by replicating the actual distributions of image features. Therefore, this method is significant as compared to the traditional augmentation approaches such as geometrical transformations [10]. GANs have demonstrated remarkable advancements in image synthesis in the biomedical imaging domain [11].

State-of-the-art GANs such as ProGAN [12], StyleGAN [13], and MSG-GAN [14] have been used for biomedical image synthesis. These GAN architectures have demonstrated significant performance in generating diverse images [15]. Minibatch discrimination, PixNorm, progressive growth of GAN layers, and Spectral normalization techniques have also been utilized to enhance the diversity of synthetic images. The multi-scale gradient technique enables the discriminator learning more robust for the classification of real and synthetic images [16]. Biomedical images contain salient disease features such as the location,

size, color, and structure of the disease region of interest. These features are susceptible and important to predict and analysis of the disease. GANs learn images through convolutional features without giving attention to these salient features when generating synthetic images. However, it is important for a GAN to learn these biomedical image features during the training process.

In the domain of image recognition, self-attention is considered the best approach to focusing on diverse features of the images [17]. The self-attention measures relative information of features based on their feature maps and combines them globally with a weighted scoring function. Consequently, it helps to focus on the significant features for the specific application tasks [5].

To address the training challenges of GANs, several GAN variants based on the attention mechanisms have attempted to improve the training performance of GANs for natural and biomedical images [17]. Self-attention improves the learning of generator and discriminator models in generating diversified biomedical images [18].

In order to balance and stabilize the training of a GAN, the loss function has also a great impact on the GAN's training performance for generating realistic synthetic images. Loss functions such as WGAN-GP, Hinge, and relativistic hinge losses have shown a reasonable improvement in generating diversified synthetic images [19]. However, the hinge loss has shown a great capacity to improve the GAN's learning to generate diverse biomedical images [20].

The occurrence of mode collapse and diversity of synthetic images is assessed by the Multi-scale Structural Similarity Index Measure (MS-SSIM) and Frechet Inception Distance (FID). The MS-SSIM score can detect the lack of diversity using perceptual similarity measures in synthetic images while the FID score provides a distance between the feature distributions of real and synthetic images [21].

This work contributes a novel GAN architecture for diversified X-ray image synthesis. The generator and discriminator models use multi-scale gradient learning to learn the gradient information at intermediate layers of the generator and discriminator models using multi-scale image resolutions during the training of GAN. A self-attention layer is proposed in the generator and discriminator models to learn the long-range dependencies of X-ray image features during training through a multi-scale gradient approach. The relativistic-hinge loss is used to stabilize the training and generate diverse synthetic images. The MS-SSIM and FID scores are used to evaluate the diversity of generated images.

2 Related Work

Several GAN models with modified architectures and loss functions have been proposed to improve the generation of diverse synthetic images. GAN architectures have been proposed with novel discriminators and generators based on the application domains. The performance of GANs has improved by embedding new convolutional layers, normalization, and regularization techniques in the generator and discriminator models [29–31]. Several loss functions have been proposed

Table 1. Attention mechanisms integrated into GANs for biomedical image analysis

Year	GAN_Variant	Attention_Type	Embedding	Image_Type	Application_Type
2022 [22]	MtAA-NET	Multi-task Attention	Generator	CT	Segmentation
2022 [23]	CycleGAN	Channel Attention	Generator	PET	Reconstruction
2021 [24]	AUGAN	Pixel-aware Attention	Generator	Ultrasound	Reconstruction
2021 [25]	AMGAN	Dual Attention	Generator	MRIs	Segmentation
2021 [26]	P2PGAN	Residual Attention	Generator	MRIs	Segmentation
2021 [18]	SPGGAN	Self Attention	Both	Dermoscopic	Synthesis
2021 [27]	MAGAN	Mask Attention	Both	CT	Synthesis
2020 [28]	A-CycleGAN	Self Attention	Discriminator	MR-CT	Translation

to stabilize the training of GANs [32]. These advancements demonstrate significant improvements in GANs but have a limited scope for synthesizing improved diversified and high-quality images for different application domains.

In the domain of biomedical imaging, despite of above contributions, variants of attention mechanisms are proposed in GAN architectures to enhance the capacity of GANs to generate diversified and high-quality images as detailed in Table 1. Several attention mechanisms with GANs have been proposed for different applications such as image segmentation, image reconstruction, image synthesis, and image-image translation as detailed in Table 1. The attention mechanisms embedded in the generator, discriminator, or both models can improve the diversity and quality of generated images. These GANs utilize conditional information for the segmentation and reconstruction of biomedical images using different attention mechanisms. For image synthesis, self-attention with progressively growing GAN is proposed to generate diversified dermoscopic images. The authors succeed to alleviate partial mode collapse in their GAN architecture. Similarly, a mask-attention is proposed to generate high-quality Computed Tomography (CT) images with a conditional GAN. The authors utilize additional information on attention maps of targeted diseases to improve the quality of generated images. This approach also requires additional effort for mapping the attention masks of the diseases.

Generally, conditional masks of diseases are not available publicly in the domain of biomedical imaging. It requires an additional effort from physicians to annotate the disease masks. This problem limits the scope of GANs to only annotated biomedical image datasets. However, unconditional biomedical images require more work in the context of GANs to address this limitation. Therefore, this work investigates the utility of self-attention feature maps to guide a GAN using multi-scale gradient learning for synthesizing diversified biomedical images.

3 Methodology

The workflow of the proposed approach has been depicted in Fig. 1. The MSG-SAGAN generates synthetic X-ray images using multi-scale gradient learning

between the intermediate layers of the generator and discriminator models. The generator and discriminator models are developed with the convolutional and self-attention layers to enable the relationships among long-range dependencies of image features for stabilizing the training and generating diversified X-ray images. Self-attention utilizes feature attention maps to improve the learning of the generator and discriminator models as depicted in Fig. 2.

3.1 Dataset

In this work, the publicly available dataset of Corona Virus Disease (COVID-19) chest X-ray images is utilized [33]. The dataset contains 3616 X-ray images. The images were resized into 64×64 resolution. The X-ray images were preprocessed using a horizontal flipping to augment the data size.

3.2 GAN Architecture

The Multi-scale Gradient Self-attention GAN (MSG-SAGAN) architecture utilizes a multi-scale gradient [16] learning approach between the generator and discriminator models. In MSG-SAGAN, the discriminator analyzes the output of the intermediate layers of the generator instead of looking only at the final layer output. The discriminator sends gradient feedback to multiple scales of the generator that helps a generator to create realistic diversified images. The training stabilizing techniques such as PixNorm and Mini-batch standard deviation are implemented within the GAN architecture. The PixNorm is embedded in the generator model to normalize the feature vectors. The Mini-batch standard layer is embedded into the discriminator of the GAN architecture to improve the diversity of generated image samples. The MSG-SAGAN architecture is trained with 500 epochs with a batch size of 16. As a baseline, the MSG-GAN [16] is reimplemented and trained on the CelebA dataset using the same parameters such as WGAN-GP loss, RMSprop optimizer, and 0.003 learning rates for the generator and discriminator models.

Hyperparameters: The hyperparameters have a huge impact on the training performance of MSG-SAGAN architecture. The selection of efficient hyperparameters can improve the stability of GANs and their capacity to generate diversified synthetic images. In this work, the proposed MSG-SAGAN is trained with an Adam optimizer. The generator and discriminator models are fine-tuned using different learning rates such as 0.003, 0.0003, 0.0002, and 0.0001 to evaluate the MSG-SAGAN for diverse image synthesis. The equalized learning rates are used for both generator and discriminator models to balance the training of MSG-SAGAN.

Spectral Normalization: Spectral normalization is used in the generator and discriminator models of the MSG-SAGAN. It helps the MSG-SAGAN avoid

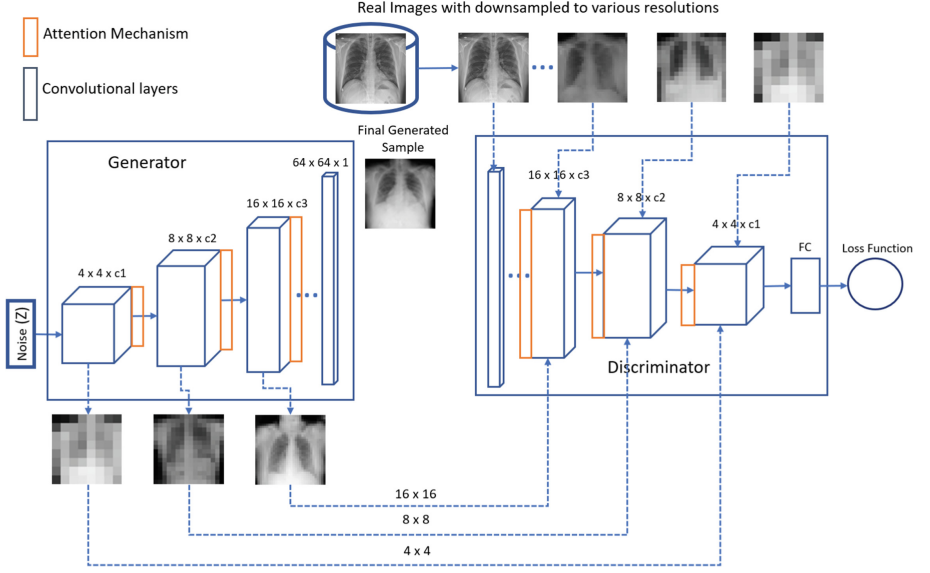


Fig. 1. The proposed architecture of MSG-SAGAN. The MSG-SAGAN is trained using multi-scale gradient learning at intermediate layers of the generator and discriminator models to generate X-ray images. The embedding of the self-attention mechanism in each block of the generator and discriminator models helps to generate improved diversified images through learning long-range dependencies of image features.

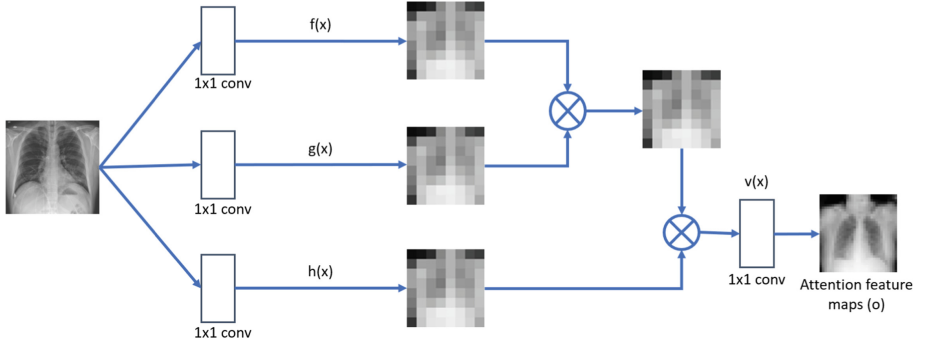


Fig. 2. Self-attention mechanism of MSG-SAGAN. The attention score is measured using different feature maps extracted from convolutional layers of the generator and discriminator models.

noisy gradients and enables fewer discriminator updates per generator, reducing the computational cost of training and improving the diversity of synthetic images.

Loss Function: The experiments were conducted using a relativistic-hinge loss function as defined in Eq. 1 and 2. Relativism in the hinge loss helps the discriminator to improve its learning by making predictions of the real images as half of the images are fake on average instead of taking them all as real. This prior training information helps the discriminator to classify and predict the real and fake images more accurately [19].

$$L_D^{\text{HingeGAN}} = \mathbb{E}_{x_r \sim \mathbb{P}} \left[\max \left(0, 1 - \tilde{D}(x_r) \right) \right] + \mathbb{E}_{x_g \sim \mathbb{Q}} \left[\max \left(0, 1 + \tilde{D}(x_g) \right) \right] \quad (1)$$

$$L_G^{\text{HingeGAN}} = \mathbb{E}_{x_g \sim \mathbb{P}} \left[\max \left(0, 1 - \tilde{D}(x_g) \right) \right] + \mathbb{E}_{x_r \sim \mathbb{Q}} \left[\max \left(0, 1 + \tilde{D}(x_r) \right) \right] \quad (2)$$

$$\begin{aligned} \tilde{D}(x_r) &= C(x_r) - \mathbb{E}_{x_g \sim \mathbb{Q}} C(x_g) \\ \tilde{D}(x_g) &= C(x_g) - \mathbb{E}_{x_r \sim \mathbb{P}} C(x_r) \end{aligned}$$

In Eq. 1 and 2 as reported in [19], discriminator and generator losses are defined for real and generated images. The real image samples are defined with x_r and the generated samples are defined with x_g where P and Q refer to the distributions of real and generated data respectively. The non-transformed layer is denoted by $C(x)$ while $D(x)$ denotes the transformed layer.

Self-attention Mechanism: The self-attention is embedded in the generator and discriminator models of the MSG-SAGAN. The self-attention has a significant capacity for modeling relationships between diverse features in images. These diverse features include different spatial regions, channels, and pixels of images [17]. The self-attention utilizes two feature spaces f and g transformed by previous hidden layer $x \in \mathbb{R}^{C \times N}$ to calculate the attention [5] shown in Fig. 2. The attention function is calculated using the following equation where feature spaces f and g are $f(x) = W_f x, g(x) = W_g x$:

$$\beta_{j,i} = \frac{\exp(s_{ij})}{\sum_{i=1}^N \exp(s_{ij})}, \text{ where } s_{ij} = f(x_i)^T g(x_j) \quad (3)$$

In Eq. 3, $\beta_{j,i}$ indicates the range of attention where the model computes mapping of j^{th} location of the j^{th} feature regions. Moreover, C denotes the number of channels while N denotes the number of feature locations of features transformed by the prior hidden layer. The output of the overall attention layer is formulated [5] as follows:

$$o_j = v \left(\sum_{i=1}^N \beta_{j,i} h(x_i) \right), h(x_i) = W_h x_i, v(x_i) = W_v x_i \quad (4)$$

In Eq. 4, the output o is unrolled as $o = (o_1, o_2, \dots, o_j, \dots, o_N) \in \mathbb{R}^{C \times N}$ while $W_g \in \mathbb{R}^{C \times C}, W_f \in \mathbb{R}^{C \times C}, W_h \in \mathbb{R}^{C \times C}$, and $W_v \in \mathbb{R}^{C \times C}$ are learned weight metrics. These weight metrics are implemented as 1×1 convolutions within

the attention mechanism. The channel count is reduced as c/k to improve the memory efficiency where k is set to 8 as suggested in [5].

Furthermore, the output of the attention layer is multiplied by a scale parameter and appended back to the input feature map [5]. So, the final output of the self-attention layer will be:

$$y_i = \gamma o_i + x_i \quad (5)$$

In Eq. 5, γ is a learnable scale parameter that is initialized at zero.

3.3 Identification of Mode Collapse Problem

The occurrence of mode collapse is identified by the MS-SSIM. The MS-SSIM computes the similarity score between two images using contrast, structure, and luminance features. MS-SSIM score is measured using randomly selected image pairs from the dataset to assess the diversity of synthetic images. The diversity of images is compared by measuring the MS-SSIM score from the real dataset and synthetic image dataset generated by GANs. A higher MS-SSIM score of the synthetic dataset indicates the occurrence of mode collapse in GANs. MS-SSIM can be computed between two image samples a and b as defined in Eq. 6 [34].

$$\text{MS-SSIM}(a, b) = I_M(a, b)^{\alpha_M} \prod_{j=1}^M C_j(a, b)^{\beta_j} S_j(a, b)^{\gamma_j} \quad (6)$$

Contrast (C) and structural (S) features of images are computed at scale j as denoted in Eq. 6. Luminance (I) is calculated at the coarsest scale (M). The α , β , and γ are the weight parameters as detailed in [35]. In this work, 3616 real and 3616 synthetic X-ray images are used to compute the MS-SSIM scores of real and synthetic image datasets.

3.4 Evaluation of the Diversity and Quality of Synthetic X-ray Images

The diversity and quality of generated images are evaluated using the FID scores. FID computes the Wasserstein-2 distance between synthetic images and real images using feature activations [36]. It captures the multivariate Gaussian activations by calculating the mean and covariance of the images (real and synthetic) using the last pooling layer of an Inception-V3 model. The FID score is calculated as shown in Eq. 7, [34].

$$\text{FID}(r, s) = \|\mu_r - \mu_s\|_2^2 + \text{Tr} \left(\Sigma_r + \Sigma_s - 2(\Sigma_r \Sigma_s)^{\frac{1}{2}} \right) \quad (7)$$

In Eq. 7, r and s denote real and synthetic images while (μ_r, Σ_r) and (μ_s, Σ_s) denote the mean and covariances of real and synthetic images. The FID score ranges from 0.0 to $+\infty$. The higher FID score shows a larger distance between synthetic and real data distributions that indicates the occurrence of mode collapse [34]. A lower FID score shows a smaller distance between synthetic and real data distributions that indicates a higher degree of diversity. This work measures FID using 3616 real and 3616 generated images.

Table 2. Analysis of the MS-SSIM and FID scores for the proposed MSG-SAGAN architecture and the MSG-GAN architecture to evaluate the diversity of generated synthetic X-ray images. Best scores are highlighted in bold values.

GANs	PN	SN	MBD	AM	FA	Opt	LR	Loss	Data	FID	MR	MG
MSG-GAN [16]	✓	x	✓	x	✓	RMSprop	0.003	WGAN-GP	CelebA	8.86	–	–
MSG-GAN ₁ (Re)	✓	x	✓	x	✓	RMSprop	0.003	WGAN-GP	CelebA	18.5	–	–
MSG-GAN ₂	✓	x	✓	x	✓	RMSprop	0.003	WGAN-GP	X-ray	380	0.50	0.74
MSG-GAN ₃	✓	x	✓	x	x	RMSprop	0.003	WGAN-GP	X-ray	310.2	0.50	0.51
MSG-GAN ₄	✓	x	✓	x	✓	Adam	0.003	WGAN-GP	X-ray	330.33	0.50	0.66
MSG-GAN ₅	✓	x	✓	x	✓	RMSprop	0.003	RLHinge	X-ray	200	0.50	0.50
MSG-GAN ₆	✓	x	✓	x	✓	Adam	0.003	RLHinge	X-ray	167.1	0.50	0.47
MSG-GAN ₇	✓	x	✓	x	x	Adam	0.003	RLHinge	X-ray	194.85	0.50	0.51
MSG-SAGAN ₁	✓	x	✓	✓	x	Adam	0.003	RLHinge	X-ray	557.28	0.50	0.99
MSG-GAN ₈	✓	x	✓	x	x	Adam	0.0003	RLHinge	X-ray	217.3	0.50	0.54
MSG-GAN ₉	✓	x	✓	x	x	Adam	0.0002	RLHinge	X-ray	272.02	0.50	0.53
MSG-GAN ₁₀	✓	x	✓	x	x	Adam	0.0001	RLHinge	X-ray	254.0	0.50	0.58
MSG-GAN ₁₁	✓	✓	✓	x	x	Adam	0.003	RLHinge	X-ray	413.8	0.50	1.0
MSG-SAGAN ₂	✓	✓	✓	✓	x	Adam	0.003	RLHinge	X-ray	413.8	0.50	1.0
MSG-SAGAN ₃	✓	✓	✓	✓	x	Adam	0.0003	RLHinge	X-ray	387.2	0.50	0.55
MSG-SAGAN ₄	✓	✓	✓	✓	x	Adam	0.0002	RLHinge	X-ray	198.8	0.50	0.55
MSG-SAGAN ₅	✓	✓	✓	✓	x	Adam	0.0001	RLHinge	X-ray	282.2	0.50	0.54
MSG-SAGAN ₆	✓	✓	✓	✓	✓	Adam	0.0003	RLHinge	X-ray	243.0	0.50	0.47
MSG-SAGAN ₇	✓	✓	✓	✓	✓	Adam	0.0002	RLHinge	X-ray	366.2	0.50	0.53
MSG-SAGAN ₈	✓	✓	✓	✓	✓	Adam	0.0001	RLHinge	X-ray	139.6	0.50	0.50

Re: Reimplemented; PN: Pixel Norm; SN: Spectral Norm; MBD: Minibatch Std Dev
AM: Attention Mechanism; FA: Flip Augment; Opt: Optimizer; LR: Learning Rate
MR: MS-SSIM Real; MG: MS-SSIM Generated; RLHinge: Relativistic Hinge

4 Results and Discussion

The MSG-SAGAN is proposed to alleviate the mode collapse in the MSG-GAN and improve the diversity of generated synthetic images in the context of X-ray images. MSG-SAGAN is a variant of MSG-GAN that utilizes an attention mechanism with multi-scale gradient learning to enhance the efficacy of synthesizing improved diversified X-ray images. The MS-SSIM score is used to identify the occurrence of mode collapse while the FID scores are used for the evaluation of the diversity in synthetic images. Resultant MS-SSIM and FID scores of MSG-GAN and MSG-SAGAN architectures are compared under a range of parameter settings as denoted in Table 2.

The reimplementation of the MSG-GAN as detailed in [16] resulted in a higher FID score than the original work when evaluated against the CelebA dataset. This was likely due to the number of real and synthetic images used in the calculation of FID. These details are omitted from [16] while in this work 10,000 real and 10,000 synthetic images were used in calculating the FID.

In the context of diverse synthetic X-ray images, the MSG-GAN₂ is trained using the same parameter settings including the loss, optimizer, learning rate,

and horizontal flipping data augmentation. MSG-GAN underperformed in synthesizing diversified X-ray images as indicated by the degraded MS-SSIM and FID scores.

The WGAN-GP loss is used to stabilize the training of GANs by avoiding the vanishing gradient problem. However, the RMSprop optimizer does not converge the training using the WGAN-GP loss for X-ray images because the RMSprop only relies on the second-order moment of gradients which leads to unstable training. Therefore, this parameter setting of MSG-GAN was not efficient to alleviate the mode collapse, stabilize the training, and generate diversified X-ray images.

The X-ray images contain salient features such as the spine, heart, and lungs with their visual signatures like ribs, aortic arch, and distinct curvature of lower lungs. All these features are important to learn by the discriminator so that it can provide constructive feedback to the generator model. So, a GAN should focus on these X-ray image features when generating synthetic images. The proposed architecture of MSG-SAGAN has the capacity to learn these X-ray features using the attention feature maps as depicted in Fig. 2.

Firstly, the effect of data augmentation is analyzed. The MSG-GAN₃ does not utilize the horizontal flipping and the results of MS-SSIM and FID are slightly improved but no significant improvement was seen as the higher MS-SSIM score of synthetic X-ray images than the MS-SSIM score of real images indicates the occurrence of mode collapse. The MSG-GAN₇ with Adam optimizer and relativistic hinge loss is trained without horizontal flipping but the results were degraded as compared to MSG-GAN₆ with flipping. Furthermore, the MSG-SAGAN₆₋₈ utilizes the horizontal flipping and alleviated mode collapse, and improved the diversity of synthetic images as compared to the MSG-SAGAN₃₋₅ that does not utilize the horizontal flipping.

Secondly, the MSG-GAN₄ is trained with an Adam optimizer and WGAN-GP loss that degrade the results. Moreover, the MSG-GAN₆₋₁₁ and MSG-SAGAN₁₋₈ are trained with the Adam optimizer and the relativistic hinge loss that alleviates the mode collapse and improves the diversity of generated images. The degraded results are evident from the other parameters such as spectral norm and attention mechanism. The Adam optimizer outperformed RMSprop due to the fact that it has the capacity to stabilize the training and converge faster because it uses both first and second-order moments of the gradients.

Thirdly, the relativistic hinge loss is used with the Adam and RMSprop optimizer in the MSG-GAN₆₋₁₁ and MSG-SAGAN₁₋₈. The relativistic hinge loss indicates significant improvement to alleviate the mode collapse and improve the diversity of synthetic images because relativism in the hinge loss helps a discriminator to provide constructive feedback to the generator.

The learning rate has a huge impact on the training of the GAN architectures. The most performant learning rate for MSG-GAN was 0.003 while 0.0001 for MSG-SAGAN. This happens because the multi-scale gradient learning stabilizes the training with a learning rate of 0.003 while the self-attention mechanism balances the training with a learning rate of 0.0001 as indicated in Table 2.

Results indicate that spectral normalization degrades the training of the MSG-GAN while improving the training of the MSG-SAGAN as indicated in Table 2. In the MSG-GAN, spectral normalization degrades the significant gradients that are flowing between the generator and the discriminator models (See MSG-GAN₁₁). Whereas, spectral normalization helps to avoid noisy gradients that are produced during the training of MSG-SAGAN due to the attention mechanism.

MSG-SAGAN₈ outperforms the MSG-GAN₆ in terms of synthesizing diversified images and stabilizing the training process. Integrating the self-attention mechanism improves the flow of multi-scale gradients between the generator and discriminator models with small learning rates while degrading with large ones. The multi-scale gradients help improve the generator’s learning capacity and discriminator models by propagating the gradients between the intermediate layers of the generator to the discriminator and vice versa. Consequently, the feature attention maps help a GAN to make relationships between long-range dependencies of the diverse image features.

The most performant MSG-GAN₆ instance results in an improved MS-SSIM of 0.474 for synthetic X-ray images as compared to real images and an FID of 167.1. However, the most performant MSG-SAGAN₈ instance results in an improved MS-SSIM of 0.50 for synthetic X-ray images as compared to real images and an improved FID of 139.6. The MS-SSIM and FID scores for MSG-SAGAN₈ indicate a stable training period and a reduction in the impact of mode collapse while synthesizing improved diversified X-ray images as compared to the alternate instances evaluated.

5 Conclusion

In this work, MSG-SAGAN was proposed to reduce the impact of mode collapse and training instability for generating synthetic X-ray images. The MSG-SAGAN demonstrated an improved capacity for the synthesis of diversified X-ray images using the attention mechanism as compared to the MSG-GAN. The MSG-SAGAN was evaluated under different settings to quantify their impact on the diversity of synthetic images generated. Results were evaluated using the MS-SSIM and FID scores. The most performant MS-SSIM (0.50) and FID (139.6) were produced by MSG-SAGAN.

The MS-SSIM and FID scores indicate that the multi-scale gradients approach in a GAN is performant with a learning rate of 0.003 for X-ray images. However, an attention mechanism with multi-scale gradient learning is the most performant with a learning rate of 0.0001. These results of MS-SSIM and FID demonstrate the impact of learning rates in the training of GANs to synthesize diversified X-ray images. A learning rate of 0.0001 utilizes small training steps to update the gradient weights for each iteration to converge the MSG-SAGAN training to balance and stabilized training.

Spectral normalization degrades the training stability of MSG-GAN while improving the training stability of MSG-SAGAN. Adam was the most performant optimizer in both MSG-GAN and MSG-SAGAN. Relativistic hinge loss

stabilizes the training and improves the generation of diversified X-ray images. The data augmentation of horizontal flipping indicates a significant improvement in stabilizing the training of MSG-SAGAN to synthesize diversified X-ray images. Horizontal flipping provides mirror copies of X-ray images that improve the learning of MSG-SAGAN with training more on salient features of X-ray images.

In future work, different variants of attention mechanisms will be investigated with a multi-scale gradient approach in the GAN architecture for synthesizing X-ray images. The self-attention will be integrated with different positions in the generator and discriminator models or only with the generator or discriminator model in the MSG-SAGAN. Different learning rates will also be investigated to synthesize the improved diversified X-ray images. This work will be extended with the integration of self-attention and its variants into state-of-the-art GANs such as StyleGAN V3, and Projected GANs.

References

1. Wang, Z., She, Q., Ward, T.E.: Generative adversarial networks in computer vision: a survey and taxonomy. *ACM Comput. Surv. (CSUR)* **54**(2), 1–38 (2021)
2. Goodfellow, I., et al.: Generative adversarial networks. *Commun. ACM* **63**(11), 139–144 (2020)
3. Jabbar, A., Li, X., Omar, B.: A survey on generative adversarial networks: variants, applications, and training. *ACM Comput. Surv. (CSUR)* **54**(8), 1–49 (2021)
4. Wu, Z., Wang, Z., Yuan, Y., Zhang, J., Wang, Z., Jin, H.: Black-box diagnosis and calibration on GAN intra-mode collapse: a pilot study. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **17**(3s), 1–18 (2021)
5. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: *International Conference on Machine Learning*, pp. 7354–7363. PMLR (2019)
6. Liu, Z., et al.: A survey on applications of deep learning in microscopy image analysis. *Comput. Biol. Med.* **134**, 104523 (2021)
7. AlAmir, M., AlGhamdi, M.: The role of generative adversarial network in medical image analysis: an in-depth survey. *ACM Comput. Surv. (CSUR)* **55**, 1–36 (2022)
8. Aggarwal, R., et al.: Diagnostic accuracy of deep learning in medical imaging: a systematic review and meta-analysis. *NPJ Digit. Med.* **4**(1), 1–23 (2021)
9. Alvarez-Rodríguez, L., de Moura, J., Novo, J., Ortega, M.: Does imbalance in chest X-ray datasets produce biased deep learning approaches for Covid-19 screening? *BMC Med. Res. Methodol.* **22**(1), 1–17 (2022)
10. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. *J. Big Data* **6**(1), 1–48 (2019)
11. Ahmad, W., Ali, H., Shah, Z., Azmat, S.: A new generative adversarial network for medical images super resolution. *Sci. Rep.* **12**(1), 1–20 (2022)
12. Kim, M., Kim, S., Kim, M., Bae, H.J., Park, J.W., Kim, N.: Realistic high-resolution lateral cephalometric radiography generated by progressive growing generative adversarial network and quality evaluations. *Sci. Rep.* **11**(1), 1–10 (2021)
13. Hong, S., et al.: 3D-StyleGAN: a style-based generative adversarial network for generative modeling of three-dimensional medical images. In: Engelhardt, S., et al. (eds.) *DGM4MICCAI/DALI 2021. LNCS*, vol. 13003, pp. 24–34. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88210-5_3

14. Molahasani Majdabadi, M., Choi, Y., Deivalakshmi, S., Ko, S.: Capsule GAN for prostate MRI super-resolution. *Multimed. Tools Appl.* **81**(3), 4119–4141 (2022)
15. Park, H.Y., et al.: Realistic high-resolution body computed tomography image synthesis by using progressive growing generative adversarial network: visual turing test. *JMIR Med. Inform.* **9**(3), e23328 (2021)
16. Karnewar, A., Wang, O.: MSG-GAN: multi-scale gradients for generative adversarial networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7799–7808 (2020)
17. Guo, M.H., et al.: Attention mechanisms in computer vision: a survey. *Comput. Vis. Media* **8**, 331–368 (2022)
18. Abdelhalim, I.S.A., Mohamed, M.F., Mahdy, Y.B.: Data augmentation for skin lesion using self-attention based progressive generative adversarial network. *Expert Syst. Appl.* **165**, 113922 (2021)
19. Jolicoeur-Martineau, A.: The relativistic discriminator: a key element missing from standard GAN. *arXiv preprint [arXiv:1807.00734](https://arxiv.org/abs/1807.00734)* (2018)
20. Kim, E., Cho, H., Ko, E., Park, H.: Generative adversarial network with local discriminator for synthesizing breast contrast-enhanced MRI. In: *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 1–4. IEEE (2021)
21. Saad, M.M., Rehmani, M.H., O'Reilly, R.: Addressing the intra-class mode collapse problem using adaptive input image normalization in GAN-based X-ray images. In: *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 2049–2052 (2022). <https://doi.org/10.1109/EMBC48229.2022.9871260>
22. Li, H., et al.: Explainable attention guided adversarial deep network for 3D radiotherapy dose distribution prediction. *Knowl.-Based Syst.* **241**, 108324 (2022)
23. Shang, C., et al.: Short-axis pet image quality improvement by attention CycleGAN using total-body pet. *J. Healthcare Eng.* **2022** (2022)
24. Tang, J., Zou, B., Li, C., Feng, S., Peng, H.: Plane-wave image reconstruction via generative adversarial network and attention mechanism. *IEEE Trans. Instrum. Meas.* **70**, 1–15 (2021)
25. Yin, J., Zhou, Z., Xu, S., Yang, R., Liu, K.: A generative adversarial network fused with dual-attention mechanism and its application in multitarget image fine segmentation. *Comput. Intell. Neurosci.* **2021** (2021)
26. Deng, H., Zhang, Y., Li, R., Hu, C., Feng, Z., Li, H.: Combining residual attention mechanisms and generative adversarial networks for hippocampus segmentation. *Tsinghua Sci. Technol.* **27**(1), 68–78 (2021)
27. Liu, Y., Meng, L., Zhong, J.: MAGAN: mask attention generative adversarial network for liver tumor CT image synthesis. *J. Healthcare Eng.* **2021** (2021)
28. Kearney, V., et al.: Attention-aware discrimination for MR-to-CT image translation using cycle-consistent generative adversarial networks. *Radiol. Artif. Intell.* **2**(2) (2020)
29. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint [arXiv:1511.06434](https://arxiv.org/abs/1511.06434)* (2015)
30. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. *arXiv preprint [arXiv:1802.05957](https://arxiv.org/abs/1802.05957)* (2018)
31. Nie, W., Patel, A.B.: Towards a better understanding and regularization of GAN training dynamics. In: *Uncertainty in Artificial Intelligence*, pp. 281–291. PMLR (2020)

32. Pan, Z., et al.: Loss functions of generative adversarial networks (GANs): opportunities and challenges. *IEEE Trans. Emerg. Top. Computat. Intell.* **4**(4), 500–522 (2020)
33. Rahman, T., et al.: Exploring the effect of image enhancement techniques on Covid-19 detection using chest X-ray images. *Comput. Biol. Med.* **132**, 104319 (2021)
34. Borji, A.: Pros and cons of GAN evaluation measures. *Comput. Vis. Image Underst.* **179**, 41–65 (2019)
35. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*, vol. 2, pp. 1398–1402. IEEE (2003)
36. Miyato, T., Koyama, M.: cGANs with projection discriminator. In: *International Conference on Learning Representations* (2018). <https://openreview.net/forum?id=ByS1VpgRZ>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Spot the Fake Lungs: Generating Synthetic Medical Images Using Neural Diffusion Models

Hazrat Ali¹(✉), Shafaq Murad², and Zubair Shah¹

¹ College of Science and Engineering, Hamad Bin Khalifa University,
Qatar Foundation, Doha, Qatar
{haali2,zshah}@hbku.edu.qa

² Manchester University NHS Foundation Trust, Manchester Royal Infirmary,
Oxford Road, Manchester M13 9WL, UK
shafaq.murad@mft.nhs.uk

Abstract. Generative models are becoming popular for the synthesis of medical images. Recently, neural diffusion models have demonstrated the potential to generate photo-realistic images of objects. However, their potential to generate medical images is not explored yet. We explore the possibilities of synthesizing medical images using neural diffusion models. First, we use a pre-trained DALLE2 model to generate lungs X-Ray and CT images from an input text prompt. Second, we train a stable diffusion model with 3165 X-Ray images and generate synthetic images. We evaluate the synthetic image data through a qualitative analysis where two independent radiologists label randomly chosen samples from the generated data as real, fake, or unsure. Results demonstrate that images generated with the diffusion model can translate characteristics that are otherwise very specific to certain medical conditions in chest X-Ray or CT images. Careful tuning of the model can be very promising. To the best of our knowledge, this is the first attempt to generate lungs X-Ray and CT images using neural diffusion models. This work aims to introduce a new dimension in artificial intelligence for medical imaging. Given that this is a new topic, the paper will serve as an introduction and motivation for the research community to explore the potential of diffusion models for medical image synthesis. We have released the synthetic images on <https://www.kaggle.com/datasets/hazrat/awesomelungs>.

Keywords: Diffusion models · Generative models · Artificial intelligence · Medical imaging · Lungs · CT · X-ray

1 Introduction

During the last decade, there has been a surge in studies on generative models for medical image synthesis [1, 2]. Generative Adversarial Networks (GANs) and deep autoencoders are two primary examples of deep generative models that have shown remarkable advancements in synthesis, denoising, and super-resolution of

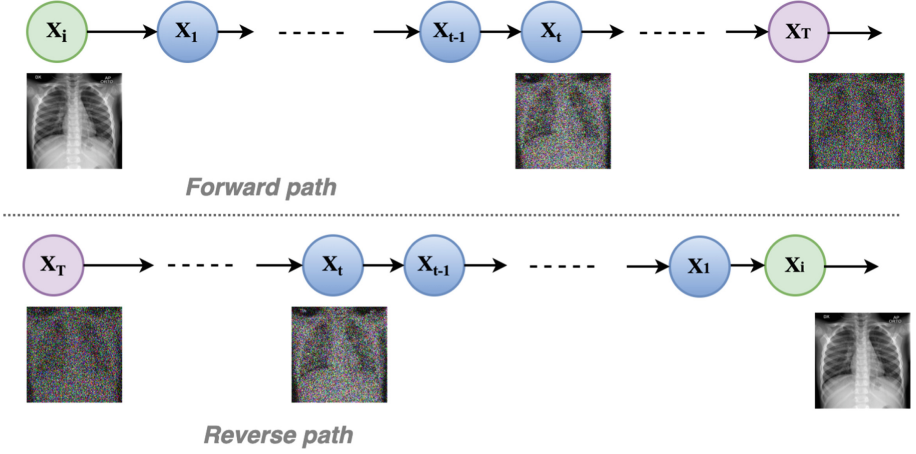


Fig. 1. Forward pass and reverse pass in diffusion model training. Figure modified from [7].

medical images [1, 3]. Many studies have shown the great potential of GANs to generate realistic magnetic resonance imaging (MRI), Computed Tomography (CT), or X-Ray images that can help in training artificial intelligence (AI) models [1, 4–6]. With the recent success of neural diffusion models for the synthesis of natural images [8, 9], there is now an increasing interest in exploring the potential of neural diffusion models to generate medical images. For generating natural images such as art images, objects, models such as DALLÉ2¹, Mid-Journey², and Stable Diffusion³ have pushed the state-of-art. Amongst the three, only the latter is available with open-source code. Compared to GANs, diffusion models are becoming popular for their training stability.

A diffusion model, in simple words, is a parameterized Markov chain trained using variational inference. The transition is learned through a diffusion that adds noise to the data. In principle, the diffusion model transforms the input data into noisy data by adding Gaussian noise and then recovers the data distribution by reversing the noise. Once the model learns the distribution, it can generate useful data from random noise input. So, diffusion models transform a latent encoded representation into a more meaningful representation of image data. In this context, diffusion models can be compared to denoising autoencoders. As shown in Fig. 1, the overall process can be summarized as a two-step phenomenon, the forward pass, i.e., the transformation of the data distribution to noise (X_i to X_T), and the reverse pass, i.e., reversing the noise distribution to data distribution (X_T to X_i). Training a diffusion model implies the learning of the reversing process i.e., $p(x_{t-1}|x_t)$. The diffusion model can be implemented

¹ <https://openai.com/>.

² <https://www.midjourney.com/home/>.

³ <https://github.com/CompVis/stable-diffusion>.

by using a neural network for the forward and reverse training steps. However, the architecture must have the same input and output dimensions.

While previously, the generating ability of diffusion models was mostly used for unconditional generation of data, more recent attempts have shown conditioned generation by introducing guided-diffusion models [8–10]. These works have demonstrated the generation of photo-realistic images guided by the context of the input text or image. The existing use cases of diffusion models comprise text-to-image applications, i.e., generating images according to a given text prompt. In addition, Han et al. [11] presented a classification and regression diffusion model (CARD), and demonstrated the use of the diffusion model for classification as well as regression tasks. In CARD, the authors approached the task of supervised learning using generative modeling conditioned on the class labels. Though the objective was not to claim state-of-the-art results, the method has shown promising results on the benchmark dataset. For CIFAR-10 classification, the model reached an accuracy of 90.9%.

Given the potential of diffusion models to learn the representation, one can expect their potential to generate a diverse set of medical images. Furthermore, they can add a new dimension to existing approaches for medical image applications, such as noise adaptation, noise removal, super-resolution, domain-to-domain translation, and data augmentation. To the best of our knowledge, no work other than the recent pre-print [12], exists currently on the synthesis of medical images using neural diffusion models. Walter et al. [12] used latent diffusion models to generate T1w MRI images of the brain. Using 31,740 brain MRI images from the UK Biobank, they have generated a stack of 100,000 images conditioned on key variables such as age, sex, and brain volume. In this work, we explore neural diffusion models to generate synthetic images of lung CT and X-Ray. We use the DALLE2 model and the stable diffusion model to generate the images and present them to two radiologists for their feedback. We then summarize the feedback received from the radiologists and identify some of the challenges in using the neural diffusion model for medical image synthesis.

The remaining paper is organized as: Sect. 2 explains the methodology of our work. Section 3 presents the results of generating lung CT and X-Ray images, while Sect. 4 provides insights into the results and also highlights the limitations of the approach. Finally, Sect. 5 concludes the paper.

2 Methodology

In this work, we devised two experiments for generating synthetic images of lungs X-Ray and CT. In the first experiment, we used the *OpenAI* DALLE2 API⁴ to generate images based on the input text. The DALLE2 model recently gained much attention for its ability to generate photo-realistic images of objects given a certain input text. Using the API, we generated multiple images of lungs CT and X-Ray. We then presented a randomly selected set of the generated images to two trained radiologists. We asked the radiologists for two key tasks. First,

⁴ <https://openai.com/>.

we asked them to label each image as real, fake, or uncertain about, as per their perceived understanding. Second, we asked them to provide a brief description of the possible information related to lung condition or diagnosis of disease (for example, normal lungs, severely damaged lungs, pneumonia-affected lungs, etc.). The radiologists did not have prior information on the labels of the images. In fact, all the images that we presented to the radiologists were synthetic. The radiologists did not know each other and performed the tasks independently. Of the two radiologists, one radiologist had prior knowledge of artificial intelligence and generative models, while the other radiologist was naïve to deep generative models.

In the second experiment, we used the stable diffusion model [13]. We trained the stable diffusion model using 3165 X-Ray images from [14]. We resized the images to 256 by 256 resolution. No other pre-processing was done. Using the X-Ray images, we trained a stable diffusion model on a server equipped with NVIDIA Quadro RTX 8000 GPU with a 48 GB memory. We set the batch size equal to 32 and ran the training for 700000 training steps.

3 Results

Using the DALLE2 API, we generated a total of 150 images. We have uploaded the synthetic images to Kaggle⁵. We believe the number of generated images is only limited by the tokens available to us. Sample X-Ray and CT images are shown in Fig. 2 and Fig. 3, respectively. Out of 40 images that we presented to the radiologists, radiologist \mathcal{A} identified 14 X-Ray images and three CT images as real, while four X-Ray and 17 CT images as fake. Radiologist \mathcal{A} labeled two X-Ray images as unsure. The second radiologist (radiologist \mathcal{B}) identified ten X-Ray images and only two CT images as real, while all the remaining images as fake.

Agreement between radiologists: Of the 20 CT images, only three images were labeled as real by both radiologists. Similarly, five X-Ray images were marked as real by both radiologists. There were two X-Ray and two CT images for which both the radiologists were uncertain.

For task 2, where we asked the radiologist to provide a brief description of what the images may reveal, the radiologists made some interesting observations. For example, some descriptions are listed in Table 1. These descriptions clearly reveal that some of the images carried representations similar to real X-Ray or CT images, and the model was able to generate features that are specific lung conditions.

4 Discussion

Some of the generated images lacked the characteristics of realistic images and were quickly identified by the radiologists as fake. These images were termed

⁵ <https://www.kaggle.com/datasets/hazrat/awesomelungs>.

Table 1. Samples of remarks from radiologists (no-specific order)

Image modality	Remarks*
CT	Possible effusions
	Pneumonia
X-Ray	Left lower lobe effusions
	Possibility of pneumonia
	Bilateral infection

*The remarks do not imply a definite decision.

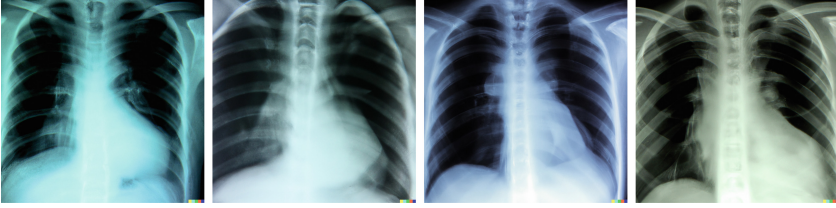


Fig. 2. Samples of lungs X-Ray images generated with the diffusion model.

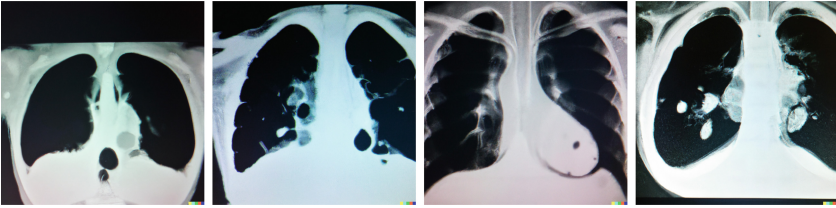


Fig. 3. Samples of lungs CT images generated with diffusion model.

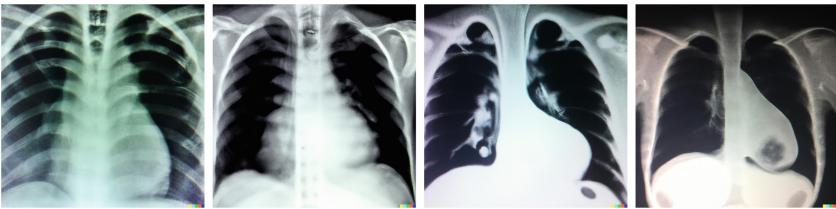


Fig. 4. Samples of synthetic images for lungs X-Ray (left two images) and CT (right two images) identified as fake by at least two radiologists.

as having unusual ribs appearance or showing unusual exposure. Similarly, it was easy to spot big vessels contour and lung fields that appeared to have been drawn and not imaged. One key observation for fake images was that the trachea is visible behind the heart shadow, which does not happen in real X-Ray imaging. A few sample images that were termed fake by at least two radiologists are

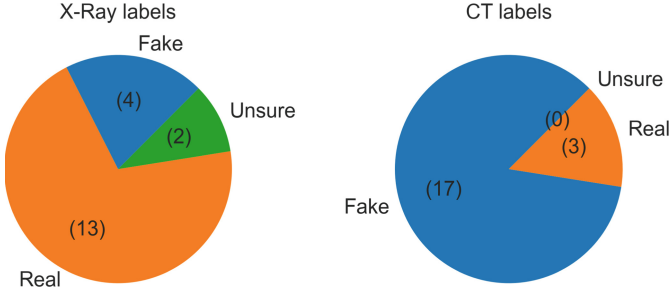


Fig. 5. Evaluation by Radiologist \mathcal{A}

shown in Fig. 4. Many of the generated images from the pre-trained model clearly lacked the characteristics of realistic images and were quickly identified by the radiologists as fake. These images were termed as having unusual ribs appearance, strange clavicle appearance, or showing unusual exposure. The evaluation by radiologist \mathcal{A} is summarized in Fig. 5.

4.1 Limitations

One challenge identified in diffusion models is the limited ability to produce details in complex scenes [9]. So, generating complex medical images would need to be complemented with noise adaptation or super-resolution techniques [5]. Like many other AI models, diffusion model training is prone to bias in the dataset; for example, unbalanced representation of medical conditions in the input X-Ray or CT image or inherent noise in data. Thus, the synthetic data from such a diffusion model will also carry the bias. Eventually, if the generated data are made public and used for onward model training, the bias may turn into a cascade behavior and will be further augmented [7]. The model has been used pretty much as a black-box model; hence, not much explainability can be offered on how certain images were generated. Unlike the work reported in Walter et al. [12], our generated images are not conditioned on additional variables such as gender, age, etc. Diffusion models are very slow to train as they require the number of training steps to be in the order of several hundred thousand. Our training took around one day for 100k training steps. This study is presented as a means to infuse interest in the potential of diffusion models for the synthesis of medical images.

5 Conclusion and Future Work

In this work, we have demonstrated the potential of neural diffusion models for the synthesis of lungs X-Ray and CT images. Though the radiologists spotted many images as fake, few images were still labeled as real by them. The labeling from the radiologists reflects that some of the generated X-Ray images carried

a great resemblance to real images. However, the identification of fake images was straightforward for the CT images. Through qualitative analysis of the generated images, we showed that neural diffusion models have great potential to learn complex representations of medical images. Although the performance of diffusion models is superior to GANs-based methods for synthesizing natural images, research efforts on the diffusion model for medical image synthesis have yet to mature.

Acknowledgments. The authors are grateful to Surendra Maharjan from Indiana University Purdue University Indianapolis, USA, for useful comments on this work. The authors are thankful to Dr. Jens Schneider for facilitating the GPU access.

References

1. Yi, X., Walia, E., Babyn, P.: Generative adversarial network in medical imaging: a review. *Med. Image Anal.* **58**, 101552 (2019)
2. Jiang, Y., Chen, H., Loew, M., Ko, H.: Covid-19 CT image synthesis with a conditional generative adversarial network. *IEEE J. Biomed. Health Inform.* **25**(2), 441–452 (2020)
3. Chen, M., Shi, X., Zhang, Y., Wu, D., Guizani, M.: Deep feature learning for medical image analysis with convolutional autoencoder neural network. *IEEE Trans. Big Data* **7**(4), 750–758 (2017)
4. Ali, H., et al.: The role of generative adversarial networks in brain MRI: a scoping review. *Insights Imaging* **13**(1), 1–15 (2022)
5. Ahmad, W., Ali, H., Shah, Z., Azmat, S.: A new generative adversarial network for medical images super resolution. *Sci. Rep.* **12**(1), 1–20 (2022)
6. Munawar, F., Azmat, S., Iqbal, T., Grönlund, C., Ali, H.: Segmentation of lungs in chest X-ray image using generative adversarial networks. *IEEE Access* **8**, 153535–153545 (2020)
7. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851 (2020)
8. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. In: *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794 (2021)
9. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. *arXiv preprint [arXiv:2204.06125](https://arxiv.org/abs/2204.06125)* (2022)
10. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. *arXiv preprint [arXiv:2011.13456](https://arxiv.org/abs/2011.13456)* (2020)
11. Han, X., Zheng, H., Zhou, M.: CARD: classification and regression diffusion models. *arXiv preprint [arXiv:2206.07275](https://arxiv.org/abs/2206.07275)* (2022)
12. Pinaya, W.H., et al.: Brain imaging generation with latent diffusion models. *arXiv preprint [arXiv:2209.07162](https://arxiv.org/abs/2209.07162)* (2022)
13. O'Connor, R.: How to run stable diffusion locally to generate images. <https://www.assemblyai.com/>. Accessed 01 Oct 2022
14. Chowdhury, M.E., et al.: Can AI help in screening viral and Covid-19 pneumonia? *IEEE Access* **8**, 132665–132676 (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Multi-Graph Convolutional Neural Network for Breast Cancer Multi-task Classification

Mohamed Ibrahim^(✉), Shagufta Henna, and Gary Cullen

Atlantic Technological University, Letterkenny, Donegal, Ireland
{L00157116,Shagufta.henna,Gary.cullen}@atu.ie

Abstract. Mammography is a popular diagnostic imaging procedure for detecting breast cancer at an early stage. Various deep-learning approaches to breast cancer detection incur high costs and are erroneous. Therefore, they are not reliable to be used by medical practitioners. Specifically, these approaches do not exploit complex texture patterns and interactions. These approaches warrant the need for labelled data to enable learning, limiting the scalability of these methods with insufficient labelled datasets. Further, these models lack generalisation capability to new-synthesised patterns/textures. To address these problems, in the first instance, we design a graph model to transform the mammogram images into a highly correlated multigraph that encodes rich structural relations and high-level texture features. Next, we integrate a pre-training self-supervised learning multigraph encoder (SSL-MG) to improve feature presentations, especially under limited labelled data constraints. Then, we design a semi-supervised mammogram multigraph convolution neural network downstream model (MMGCN) to perform multi-classifications of mammogram segments encoded in the multigraph nodes. Our proposed frameworks, SSL-MGCN and MMGCN, reduce the need for annotated data to 40% and 60%, respectively, in contrast to the conventional methods that require more than 80% of data to be labelled. Finally, we evaluate the classification performance of MMGCN independently and with integration with SSL-MG in a model called SSL-MMGCN over multi-training settings. Our evaluation results on DSSM, one of the recent public datasets, demonstrate the efficient learning performance of SSL-MNGCN and MMGCN with 0.97 and 0.98 AUC classification accuracy in contrast to the multitask deep graph (GCN) method Hao Du et al. (2021) with 0.81 AUC accuracy.

Keywords: Graph modelling · Self-supervised learning · Semi-supervised learning · Breast cancer classification · Graph convolutional neural networks

1 Introduction

Breast cancer is the most common malignancy among adult women of all ages, accounting for over 7.8 million cases in the last five years [1]. Early detection of

breast cancer improves survival rates by significantly limiting the risk of tumour progression and helping to increase patients' life expectancy [2, 3]. Screening for cancers in mammography involves diagnosing methods to expose most breast malignancies in early stages. Radiologists diagnose these malignancies by detecting and examining the mass and calcification regions based on various visual signs, including size, edges, distribution, relations, and clustering [4, 5]. However, exposing these signs requires substantial expertise and are prone to high error rates of 20% [6]. Because of these challenges, and especially with the advancements in machine learning, recent years have witnessed dramatic developments of several computer vision models striving to extract enough hidden features from mammogram images to improve detection and classification sensitivity of breast cancers [7]. However, most of these techniques are significantly hindered by supervised machine learning approaches that require large datasets of accurately annotated images for training. Furthermore, in mammography, labelling malignancy regions, i.e., regions of interest (ROI), is a tedious procedure requiring pathologic expertise for considered time, making the process time-consuming and costly [8]. Thus, the availability of sufficiently labelled data is a critical bottleneck for supervised learning models, limiting the training, therefore, the performance and accuracy of the most recent models. As a result, current methods consistently adopt various techniques, including data augmentation, multi-view image generation, and transfer learning to mitigate inadequate data limitations and tune classification performance [9]. Work in [10] addressed the challenges of data limitation in the breast cancer domain by using transfer learning in CNN. The proposed method combined the pre-trained CNN, VGG16 [11], with a fully connected layer to perform binary classification of normal and abnormal mass in mammograms. Another work in [12] augmented the pre-trained VGG16 and Resnet50 [13] to a convolutional network model to perform a whole mammogram image classification. Authors in [14] applied multi-view, transfer learning and augmentation techniques to improve a CNN model performance with limited data.

Apparently, most of the techniques proposed to tackle the data limitation augmented to various end-to-end convolutional neural networks (CNNs) architectures, i.e., VGG16, Resnet, AlexNet, GoogleNet [15, 16]. CNNs employ fixed 2D kernels to encode images that contain well-defined and distinguishable objects, excluding the positions and orientations. However, mammography images are rich in heterogeneous textures that are difficult to classify based solely on their morphological shapes, so their geometric relations and dependencies should be considered [17].

Noticeably, a handful of approaches privilege the relationship between texture features to improve the performance of the CNN-based framework. Heyi Li et al. [33] augmented locality preserving and conditional graph learners module to a dual CNN model that maps between the ROIs and provided labels to improve the classification performance of breast mass. In addition, works in [25, 26] proposed a cross-view CNN model to construct the relationship between the features of two views of the mammograms, i.e., the mediolateral oblique

(MLO) and the craniocaudal (CC). These techniques improve the performance of the mass detection models by exploiting the feature correlations. However, these methods lack generalisation capabilities as they are restricted to detecting the mass abnormalities in mammograms that are relatively large compared with other abnormalities such as calcifications clusters. More recently, graph-based deep learning approaches have demonstrated excellent advancements in machine learning, from solving complex geometric problems to handling massive data connections and learning data dependencies [18]. Moreover, relational awareness of graph-based models enables semi-supervised, and self-supervised learning approaches in various domains [20]. Consequently, graph-based models are proficient at circumventing the availability constraints of labelled mammograms by effectively privileging the inherited relations and dependencies in data to achieve improved accuracy with fewer labelled examples.

Very recently, several efforts have emerged to classify breast cancer using graphs, such as those used in [19, 21, 22]. These methods illustrate the advantages of graph-based models over conventional CNN models by modelling mammograms into graphs and performing binary graph classification. Another work in [17] highlights these advancements by performing a multi-classification of graphs modelled for calcification distributions in mammograms. The authors used the graph convolutional network (GCN) model that outperformed various CNN-based models, with a margin of over 10%. However, these techniques model ROIs in mammography into graphs, thus they are still limited because of the necessity of sufficiently well-annotated data.

Noticeably, in the entire cancer detection domain, significantly few graph-based models augment techniques to tackle the limitations of labelled data. For example, work in [23] proposed a weakly supervised GCN model to detect prostate cancer rates in histopathology slides. The proposed model outperforms the baseline supervised GCN by 36% and achieves 96% accuracy. Another method in [24] considers a self-supervised learning task to improve the performance of the graph neural networks (GNN) to classify breast cancer in histopathology images. The proposed approach outperforms other supervised GNN models by almost 20%. However, these methods assume a general classification of specific regions of histopathological images, which are less complex and computationally simple than mammography.

Considering all recent techniques, detecting and classifying breast cancer in mammography with minimal required annotated data and considering the relationship and pattern of the texture features is still an open problem. To the best of our knowledge, no self-supervised or semi-supervised graph-based technique has been previously proposed to process the high-resolution mammogram images and perform multi-classification of the anomalous regions with less annotated data requirements for the training process. However, as the learning capacities of graph-based models rely on the features and relations embedded in the graph, a well-engineered preprocess is necessary to transform the raw data of digitised mammogram images into a rich relational graph network.

This work models full mammogram images into efficient graph representations by capturing the heterogeneous features of high-level texture details and critical relations and patterns that contribute to diagnosing decisions. The proposed framework comprises a mammogram to multigraph transformer module (MMG) that segments the full-scale mammogram images into focused multi-region. It augments a pre-trained residual neural network (Res-Net) to transform each segment into high-level textures and spatial features called embeddings, resulting in a weighted graph. MMG also reinforces the features representation by generation multigraph that combines hundreds of graphs into a highly correlated network of thousands of nodes and edges.

The proposed framework includes a semi-supervised module, namely mammogram multigraph convolutional network module (dubbed MMGCN) for node classification. The MMGCN processes graph embeddings through stacked convolutional neural network layers followed by a fully connected network. It improves graph representations through semi-supervised learning replaces the embedding of each node with higher-level augmented embedding.

Furthermore, to reduce the need for a large annotated dataset, this work integrates a pre-training self-supervised learning process into the MMGCN by augmenting a self-supervised learning multigraph encoder (SSL-MG) to improve the feature representations. The SSL-MG improves the nodes embeddings through an adversarial process, discriminating between the series of node pairs, i.e., ordered and randomly generated nodes. Finally, the proposed framework classifies each node into normal cells or any of the breast abnormalities, i.e., mass malignant or benign and calcification malignant or benign.

2 Proposed Method

2.1 Notations and Problem Definition

Given a mammogram dataset D that consists of a number of images, $I = \{I_i\}_1^{|D|}$. Let each image I_i can be divided into K segments $S = \{S_i\}_1^{|K|}$ where each segment S_i has texture features S_i^T , spatial details S_i^S , and category $S_i^C \in \{0 : normal, 1 : massMalignant, 2 : massBenign, 3 : calcificationMalignant, 4 : calcificationBenign\}$.

Each image I_i can be modelled as a graph $G_i = (V, E)$ where $V \leq |S|$ is the set of nodes assigned to non-zero segments, and $E \subseteq A = V \times V$ is a set of edges connecting the nodes based on an adjacency matrix A . If $\{v_i, v_j\} \in V$ are two nodes representing adjacent segments, so an edge connect them and denoted as $e_{i,j} \in E$. Graph G_i is weighted using the correlation between the segmented images as features H_E added to all edges E and the vectorization of the high level texture features of the image segments S as features $H_V \in \mathbb{R}^{d|S|}$ added to all nodes.

Modelling a complete mammogram dataset D consisting of $|D|$ images generating a set of weighted graphs $G = \{G_i\}_1^{|D|}$. In order to enrich the encoded

mammograms features and relationships, a complex multi-graph \mathcal{G} is constructed by connecting all graphs as a united graph $\mathcal{G} = \bigcup_{G_1}^{G_{|D|}} (G_i)$.

Given a multi-graph \mathcal{G} with initial embeddings H^0 and a small subset of labelled nodes V^L , our aim in this work is to improve graph representation through a self-supervised pretext task. Then use semi-supervised downstream model to computes the loss between the given labels S^C and embeddings H^l of labelled nodes V^L and update the learnable weight W . Finally each node gets final embedding Z_i and as each Z_i present a segment in a mammogram, so each segment get classified as $S_i \rightarrow S_i^C$ with better accuracy than predicting a general class for the whole image.

2.2 Mammograms to Multi-Graph Modelling (MMG)

This work proposes a mammogram multi-graph transformer (MMG) as presented in Fig. 1 and given in *Algorithm 1* in the appendix. Mammograms are high-resolution images composed of heterogeneous pixels with values varying between black and white, i.e., $0 \sim 1$. To fully capture the features in these mammograms, the proposed MMG module transforms each image to a graph embedded with texture and spatial features representing nodes and edges.

Initially, MMG divides each mammogram image into K segments, then encode the texture features S_i^T of these segments using a pre-trained ResNet-18 [27] model. ResNet-18 is composed of a series of residual blocks of localised convolutional and pooling layers that vectorise the texture features S_i^T of each sub-image S_i into a 512-length vector \vec{X} as given in Eq. (1). MMG embeds the encoded vectors \vec{X} as node features H_V embedded into graph nodes V .

$$\vec{X} = \mathcal{F}_{Res}(\mathbf{S}_i^T, \{W_i\}) + W_s \mathbf{S}_i^T \quad (1)$$

Here S_i^T and \vec{X} denotes input and output of the residual network layers, while W_i and W_s represents the layers and linear projections of the ResNet.

MMG encodes the Cartesian coordinates of each mammogram segments to generate edges list and a adjacency matrix A defining the connected nodes in each graph \mathcal{G}_i . In order to preserve the correlation between the segmented images of the mammogram, MMG uses the cosine similarity [28] illustrated in Eq.(2) to weight graph edges by values varying between 1 for edges connecting nodes representing the same features and 0 for pairs of nodes with entirely unmatched features.

$$H_E \leftarrow \cos(A, B) = \sum_{k=1}^n \frac{A_k \cdot B_k}{\|A_k\| \cdot \|B_k\|} \quad (2)$$

A_k and B_k denote vectors A and B components, whereas n represents the number of components. As equal length N-dimensional arrays represent both vectors, the components are the elements of these arrays. MMG optimises the generated graph by pruning nodes and edges representing the background segments of the mammogram image. Then it assigns a class for each node using the region of

interest (ROI) binary masks. The binary mask consists of pixels with 0 values except for the region of abnormality with pixel values of 1. MMG combines the optimised graphs using the common nodes in non-Euclidean spaces to generate the final complex multi-graph network as given in Eq. (3). The equation unites a set of N graphs representing the entire mammography dataset images D where $N = |D|$ and each graph is composed of nodes \mathcal{V} , edges \mathcal{E} , and features $\mathcal{H}_{\mathcal{V}}, \mathcal{H}_{\mathcal{E}}$.

$$\mathcal{G} = \bigcup_{G_1}^{G_{|D|}} (\mathcal{V}, \mathcal{E}, \mathcal{H}_{\mathcal{V}}, \mathcal{H}_{\mathcal{E}}) \quad (3)$$

Now \mathcal{G} is the modelled graph network for the entire mammogram dataset. As \mathcal{G} composes all segments of mammogram images as nodes, each node can be classified based on the embedded features and the relation to other nodes into one of 5 classes. These classes are normal, mass-Malignant, mass-Benign, calcification-Malignant, and calcification-Benign.

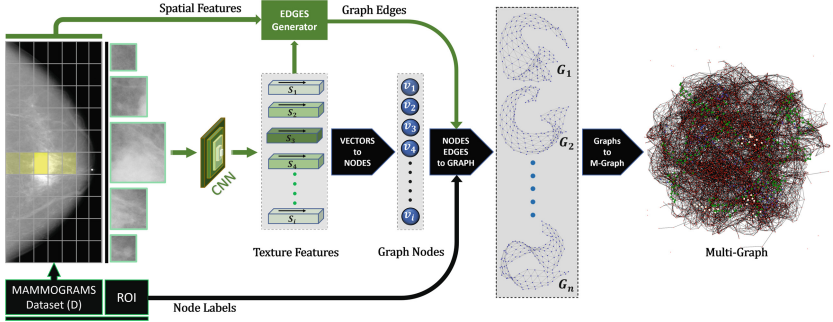


Fig. 1. The Multi-Graph modelling framework processes the mammogram images, segments them and generate nodes and edges to model a graph for each image, then it combines all generated graphs into a multi-graph.

2.3 Multi-Graph Self-Supervised Learning (SSL-MG)

This stage process the modelled mammogram multi-graph \mathcal{G} by the proposed SSL-MG encoder to improves the segmented image features embedded in nodes based on a self-supervised pretext task. SSL-MG encoder comprises nodes and graph readers, discriminators, and GCN layers [32] stacked with pooling and fully connected layers. SSL-MG employs a mini-batch generator [29] to process the multi-graph \mathcal{G} as a series of sub-graphs \mathcal{G}^* to fit less memory. As features of the mammogram segmented images are vectors $H_{\mathcal{V}}$ of length K embedded in multi-graph nodes have large scale varies values, so SSL-MG normalises them

for better computation to values between 0 and 1 using Eq. (4).

$$\hat{H} = \frac{H}{\sqrt{\sum_{k=1}^n H_k^2}} \quad (4)$$

Additionally, the weighted adjacency matrix A of the mammogram multi-graph is normalised using the symmetric normalisation trick illustrated by Kipf and Welling in [20]. Equation (5) normalises A after adding self connection for all nodes using the unit matrix I_N then multiplies it with the two inverses of the square root of the degree matrix D [32].

$$\hat{A} = D^{-1/2} * (A + I_N) * D^{-1/2} \quad (5)$$

SSL-MG first aggregates and down-samples the features H into an embedding Z^* that summarises the sub-graph \mathcal{G}^* . Equation (6) computes Z^* by matrix multiplication of the normalised adjacency matrix of the sub-graph \hat{A}^* , the normalised features \hat{H} , and network weight W . SSL-MG then uses this embedding in a self-supervised pretext task to discriminate between a series of features, one for the nodes of the same sub-graph h_i and another for random nodes h_i^T (Fig. 2).

$$Z^* = \hat{A}^* * \hat{H} * W \quad (6)$$

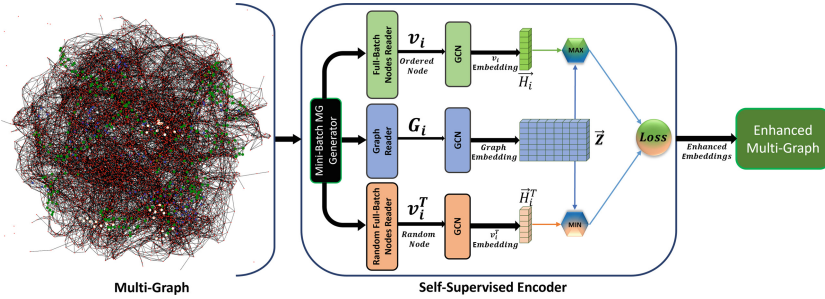


Fig. 2. SSL-MG encoder: The encoder processes the complex multi-graph network in batches, generates a graph summary of each batch, and compares it to the embeddings of pair of nodes, one in sorted order and the second using random node. The loss function at the end compares the similarity of these embeddings.

SSL-MG process three inputs include embeddings of the sorted nodes h_i , embeddings of an opposing random node h_i^T , and the computed graph summary Z^* . The encoder learns the node presentation by maximizing the similarities between the sorted nodes and the graph summary while decreasing it for the random nodes. For that, SSL-MG in Eq. (7) uses logistic sigmoid non-linear function σ to compute the probability of (h_i, Z^*) and (h_i^T, Z^*) , then compute

the sub-graph sigmoid cross-entropy loss \mathcal{L}^*_{SCE} for all the nodes M and N . The total loss \mathcal{L}_{SCE} then calculated by aggregating the loss of a k of \mathcal{G}^*

$$\mathcal{L}_{SCE} = \sum_{i=1}^K \frac{1}{N+M} \left(\sum_{i=1}^N \log(\sigma(h_i \mathbf{W} Z^*)) + \sum_{j=1}^M \log(1 - \sigma(h_j^T \mathbf{W} Z^*)) \right) \quad (7)$$

SSL – MG encoder minimizes the cross-entropy loss calculated by Eq. (7) by using the adaptive momentum (Adam) function. This let the encoder learn the representation of the graph and generate high-level embeddings to replace the existing for each node. The MG-SSL encoder tunes the features of segmented mammogram images embedded in the multi-graph nodes. Later, these embeddings are used as an input for the downstream model.

2.4 Mammogram Multi-Graph Convolutional Network Classifier (MMGCN)

MMGCN is a multi-node classifier model designed to either processes the initial features of the mammogram segmented images embedded in the multi-graph or the tuned nodes embeddings generated from the SSL-MG encoder as depicted in Fig. 3. *MMGCN* processes the input of the mammogram multi-graph batches \mathcal{G}^* same way as the SSL-MG by normalising the nodes embeddings and adjacency matrix using Eqs. (4) and (5) respectively. In addition, *MMGCN* employs a data balancing procedure to guarantee that the nodes categories $S_i^C \in \{0 : normal, 1 : massMalignant, 2 : massBenign, 3 : calcificationMalignant, 4 : calcificationBenign\}$ are presented equally in each sub-graph \mathcal{G}^* . As the mammogram multi-graph includes nodes represent images segments of normal sections in large numbers compared to the other categories, this step required to avoid any bias through the downstream process.

MMGCN includes 4 GCN layers to aggregate the features of each node and its neighbours, then normalises and processes each aggregation with learnable weight W through a standard dense layer. The GCN layers perform that through matrix multiplication of the normalised adjacency matrix with self-connection $\hat{\mathbf{A}}^*$, the normalised features matrix $\hat{\mathbf{X}}$ and the learnable weight W . As in Eq. 8, these multiplications get activated using none linear function typically Relu. However, the last GCN layer uses softmax activation function as in Eq. 9.

$$\mathbf{H}_i^L = Relu(\hat{\mathbf{A}}^* * \hat{\mathbf{H}}_i^0 * \mathbf{W}^0) \quad (8)$$

$$\mathbf{Z}_i = SoftMax(\hat{\mathbf{A}}^* * \mathbf{H}_i^L * \mathbf{W}^L) \quad (9)$$

As the initial mammogram multi-graph composes nodes, each one is embedded with the encoded features h_i of a single image segment S_i . Now *MMGCN* generates higher-level embedding Z_i that embeds features of all neighbour segments in each node. By getting an embedding $Z_i = \sum_i \exp(h_{V_i})$ for each node, the softmax uses it to calculate the probability of each node class $p(\mathcal{S}_i^C | \mathbf{Z}_i)$.

Using a subset of labelled nodes $V^L \in V$ represent annotated mammogram segments $S_i^C \in S$, the categorical cross-entropy loss \mathcal{L}_{CCE} can be calculated

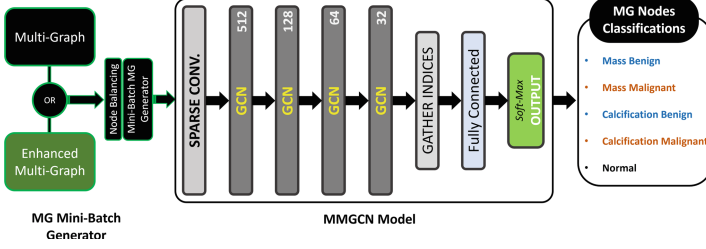


Fig. 3. The MMGCN model either processes the initial mammogram multi-graph generated by the MMG or the improved multi-graph generated by SSL-MG encoder. The MMGCN use a mini-batch generator and sparse convolution layer as an input layer to process all the input tensors efficiently. Additionally, the model comprises of four graph convolutional layers of 512, 128, 64, and 32 units, followed by an aggregation of indices and a fully connected layer to sort and compute the output followed by softmax function.

through a semi-supervised training using Eq. (10). Finally, a stochastic gradient descent optimiser uses this loss to train the neural network weights W .

$$\mathcal{L}_{CCE} = \sum_{i=1}^{|V^L|} S_i^C \cdot \log Z_i \quad (10)$$

3 Experiments

3.1 Dataset

We validate our frameworks, i.e., MMGCN, and SSL-MMGCN, with public mammography dataset, CBIS-DDSM [31]. The dataset contains scanned images of digitised mammograms in the digital imaging and communications in medicine format (DICOM), a standard format for screening in the medical domain. The dataset contains 2,620 mammography images in two standard views, MLO and CC. In addition, CBIS-DDSM has training samples that include annotation binary masks for the ROI that indicate the general positions of anomalies within mammograms. The dataset included 557 patient mammograms with calcification anomalies, 646 with mass anomalies, and 45 with both anomalies. Moreover, each type of anomaly is classified as either malignant or benign. The mammograms in the raw data have varied large-scale dimensions to provide enough capability for zooming and analysis. Using the CBIS-DDSM dataset, MMG encodes 1138 mammograms in a complex multi-graph. This multi-graph contains 285413 nodes: 3478 represent mass-malignant regions, 2928 represent mass-benign regions, 1596 represent calcification-malignant regions, and 2033 represent calcification-benign regions, while the remaining nodes encode normal lesions.

3.2 Experiment Setup

The experiment setup is crucial in machine learning, as we should consider various measurements to avoid data leakage, overfitting, and bias. Especially in graph learning message passing and feature smoothing over neighbouring nodes. Hence, for the training process of the SSL-MMGCN and MMGCN models, we load a multi-graph with 40% and 50% of labelled nodes from each class, respectively. We include the remaining nodes unlabeled in the multi-graph for the validation process. Then, to avoid bias during the training process, the node balancing module generates an equal number of nodes from each class in each mini-batch. Also, the MMGCN model employs a 0.5 drop rate to reduce overfitting and perform smooth learning.

3.3 Performance Evaluation

SSL-MMGCN Learning. The SSL-MG encoder is trained on a multi-graph network of 7500 unlabelled nodes for training over 200 epochs. The convergence of the model optimiser is depicted in Fig. 4(a). Over the first 50 epochs, the decay rate demonstrates a rapid convergence with a drop in the loss value from 0.7 to less than 0.1. However, with further training over the last 150 epochs, the loss steadily declines to a value close to zero. Through the training process of the SSL-MG, the model learns the node and graph representation and replaces the features of the nodes with higher-level information based on the learning efficiency of the self-supervised task. Then, by using the generated embedding as an input for the MMGCN in the SSL-MMGCN framework, semi-supervised learning training is performed using only 50% of the nodes, while the rest are for validation and testing. The SSL-MMGCN training and validation loss rate over 1000 training epochs illustrated in Fig. 4(b) shows a decrease to less than 0.25. The decline in losses and the modest variations between the training and validation losses indicate that the downstream SSL-MMGCN model has an effective learning rate. Figure 4(c) shows the accuracy improvement of the SSL-MMGCN model through this training. The model accuracy efficiently exceeds 95% at the end of the 1000 training epochs with a continuous learning rate, albeit a slow learning rate after 900 epochs, which implies the convergence of the SSL-MMGCN model. The significant increase in training and validation accuracy rates shows the learning capacity’s efficacy, especially with the labelled to unlabelled data ratio. SSL-MMGCN uses only 50% of the multi-graph nodes to calculate the categorical cross-entropy loss and adjust the learnable weight W using the ADAM optimizer. The loss in Fig. 4(b) and the accuracy in Fig. 4(c) demonstrate continuous gradient descent, learning without over-fitting. However, after 300 epochs, the loss increases and the accuracy decreases, which illustrates a non-optimal local minimum. However, after a few epochs, the model optimises with better gradient descents. The efficient fitting of the model shows that increasing the number of labelled nodes or training epochs allows SSL-MMGCN to attain improved accuracy.

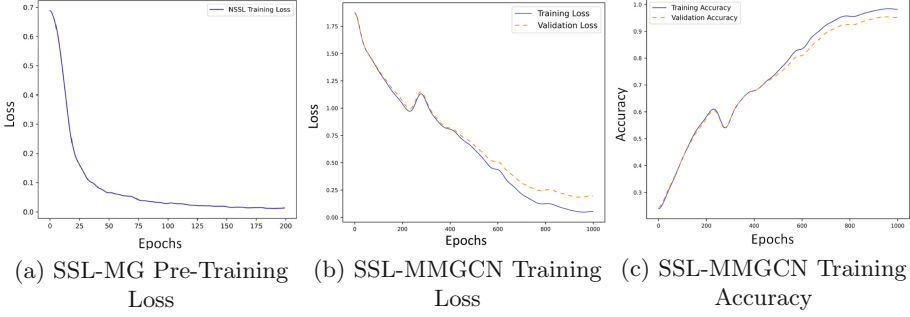


Fig. 4. The first figure shows the loss rate over the training task of the SSL-MG, while the other two figures are the training loss and accuracy of the SSL-MMGCN framework.

Mammogram Classification Analysis. In the medical domain, confusion among the classes is crucial in the diagnosis process, and the percentage of false and true positives is considered. So, to investigate the sensitivity and specificity of the MMGCN model in classifying the categories of breast cancer anomalies in mammography, the confusion matrix is computed, as shown in Fig. 6. The results show that the true-positive classification of the MMGCN across all categories varies between 97.33% and 99.13%. The maximum confusion is for classifying calcification-malignancy, where 1% is wrongly classified as benign and less than 2% among the mass and normal classes. The mass malignant and calcification benign have the same confusion rates, while the minimum confusion rate is less than 1% for classifying the normal segments wrongly.

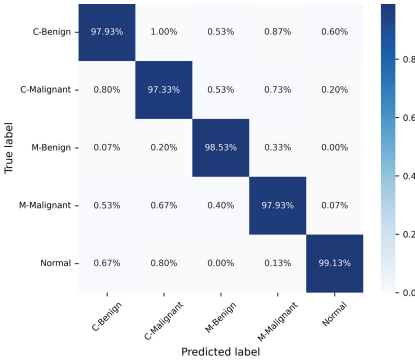


Fig. 5. The confusion-matrix of the true and predicted classes.

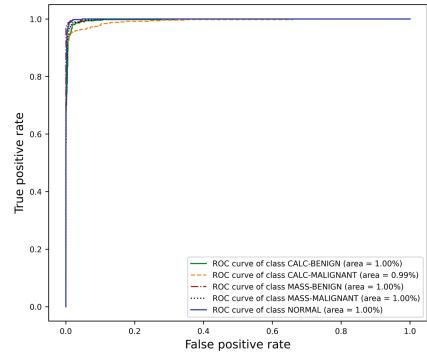


Fig. 6. The AUC-ROC curve for each individual mammogram class.

To analyse the ability of SSL-MGCN to distinguish between the five classes, we use the ROC curve evaluation metric. This curve plots the probability of each class’s true-positive versus false-positive rates, considering one-to-all classes. Figure 5 shows the ROC curves of all five classes, which demonstrate the model’s effectiveness in classifying each class correctly with almost 100%, albeit the model can misclassify the malignant calcification anomalies by 1%.

3.4 Compared Methods

To demonstrate the advantage of modelling the segments of the mammogram images into a multigraph and integrating a self-supervised pre-training encoder, we compare our frameworks, i.e., MMGCN and SSL-MMGCN, to the current state-of-the-art methods in [33, 34, 36, 37]. Table 1 lists the performance of each method as presented in their papers, including the AUC accuracy, the considered abnormalities, and the classification task. Furthermore, for fair analyses, we consider the train-test ratio for each experiment setup.

Similar to our framework, only the work in [37] adopted a whole mammogram multi-classification method to detect both the calcification and mass abnormalities. Further, the other methods are limited to the classification of only one type of abnormality, the mass abnormalities as in [33] and the calcification abnormalities as in [34].

Compared to our framework, which enhances the graph embedding by integrating a self-supervised encoder and reduces the learning rates by adopting a semi-supervised graph-based model, other methods only integrate fully supervised methods. As a result, our method requires less annotated data for training, 40% for SSL-MMGCN, and 60% for MMGCN, compared to 80% in the other methods. However, MMGCN and SSL-MMGCN outperform these methods, particularly the framework proposed in [33] and [37], which use the same dataset, i.e., DDSM, for evaluation.

Table 1. Breast cancer classification performance in AUC score for SSL-MMGCN and MMGCN and some state-of-art methods. Multi-Task Classification: (Normal, Mass-Malignant, Mass-Benign, Calcification-Malignant, Calcification-Benign). Binary-Task Classification: (Normal, Abnormal)

Methodology	Dataset	AUC	Task	Mass/Calc.	Train : Test
Li H. et al. (2021) [33]	INbreast	0.93	binary	✓/✗	80% : 20%
Li H. et al. (2021) [33]	DDSM	0.85	binary	✓/✗	80% : 20%
Hao Du et al. (2021) [34]	private	0.81	multi	✗/✓	80% : 20%
Le et al. (2019) [37]	DDSM	0.8	multi	✓/✓	80% : 20%
MMGCN	DDSM	0.98	multi	✓/✓	60% : 40%
SSL-MMGCN	DDSM	0.97	multi	✓/✓	40% : 60%

Noticeably, the work in [33] has better AUC accuracy when evaluated on the INbreast dataset rather than the DDSM dataset we use in our experiments. That

is because of the better resolution quality of the full-field digital mammogram images in INbreast than the digitised images in DDSM. So, in extended experiments, we will evaluate our framework on the most recent digital mammogram dataset, which can result in even better AUC accuracy.

4 Conclusion

This work adopts a graph-based deep learning framework that enables semi-supervised and self-supervised machine learning approaches to perform efficient breast cancer classification using mammogram data. The framework models the heterogeneous high-level texture features and their critical relations and spatial details inherent to mammograms. MMG maps each mammogram to a graph and later combines these graphs into a multi-graph to improve the representation of the relations and features in a mammogram. To perform node-level classification, we have exploited the benefits of MMGCN and SSL-MMGCN models where pre-trained self-supervised SSL-MMGCN demonstrates significant improvement in learning with limited labeled data. Self-supervision significantly improves the training time in the downstream process. Results show that with sufficient labeled data, i.e., 40% or more, the MMGCN model shows accelerated learning capacity and better multi-classification sensitivity.

Experiments results reveal the proposed graph-based framework has excellent AUC classification performance of 0.97 for the SSL-MMGCN and 0.98 for the MMGCN and outperforms state-of-the-art works for breast cancer diagnosis, including Li H. et al. [33], Hao Du et al. [34] and Le et al. [37].

In future works, we will consider the augmentation of other convolutional neural networks to encode mammogram features efficiently to accelerate accurate breast cancer diagnosis with possible consideration in clinical trials.

References

1. Bataille, V., et al.: Nevus size and number are associated with telomere length and represent potential markers of a decreased senescence in vivo. *Cancer Epidemiol. Prev. Biomark.* **16**(7), 1499–1502 (2007)
2. Kösters, J.P., Gøtzsche, P.C.: Regular self-examination or clinical examination for early detection of breast cancer. *Cochrane Database Syst. Rev.* (2) (2003)
3. Mordang, J.J., et al.: The importance of early detection of calcifications associated with breast cancer in screening. *Breast Cancer Res. Treat.* **167**(2), 451–458 (2018)
4. Hofvind, S., Iversen, B.F., Eriksen, L., Styr, B.M., Kjelleevold, K., Kurz, K.D.: Mammographic morphology and distribution of calcifications in ductal carcinoma in situ diagnosed in organized screening. *Acta Radiol.* **52**(5), 481–487 (2011)
5. Nalawade, Y.V.: Evaluation of breast calcifications. *Indian J. Radiol. Imaging* **19**(4), 282–286 (2009)
6. Bejnordi, B.E., et al.: Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA* **318**(22), 2199–2210 (2017)

7. Henriksen, E.L., Carlsen, J.F., Vejborg, I.M., Nielsen, M.B., Lauridsen, C.A.: The efficacy of using computer-aided detection (CAD) for detection of breast cancer in mammography screening: a systematic review. *Acta Radiologica* **60**(1), 13–18 (2019)
8. Katalinic, A., Bartel, C., Raspe, H., Schreer, I.: Beyond mammography screening: quality assurance in breast cancer diagnosis (The QuaMaDi Project). *Br. J. Cancer* **96**(1), 157–161 (2007)
9. Abdelhafiz, D., Yang, C., Ammar, R., Nabavi, S.: Deep convolutional neural networks for mammography: advances, challenges and applications. *BMC Bioinform.* **20**(11), 1–20 (2019)
10. Guan, S., Loew, M.: Breast cancer detection using transfer learning in convolutional neural networks. In: *Conference on AIPR 2017 IEEE Applied Imagery Pattern Recognition Workshop*, pp. 1–8 (2017)
11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint* (2014). [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
12. Shen, L., Margolies, L.R., Rothstein, J.H., Fluder, E., McBride, R., Sieh, W.: Deep learning to improve breast cancer detection on screening mammography. *Sci. Rep.* **9**(1), 1–12 (2019)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
14. Salama, W.M., Wessam, M., Aly, M.H.: Deep learning in mammography images segmentation and classification: automated CNN approach. *Alex. Eng. J.* **60**(5), 4701–4709 (2021)
15. Ballester, P., Araujo, R.M.: On the performance of GoogLeNet and AlexNet applied to sketches. In: *Thirtieth AAAI Conference on Artificial Intelligence* (2016)
16. Alom, M.Z., et al.: The history began from alexnet: a comprehensive survey on deep learning approaches. *arXiv preprint* (2018). [arXiv:1803.01164](https://arxiv.org/abs/1803.01164)
17. Du, H., Yao, M.M.S., Chen, L., Chan, W.P., Feng, M.: Multi-task Graph Convolutional Neural Network for Calcification Morphology and Distribution Analysis in Mammograms. *arXiv preprint* (2021). [arXiv:2105.06822](https://arxiv.org/abs/2105.06822)
18. Zhang, Z., Lee, W.S.: Deep graphical feature learning for the feature matching problem. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5087–5096 (2019)
19. Gallego-Ortiz, C., Martel, A.L.: A graph-based lesion characterization and deep embedding approach for improved computer-aided diagnosis of nonmass breast MRI lesions. *Med. Image Anal.* **51**, 116–124 (2019)
20. Kipf, T.N., Thomas, N., Welling, M.: Semi-supervised classification with graph convolutional networks. *arXiv preprint* (2016). [arXiv:1609.02907](https://arxiv.org/abs/1609.02907)
21. Du, H., Feng, J., Feng, M.: Zoom in to where it matters: a hierarchical graph based model for mammogram analysis. *arXiv preprint* (2019). [arXiv:1912.07517](https://arxiv.org/abs/1912.07517)
22. Zhang, Y.D., Satapathy, S.C., Guttery, D.S., Górriz, J.M., Wang, S.H.: Improved breast cancer classification through combining graph convolutional network and convolutional neural network. *Inf. Process. Manag.* **58**(2), 102439 (2021)
23. Wang, J., Chen, R.J., Lu, M.Y., Baras, A., Mahmood, F.: Weakly supervised prostate TMA classification via graph convolutional networks. In: *Conference on ISBI 2020 IEEE 17th International Symposium on Biomedical Imaging*, pp. 239–243 (2020)
24. Özen, Y.: Self-supervised representation learning with graph neural networks for region of interest analysis in breast histopathology. Doctoral dissertation, Bilkent University (2020)

25. Ma, J., Li, X., Li, H., Wang, R., Menze, B., Zheng, W.S.: Cross-view relation networks for mammogram mass detection. In: Conference on ICPR 2020 25th International Conference on Pattern Recognition, pp. 8632–8638 (2021)
26. Yang, Z., et al.: MommiNet-v2: mammographic multi-view mass identification networks. *Med. Image Anal.* **73**, 102204 (2021)
27. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
28. Dehak, N., Dehak, R., Glass, J.R., Reynolds, D.A., Kenny, P.: Cosine similarity scoring without score normalization techniques. In: *Odyssey*, p. 15 (2010)
29. Mondal, A.K., Jain, V., Siddiqi, K.: Mini-batch graphs for robust image classification. *arXiv preprint* (2021). [arXiv:2105.03237](https://arxiv.org/abs/2105.03237)
30. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**(11) (2008)
31. Lee, R.S., Gimenez, F., Hoogi, A., Miyake, K.K., Gorovoy, M., Rubin, D.L.: A curated mammography data set for use in computer-aided detection and diagnosis research. *Sci. Data* **4**(1), 1–9 (2017)
32. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. *arXiv preprint* (2016). [arXiv:1609.02907](https://arxiv.org/abs/1609.02907)
33. Li, H., Chen, D., Nailon, W.H., Davies, M.E., Laurensen, D.I.: Dual convolutional neural networks for breast mass segmentation and diagnosis in mammography. *IEEE Trans. Med. Imaging* **41**(1), 3–13 (2021)
34. Du, H., Yao, M.M.S., Chen, L., Chan, W.P., Feng, M.: Multi-task Graph Convolutional Neural Network for Calcification Morphology and Distribution Analysis in Mammograms. *arXiv preprint*, vol. 14 (2021). [arXiv:2105.06822](https://arxiv.org/abs/2105.06822)
35. Dhungel, N., Carneiro, G., Bradley, A.P.: The automated learning of deep features for breast mass classification from mammograms. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) *MICCAI 2016*. LNCS, vol. 9901, pp. 106–114. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_13
36. Al-Antari, M.A., Al-Masni, M.A., Kim, T.S.: Deep learning computer-aided diagnosis for breast lesion in digital mammogram. *Deep Learn. Med. Image Anal.* 59–72 (2020)
37. Le, T.L.T., Thome, N., Bernard, S., Bismuth, V., Patoureaux, F.: Multitask classification and segmentation for cancer diagnosis in mammography. *arXiv preprint* (2019). [arXiv:1909.05397](https://arxiv.org/abs/1909.05397)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





A Transformer Architecture for Online Gesture Recognition of Mathematical Expressions

Mirco Ramo^{1,2(✉)} and Guénolé C. M. Silvestre²

¹ Dip. Ingegneria dell'Informazione, University of Pisa, Pisa, Italy
`mirco.ramo@ucdconnect.ie`

² School of Computer Science, University College Dublin, Dublin, Ireland
`guenole.silvestre@ucd.ie`

Abstract. The Transformer architecture is shown to provide a powerful framework as an end-to-end model for building expression trees from online handwritten gestures corresponding to glyph strokes. In particular, the attention mechanism was successfully used to encode, learn and enforce the underlying syntax of expressions creating latent representations that are correctly decoded to the exact mathematical expression tree, providing robustness to ablated inputs and unseen glyphs. For the first time, the encoder is fed with spatio-temporal data tokens potentially forming an infinitely large vocabulary, which finds applications beyond that of online gesture recognition. A new supervised dataset of online handwriting gestures is provided for training models on generic handwriting recognition tasks and a new metric is proposed for the evaluation of the syntactic correctness of the output expression trees. A small Transformer model suitable for edge inference was successfully trained to an average normalised Levenshtein accuracy of 94%, resulting in valid postfix RPN tree representation for 94% of predictions.

Keywords: Online gesture recognition · Transformer · Multilevel segmentation · Expression tree · Transfer learning · RPN

1 Introduction

Modern edge communicating devices are built around touch-sensitive display panels equipped with handwriting recognition systems. These systems are of great assistance eschewing the need for structured UIs such as virtual keyboards that are often slow and error-prone while also distant to the natural handwriting experience with pens.

In this context, online recognition of glyphs (as opposed to offline that takes a graphical image representation as input) refers to the problem of mapping spatio-temporal samplings of user gestures corresponding to handwritten text into a symbolic representation. Each 3-dimensional sample individuates a touch. A coherent and consecutive sequence of touches defines a stroke that can be combined to form glyphs. Glyphs correspond to characters or symbols encoded

Table 1. Terminology

Term	Definition
Touch/point	$(x, y, [t])$ tuple of finger location on touch panel sampled at time t
Stroke	Sequence of points where finger consecutively touches the panel
Glyph	List of one or more strokes individuating an element in the vocabulary
Numeral/operand	Ordered list of glyphs corresponding to digits $\{0, \dots, 9\}$ and comprising at most one decimal notation mark $\{.\}$ – note: a numeral always evaluates in a finite rational numeric value
Symbol	One of the 4 operators $\{+, -, \times, \div\} \cup \{=, (,)\}$

in a language vocabulary. In this work, we will consider the online input of mathematical arithmetic expressions as a formally correct sequence of gestures of numerals, operators and symbols. Note that some numerals and operators may require more than one stroke to be represented as depicted in Fig. 1. Table 1 formalizes the terminology adopted in this work.

Gesture recognition applications must solve several problems at once, namely: i) feature extraction in a multi-dimensional spatio-temporal space, ii) segmentation of stroke sequences into glyph items, iii) glyph segmentation with the aim of numeral recognition, and iv) the encoding of expression rules and patterns to form a correct symbolic output. An example of an online gesture sequence is shown in Fig. 1.

With mathematical expressions, users often wish to go beyond the mere recognition of glyphs and hope for additional tasks to be performed such as automatic evaluation, step-by-step simplification or listing of equivalent forms. The Expression Tree (ExpTree) formalism [1] was introduced to represent mathematical expressions as binary trees and consequently resolve all equivalent forms to some unique representation, scheduling its evaluation by transforming an input symbol list into a computation graph. In particular, a post-ordered traverse of tree generates the Reverse Polish Notation (RPN) using postfix notation, a unique representation that postpones operators, crushing the need for brackets.

Main Contributions

- (i) We propose a new dataset for handwritten expressions (cf. Sect. 3) obtained from several hundred users and suitable for a wide range of supervised and unsupervised machine learning applications.
- (ii) We study the ability of an attention mechanism to learn and represent implicit structures of spatio-temporal gesture data, even when the underlying syntax is not enforced (in the loss computation or model architecture).
- (iii) We prove the power of Transformers not only as language models but also as a solution to several sequence mapping tasks, demonstrating transfer

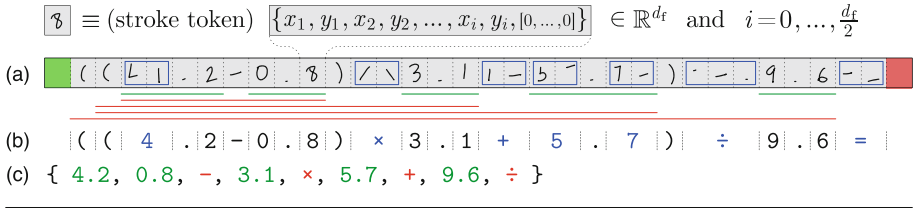


Fig. 1. Online gesture example of a stroke sequence (a) for the mathematical expression (b) and its corresponding RPN expression tree (c). Each cell in (a) depicts the linear interpolation of spatio-temporal points that forms an input token. Green and red cells denote the $\langle \text{bos} \rangle$ and $\langle \text{eos} \rangle$ respectively. Glyph segmentation, numeral segmentation and RPN ExpTree parsing are colour coded with blue, green and red respectively. (Color figure online)

learning behaviours of the encoder on unseen glyphs from online gesture input¹.

- (iv) We propose a small footprint² topology for end-to-end online mathematical expression recognition and ExpTree generation, with fast optimisation, very high accuracy and suitable for edge inference.
- (v) We test the model robustness on ablated input, showing its ability to generate compliant RPN expressions even in case of missing strokes.
- (vi) We show the multi-level segmentation capability of the attention mechanism, highlighting the correlation between syntactically correct predictions and explainability in cross-attention visualisation.

2 Related Work

The field of Handwriting Text Recognition (HTR) consists on a set of techniques and algorithms that aim at generating text directly from handwritten inputs. Most HTR systems [2] work on offline data due to dataset availability [3]. With the current popularity of the attention mechanism [4,5], the field remains in constant development. However, as noted in [6], the temporal dimension provides some valuable additional information that may simplify stroke segmentation and avoid recourse to complicated regression strategies such as text-line segmentation [7]. As a result, online methods may expect superior performance over offline counterparts as reported in a 2014 extensive survey of online HTR methods [6]. Further progress has since been observed, with much effort and resources employed on improving existing techniques [8,9].

¹ Encoder with frozen parameters (pre-trained on digit-agnostic datasets) subsequently used on a new task, taking token input from spatio-temporal sequences in a potentially infinitely large vocabulary.

² Despite its small footprint, model can perform the tasks of glyph segmentation, numeral segmentation, character recognition and tree building at remarkable performance levels, learning efficiently the input/output mapping.

In this context, Handwriting Digit Recognition (HDR) remains a popular HTR sub-problem still actively researched using both offline [10] and online methods [11]. In particular Handwritten Mathematical Expression Recognition (HMER) consists in the generation of mathematical expressions using formal syntaxes such as \LaTeX . State-of-the-art HMER models have reached impressive levels of accuracy, particularly when exploiting attention [12] and combining potentialities of online and offline data [13]. However, although predictions are mostly correct, these models fail to learn the intrinsic structure of the mathematical expression. In contrast, learning a tree representation provides a more natural form [14] and can be achieved with an RNN encoder and a HMER tree decoder to explicitly represent the tree formalism.

We propose to push this challenge further, leaving the task of learning implicitly the RPN syntax to the model, and doing this by relying on the attention mechanism embedded in a Transformer framework [15]. This provides a powerful sequence mapping architecture entirely based on the attention mechanism [4], eschewing recourse to recurrent or convolution layers, hence allowing for significant parallelisation and unattenuated gradient flow. This topology currently stands as the state-of-the-art on almost all NLP tasks [16–18], but also on a wider and more generic group of sequence transduction problems [19–23]. The Transformer popularity saw many experiments revisiting its design with several optimized architectures being proposed [24–27]. However very few are capable of clearly outperforming the original topology. As a result this work will follow the seminal Transformer proposal of [15].

3 Dataset

An important contribution of this work is that of an online gesture dataset of mathematical expressions suitable for investigating several tasks such as Handwriting Character Recognition (HCR), HDR or HMER, but also touch, stroke or glyph segmentation, automatic result computation, unsupervised generation or eventually, ExpTree building. Our handwritten database is presented as a coherent collection of tables composing a SQL Schema with spatio-temporal data for arabic numerals [11] and mathematical symbols, collected from volunteers writing on touch panels. This stage saw the contribution of 455 subjects for a total of 21 752 labelled glyphs composed by 27 477 strokes, thus over 700 thousand touches. The dataset can be used at different levels of granularity, namely *touch*, *stroke* and *glyph*.

Subjects have been split into training, validation and test sets (60/20/20 proportions) such that models were tested on unseen handwriting styles to ensure accurate estimation of the generalisation power. In addition, strokes were also randomly augmented and composed to form expressions.

An expression (E) is defined as a bounded sequence of numerals (N) and operators ($\odot \in \{+, -, \times, \div\}$). The generation of expressions is carried out according to the following grammar:

1. an expression can be a numeral: $E \rightarrow N$

2. an expression can be a binary operation: $E \rightarrow E \odot E$
3. an expression can be a binary operation between brackets: $E \rightarrow (E \odot E)$

As a supplementary rule, every expression must end with the ‘=’ symbol. For each expression, we provide 3 ground truth labels (namely ASCII text, RPN tree and numerical evaluation), for a total of 240 000 samples split as specified above. In this work, we report results at the stroke level, leaving to the model the burden of glyph segmentation.

4 Transformer Architecture and Experimental Details

Our model leverages the original Transformer architecture [15]. However crucial modifications are introduced to work with spatio-temporal data. Given some input sequence, $X \in \mathbb{R}^{d_t \times n}$, of n stroke tokens defined as interleaved spatio-temporal data with zero-padding of fixed-length d_t (maximum of 64 (x, y) touch samples per stroke, appropriately $\langle \text{bos} \rangle$ prefixed, $\langle \text{eos} \rangle$ suffixed and $\langle \text{pad} \rangle$ padded), a mask M_x is computed to ensure encoder’s attention is only paid on valid online data tokens.

As the input is composed of spatio-temporal information corresponding to touches, each encoder token embeds a stroke as d_t scalars (cf. Fig. 1) resulting in the identification within a potentially unbounded input vocabulary and therefore eschewing any form of embedding.

Positional encoding provides a strategy to embed the positional information of input tokens in the encoder, a necessary operation since the attention mechanism has no built-in concept of sequentiality. Frequency modulation is proposed in [15]. However, since we observed no performance gain with such a strategy, we use a learnable 1D embedding based on the incremental index of the token. Stroke positions are encoded in $P_x \in \mathbb{R}^{d_t \times n}$.

The encoder is trained to learn some latent sequence representation $Z = \text{Enc}(X + \alpha P_x, M_x) \in \mathbb{R}^{d_a \times d_h \times n}$ where α is a scaling factor blending the input data and positional information, d_a the number of attention heads and d_h the hidden state dimension of the attention heads. The encoder consists of a stack of l_e identical multi-head vanilla self-attention layers and a positional feed-forward network of dimension d_p . Each layer is followed by a residual connection before layer-normalisation.

In this work, we explored the transfer learning capabilities of the encoder that was never trained from scratch but relied on an optimised snapshot, pre-trained in conjunction with a language modelling decoder using a large corpus of English sentences [28] that contained almost no digits and arithmetic operators (classified as $\langle \text{unk} \rangle$ tokens). This transfer learning strategy resulted in considerable speed-up during training and model optimisation. We use a frozen encoder with Θ_e parameters as a feature extractor on this new domain.

The decoder generates a causal sequence of tokens in an auto-regressive manner given some vocabulary and relative token encoding. It is initialised with the

Table 2. Expression recognition, model hyper-parameters and dataset configuration. Performance is reported in term of Cross-Entropy Loss (XEL) and normalised Levenshtein distance Accuracy (LA). Model v_4 trained on larger expressions using 4 Heads (H) in a 4 Layer (L) decoder performs best.

Model	Data size	Bracket	Enc* (l_e, d_a)	Dec (l_d, d_a)	XEL	LA (%)
v_1	10k	No	5L, 4H	2L, 4H	0.76	71.3
v_2	240k	No	5L, 4H	4L, 4H	0.43	84.8
v_3	240k	No	5L, 4H	4L, 2H	0.43	84.8
v_4	240k	Yes	5L, 4H	4L, 4H	0.40	84.9

*Encoder: using transfer learning with frozen (untrained) parameters

$\langle \text{bos} \rangle$ token and iteratively outputs a new token using greedy sampling of the decoder’s softmax output until the $\langle \text{eos} \rangle$ token is predicted or the maximum sequence length, m , is reached. The decoder also consists of l_d identical layers, each composed by: i) a masked self-attention layer that prevents the decoder from peeking at the subsequent tokens, ii) a cross-attention layer that attends over the encoder output Z to generate predictions, and iii) a feed-forward layer as in the encoder but of dimension $3d_p$.

At each step, the decoder’s input is an auto-regressive sequence of tokens mapped into an embedding layer with positional encoding, and used to predict the next token of the output sequence. All Θ_d parameters of the decoder were trained from some randomly initialised state.

Experimental details: models were all configured with $d_f = d_a \times d_h = d_p = 128$. For v_{1-5} , $n = 2m = 24$. For v_{10-11} , $n = 2m = 48$. The encoder has $\Theta_e = 523\,520$ parameters and decoder has $\Theta_d = 934\,136$ parameters. Models were trained on Nvidia TitanX GPUs³, for a maximum of 200 epochs, using cross-entropy loss and Adam optimiser with a decay schedule (initial learning rate of 8×10^{-4} and halving every 30 epochs).

5 Experimental Results

A series of experiments were carried out to investigate two different problems, namely: (1) expression recognition in glyph sequences and (2) ExpTree recognition in RPN forms. The first task involves the recognition of a sequence of glyphs composing an arithmetic expression from stroke input as time series. The second task requires further understanding of symbolic syntax and semantics through the construction of an ExpTree using postfix notation.

Models are evaluated using a number of performance metrics on the test sets and results are reported in terms of: (a) Cross-Entropy Loss (XEL), (b) normalised Levenshtein distance [29] Accuracy (LA), (c) Character Error Rate (CER), and where applicable (d) RPN Accuracy Range (cf. Sect. 5.2). The LA and CER are both accuracy metrics based on edit distance.

³ Nvidia is acknowledged for the donation of GPUs.

Table 3. ExpTree recognition for various model hyper-parameters. Performance is reported in term of Cross-Entropy Loss (XEL), normalised Levenshtein distance Accuracy (LA), Character Error Rate (CER), and RPN Accuracy Range (RAR). Models trained on 240k expression datasets. Fine-tuned model v_{11} with $\langle \text{eon} \rangle$ for numeral segmentation provides best performance.

Name	Enc (l_e, d_a)	Dec (l_d, d_a)	XEL	LA(%)	CER	RAR (%)
v_5	5L, 4H*	4L, 4H [†]	0.46	83.7	0.19	[91.8, 91.8]
v_{10}	5L, 4H*	4L, 4H [†]	0.34	87.4	0.14	[92.4, 93.2]
v_{11}	5L, 4H [†]	4L, 4H [†]	0.24	93.7	0.07	[93.3, 94.0]

*Frozen parameters

[†]Fined-Tuned parameters using transfer learning

[‡]Trained from scratch from some randomly initialised state

5.1 Expression Recognition

In this set of experiments, models v_{1-3} are trained to output glyph sequence of simple arithmetic expressions in the absence of brackets while model v_4 adds groups of terms with brackets. Table 2 summarises training datasets, model hyper-parameter configuration and performance evaluation in this experiment.

We observe that there are no clear benefits in increasing the number of decoder heads in the absence of brackets (models v_{2-3}). However, despite an increase of vocabulary size and, in principle, also some decoding complexity, the addition of brackets resulted in better performance as seen in model v_4 . This model is capable of learning some non-trivial valuable syntax rules such as number of ‘(’ should match that of ‘)’, or an operator can never precede a ‘)’.

5.2 Expression Tree Recognition

The ExpTree recognition task requires of an additional step to glyph recognition with the construction of an RPN form. In this set of experiments, model performance is also evaluated in terms of CER and RPN Accuracy Range (RAR) defined as the range $[1 - V_\ell^{\max}, 1 - V_\ell^{\min}]$, where V_ℓ stands for violation loss. If v_i denotes the count of violations in the i -th expression, $V_\ell^{\min} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{v_i > 0}$ and $V_\ell^{\max} = \frac{1}{N} \sum_{i=1}^N v_i$, where N is the test set cardinality. Referring to the standard infix to postfix conversion algorithm in [1], a violation occurs every time the stack is in an inconsistent state while conversion is performed.

This does not required the initialisation of stack operations to be determined. Instead one can linearly scan the output using a counter, incrementing its value for a push, decrementing it for a pop. Counter value should be 1 at the end and never become negative. Adding the number of times a negative value is observed to the absolute value of the final counter minus 1 defines the number of violations.

Table 3 summarises experimental results on ExpTree predictions. Models $v_{5, 10-11}$ were trained on the same dataset size as v_{2-4} (240k expressions), with

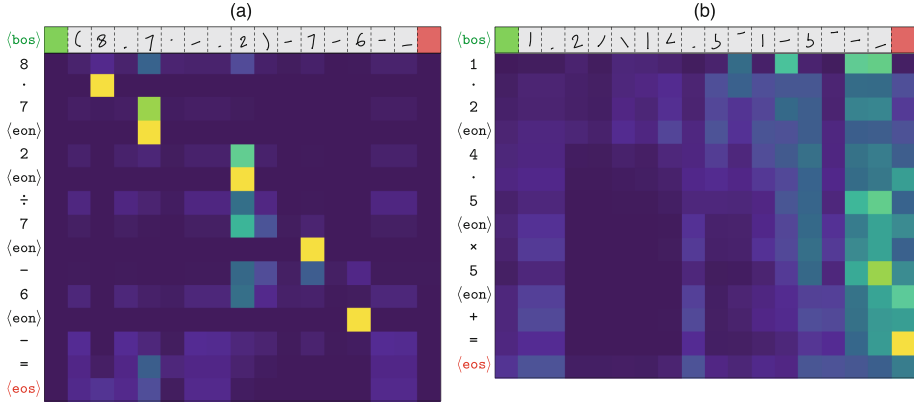


Fig. 2. Cross-attention plots. In (a), output tokens ‘.’ (decimal mark) and $\langle \text{eon} \rangle$ (end-of-numeral) can be seen tracking the previous digit; in (b), output token ‘=’ is attending token $\langle \text{eos} \rangle$.

the possible inclusion of brackets. The v_5 model training dataset further constrained numerals to contain at most one decimal digit. This restriction was lifted in training sets associated with models v_{10-11} . As a result, an end-of-numeral token, $\langle \text{eon} \rangle$, was added to the decoder’s output vocabulary for learning an additional numeral segmentation task of RPN forms.

With the same hyper-parameter configuration of Table 2, an expected degradation in performance is observed for model v_5 on this more complicated task. The addition of the $\langle \text{eon} \rangle$ token in model v_{10} showed some significant improvement in accuracy, outperforming our best results for simple expression glyph recognition. Despite the use of a larger vocabulary size for the decoder’s output, the addition of a specific token to model explicitly the language semantic of numerals is observed to yield higher accuracy once again. The new token forces the network to learn a pattern resulting in better numeral segmentation and improved performance.

In Sect. 4 we proposed to test the transfer learning capabilities of the encoder, using frozen parameters on a new domain. Excellent results have been observed, demonstrating the encoder’s ability to correctly segment and combine strokes generating latent representations that are generic enough to be valuable for any downstream tasks even when used with completely different output vocabulary.

However, further improvement can be reached with fine-tuning of all parameters as observed with model v_{11} that leveraged frozen encoder weights of model v_{10} , introducing the concepts of digits or operators for the first time. Final model achieves 94% on the Normalised Levenshtein Accuracy, with a Character Error Rate lower than 7%, generating on average 94% of strings compliant to the RPN, while mean number of violations per output expression is only 0.067.

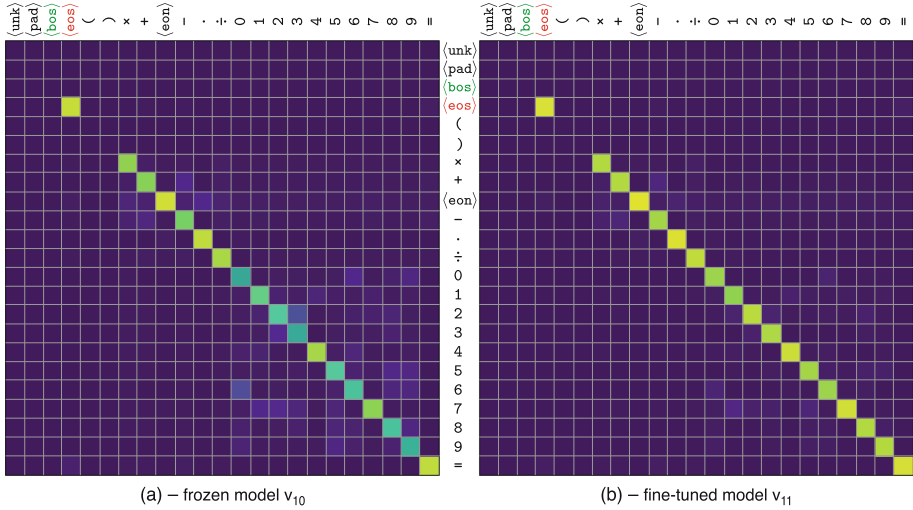


Fig. 3. Softmax distribution mean of the decoder’s output predictions showing the probability mass for all token pairs. Frozen model v_{10} in plot (a) reveals decoding errors caused by confusion between digits 2 and 3, and also between operators. Model v_{11} in plot (b) shows that fined-tuning on all glyphs reduces confusion dramatically.

6 Attention Visualisation and Output Distributions

Visualisation of attention mechanisms provides some interesting insights in the learning process. Figure 2a depicts the cross-attention weights that the decoder puts over the encoder’s output. It shows that head 1 of layer 1 is responsible for numeral segmentation. For every $\langle \text{eon} \rangle$ or decimal mark tokens, this head has learned to attend over the stroke of the previous digit. In Fig. 2b, head 4 of layer 3 attends over the $\langle \text{eos} \rangle$ token while predicting the ‘=’ token demonstrating that the model has successfully learned the syntax rule ‘every expression must end with ‘=’ symbol’.

Figure 3 shows the confusion matrix over the decoder’s vocabulary for the average probability distribution of the output softmax. This provides some insight into model mispredictions leading to errors. In Fig. 3a, model v_{10} leveraged a frozen encoder pre-trained on a completely different output vocabulary with no digits and operators. The model confuses ‘2’ with ‘3’ and, to a lesser extent, operator ‘-’ with ‘+’ since the latter is often written with an horizontal stroke. Figure 3b shows that fine-tuning the encoder in model v_{11} results in better performance and improved diagonality, which also justifies the greedy decoding strategy used in our decoder.

Table 4. Model robustness: ablation experiments with input strokes elided from input and corresponding to the equal sign in rows 1–3, a closing bracket in row 4 and an operator in rows 5–6. Metric is the Levenshtein Distance (LD).

Model	Input (X), Ground truth (Y) & Model inference (\hat{Y})	LD
v_4	$X = \text{[green]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[red]}$ $Y = \{\langle \text{bos} \rangle 4 \times 6 \langle \text{eos} \rangle\}$ $\hat{Y} = \{\langle \text{bos} \rangle 4 \times 6 = \langle \text{eos} \rangle\}$	1
v_{10}	$X = \text{[green]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[red]}$ $Y = \{\langle \text{bos} \rangle 6.3 \langle \text{eon} \rangle 3 \langle \text{eon} \rangle 2 \langle \text{eon} \rangle + - 5 \langle \text{eon} \rangle + \langle \text{eos} \rangle\}$ $\hat{Y} = \{\langle \text{bos} \rangle 6.2 \langle \text{eon} \rangle 3 \langle \text{eon} \rangle 2 \langle \text{eon} \rangle + - 5 \langle \text{eon} \rangle + = \langle \text{eos} \rangle\}$	2
v_{11}	$X = \text{[green]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[red]}$ $Y = \{\langle \text{bos} \rangle 9 \langle \text{eon} \rangle 7 \langle \text{eon} \rangle 3 \langle \text{eon} \rangle \div 2 \langle \text{eon} \rangle - + \langle \text{eos} \rangle\}$ $\hat{Y} = \{\langle \text{bos} \rangle 9 \langle \text{eon} \rangle 7 \langle \text{eon} \rangle 3 \langle \text{eon} \rangle \div 2 \langle \text{eon} \rangle - + = \langle \text{eos} \rangle\}$	1
v_4	$X = \text{[green]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[red]}$ $Y = \{\langle \text{bos} \rangle 7.4 \times (3.8 + 9 = \langle \text{eos} \rangle)\}$ $\hat{Y} = \{\langle \text{bos} \rangle 7.4 \times (3.8 + 9) = \langle \text{eos} \rangle\}$	1
v_{10}	$X = \text{[green]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[red]}$ $Y = \{\langle \text{bos} \rangle 6.3 \langle \text{eon} \rangle 3 \langle \text{eon} \rangle 2 \langle \text{eon} \rangle + - 5 \langle \text{eon} \rangle = \langle \text{eos} \rangle\}$ $\hat{Y} = \{\langle \text{bos} \rangle 6.2 \langle \text{eon} \rangle 3 \langle \text{eon} \rangle 2 \langle \text{eon} \rangle + - 5 \langle \text{eon} \rangle + = \langle \text{eos} \rangle\}$	2
v_{11}	$X = \text{[green]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[grey]} \text{[red]}$ $Y = \{\langle \text{bos} \rangle 9 \langle \text{eon} \rangle 3.5 \langle \text{eon} \rangle - 7 \langle \text{eon} \rangle + 5 \langle \text{eon} \rangle = \langle \text{eos} \rangle\}$ $\hat{Y} = \{\langle \text{bos} \rangle 9 \langle \text{eon} \rangle 3.5 \langle \text{eon} \rangle - 7 \langle \text{eon} \rangle + 1 \langle \text{eon} \rangle - = \langle \text{eos} \rangle\}$	2

7 Model Robustness

Model robustness is investigated by means of ablation studies; strokes are removed from the input sequence to observe the model’s ability to enforce domain rules even when it is fed with incorrect expressions.

Equal sign ablation: in our dataset, every expression to be considered syntactically correct must end with ‘=’. The learning of this rule is assessed by observing the inference results of models $v_{4,10-11}$ when strokes representing the equal sign are omitted in the encoder’s input. All three models are able to make the correct inference, inserting the missing ‘=’ in decoder’s output as shown in rows 1–3 of Table 4.

Closing bracket ablation: in any correct plain expressions, the number of ‘(’ should match that of ‘)’. This syntactic rule is investigated in model v_4

that was trained to recognise glyphs of an expression (not possible with models v_{10-11} as RPN forms eschew the use of brackets). When the stroke of a closing bracket was removed from the encoder’s input, the model acknowledges the input error and inserts the missing bracket in the output as shown in row 4 of Table 4. Of course, the exact position is not always guessed correctly, but the symbol is predicted so that to ensure syntax correctness of the output.

Operator ablation is investigated on models v_{10-11} , where an operator’s strokes is removed from the input as shown in rows 5–6 of Table 4. To ensure ExpTree correctness when using postfix notation, an output expression must be terminated by an operator and its total number of operators always be a unit lower than the cardinality of operands. Both models appear to have learned this rule and are able to infer the presence of additional operator at the end (actual operator can only be guessed).

8 Conclusion

This work proposed a Transformer network for mathematical expression tree building from online input gesture data corresponding to handwritten strokes of digits and mathematical symbols. The encoder’s input was modified to receive spatio-temporal data as real-valued tokens. It can directly operate at stroke level without the need for mapping on a fixed input vocabulary. Model can predict ExpTrees by handling internally the multi-level segmentation of inputs (at glyph and numeral levels) and also understanding and learning how to represent and enforce syntactic and semantic rules of data. In addition, index positional encoding was shown to be as effective as cosine modulation yet standing as a simpler and more natural encoding for the position information. The Transformer’s ability to generate complex representations and learn non-trivial input/output mapping between sequences is well known [16, 19]. However the challenge was further pushed in this work with no ad hoc solutions to represent syntax or semantic rules and the absence of an engineered loss computation and model architecture. In addition, the encoder was trained on a completely different domain [28] and used as a frozen feature extractor in most experiments. Such transfer learning capabilities suggest that the encoder can create general latent representations suitable for problems of different nature, reducing the overall number of model parameters. The objective of this work is not so much to push out some state-of-the-art model but rather to state some important considerations that may be the starting points for future works in language modelling. Neural Machine Translation may be extended in this way to online data at different granularity levels, with no need for separate input segmentation or complex positional embeddings. Finally, pre-trained encoders could be effectively leveraged with transfer learning on different domains without fine-tuning or explicit domain adaptation, accelerating training for new problem classes where computational power/time or dataset size is limited.

References

1. Gries, D.: *Compiler Construction for Digital Computers*. Wiley, New York (1971)
2. Plamondon, R., Srihari, S.: Online and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(1), 63–84 (2000)
3. Sinwar, D., Dhaka, V.S., Pradhan, N., et al.: Offline script recognition from handwritten and printed multilingual documents: a survey. *Int. J. Doc. Anal. Recogn. (IJDAR)* **24**(1), 97–121 (2021)
4. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: *International Conference on Learning Representations (ICLR)* (2015)
5. Poulos, J., Valle, R.: Character-based handwritten text transcription with attention networks. *Neural Comput. Appl.* **33**(16), 10563–10573 (2021). <https://doi.org/10.1007/s00521-021-05813-1>
6. Kim, J.H., Sin, B.-K.: Online handwriting recognition. In: Doermann, D., Tombre, K. (eds.) *Handbook of Document Image Processing and Recognition*, pp. 887–915. Springer, London (2014). https://doi.org/10.1007/978-0-85729-859-1_29
7. Barakat, B., Droby, A., Kassis, M., El-Sana, J.: Text line segmentation for challenging handwritten document images using fully convolutional network. In: *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 374–379 (2018)
8. Keyser, D., Deselaers, T., Rowley, H.A., et al.: Multi-language online handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1180–1194 (2017)
9. Graves, A.: Generating sequences with recurrent neural networks. *arXiv* (2013). <https://doi.org/10.48550/arXiv.1308.0850>
10. Shrivastava, A., Jaggi, I., Gupta, S., et al.: Handwritten digit recognition using machine learning: a review. In: *2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, pp. 322–326 (2019)
11. Corr, P.J., Silvestre, G.C., Bleakley, C.J.: Open source dataset and deep learning models for online digit gesture recognition on touchscreens. In: *2017 Irish Machine Vision and Image Processing Conference (IMVIP)* (2017). <https://doi.org/10.48550/arXiv.1709.06871>
12. Li, Z., Jin, L., Lai, S., et al.: Improving attention-based handwritten mathematical expression recognition with scale augmentation and drop attention. In: *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 175–180 (2020)
13. Wang, J., Du, J., Zhang, J., et al.: Multi-modal attention network for handwritten mathematical expression recognition. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1181–1186 (2019)
14. Zhang, J., Du, J., Yang, Y., et al.: SRD: a tree structure based decoder for online handwritten mathematical expression recognition. *IEEE Trans. Multimed.* **23**, 2471–2480 (2021)
15. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30, pp. 6000–6010 (2017)
16. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186. Association for Computational Linguistics (2019)
17. Brown, T., Mann, B., Ryder, N., et al.: Language models are few-shot learners. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901 (2020)

18. Wolf, T., Debut, L., Sanh, V., et al.: Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45 (2020). <https://doi.org/10.18653/v1/2020.emnlp-demos.6>
19. Parmar, N., Vaswani, A., et al.: Image transformer. In: International Conference on Machine Learning, pp. 4055–4064. PMLR (2018)
20. Huang, C.Z.A., Vaswani, A., et al.: Music transformer: generating music with long-term structure. In: International Conference on Learning Representations (ICLR) (2019)
21. Zhao, H., Jiang, L., Jia, J., et al.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 16259–16268 (2021)
22. Kozlov, A., Andronov, V., Gritsenko, Y.: Lightweight network architecture for real-time action recognition. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing, pp. 2074–2080 (2020)
23. D'Eusanio, A., Simoni, A., Pini, S., et al.: A transformer-based network for dynamic hand gesture recognition. In: 2020 International Conference on 3D Vision (3DV), pp. 623–632 (2020)
24. Wang, S., Li, B.Z., Khabsa, M., et al.: Linformer: self-attention with linear complexity. arXiv (2020). <https://doi.org/10.48550/arXiv.2006.04768>
25. Kitaev, N., Kaiser, L., Levskaya, A.: Reformer: the efficient transformer. In: International Conference on Learning Representations (2020)
26. Choromanski, K., Likhoshesterov, V., Dohan, D., et al.: Rethinking attention with performers. In: International Conference on Learning Representations (2021)
27. Rao, R.M., Liu, J., Verkuil, R., et al.: MSA transformer. In: International Conference on Machine Learning, pp. 8844–8856. PMLR (2021)
28. Akinremi, O., Balado, F., Silvestre, G.C.: A machine translation model for online glyph recognition. UCD Internal Research Report (2021, to be published)
29. Yujian, L., Bo, L.: A normalized levenshtein distance metric. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1091–1095 (2007)





Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Analysis of Attention Mechanisms in Box-Embedding Systems

Jeffrey Sardina¹, Callie Sardina², John D. Kelleher³,
and Declan O’Sullivan¹

¹ Trinity College Dublin, Dublin, Ireland
{sardinaj,Declan.OSullivan}@tcd.ie

² Boston College, Newton, MA 02467, USA
sardinac@bc.edu

³ Technological University Dublin, Dublin, Ireland
john.d.kelleher@tudublin.ie

Abstract. Large-scale Knowledge Graphs (KGs) have recently gained considerable research attention for their ability to model the inter- and intra- relationships of data. However, the huge scale of KGs has necessitated the use of querying methods to facilitate human use. Question Answering (QA) systems have shown much promise in breaking down this human-machine barrier. A recent QA model that achieved state-of-the-art performance, Query2box, modelled queries on a KG using box embeddings with an attention mechanism backend to compute the intersections of boxes for query resolution. In this paper, we introduce a new model, Query2Geom, which replaces the Query2box attention mechanism with a novel, exact geometric calculation. Our findings show that Query2Geom generally matches the performance of Query2box while having many fewer parameters. Our analysis of the two models leads us to formally describe the interaction between knowledge graph data and box embeddings with the concepts of semantic-geometric alignment and mismatch. We create the Attention Deviation Metric as a measure of how well the geometry of box embeddings captures the semantics of a knowledge graph, and apply it to explain the difference in performance between Query2box and Query2Geom. We conclude that Query2box’s attention mechanism operates using “latent intersections” that attend to the semantic properties in embeddings not expressed in box geometry, acting as a limit on model interpretability. Finally, we generalise our results and propose that semantic-geometric mismatch is a more general property of attention mechanisms, and provide future directions on how to formally model the interaction between attention and latent semantics.

Keywords: Box embeddings · Knowledge graph · Question answering · Attention

1 Introduction

This section is structured as follows: in Sect. 1.1, an introduction to knowledge graphs and question-answering systems is given. Section 1.2 then introduces the concept of box embeddings and the state-of-the-art Query2box model.

All code is made available here: <https://github.com/Jeffrey-Sardina/Query2Geom>

1.1 Knowledge Graphs and Question-Answering Systems

A Knowledge Graph (KG) is a data structure that represents data objects as nodes, and the relationships between them as labelled directed edges. It is commonly denoted as $G(V, E)$, where G is the graph, V is the set of nodes (or vertices) and E is the set of edges.

The smallest unit of a knowledge graph is a triple (h, r, t) , which consists of a head node h , a tail node t , and a labelled edge r connecting the head to tail. For example, $(\textit{Madrid}, \textit{capitalof}, \textit{Spain})$ could represent the fact “Madrid is the capital of Spain.”

While KGs are very simple in principle, human use is greatly limited by their size: individual KGs can have billions of triples. While this is very noticeable at the scale of the entire “Semantic Web” of interlinked Knowledge Graphs [2], it is also notable among individual KGs [1]. For example, the FB15k dataset has 592,213 triples, its subset FB15k-237 has 272,115 triples [1].

Searching for information in such large KGs is prohibitive. Several approaches seek to address this problem; notably, KG-based Question Answering (QA) systems seek to use machine learning to automatically answer queries posed based on the knowledge in a KG [5]. Such machine learning systems use latent vector representations of the nodes and edges in a knowledge graph to predict the correct answer to a posed query. Several groups have recently investigated this direction, including [3–5].

In this paper, we focus in particular on one particular class of embedding-based QA systems: box embeddings, introduced to the QA task on KGs by Ren et al. [5]. This method of question answering will be treated in detail in the following section.

1.2 Box Embeddings and Query2box

Ren et al. was the first group to apply box embeddings to question answering on knowledge graphs [5]. They did this through their model Query2box, which beat state-of-the-art performance when it was published. In short, Query2box embeds nodes in the graph as points in vector space [5]. They then embed questions as boxes whose contents should contain the answers to the question (and only the answers) [5].

This system can not only answer simple questions, but can also handle questions involving logical relationships such as conjunctions by taking the intersection of multiple boxes [5]. For example, it could model the set of nations that

border both Spain and France as the intersection of the box representing nations bordering Spain with the box representing nations bordering France.

Query2box makes the choice of calculating the intersection not through geometry, but using an attention mechanism over box embedding vectors [5]. However, they do not compare their system to a geometric-based calculation of box intersections, even though they compare to other, non-geometric baselines [5]. Thus, it remains unclear whether the attention mechanism simply learns to approximate a geometrically precise solution, or whether it is able to produce even better results by attending specifically to features of latent space.

In this paper, we address this gap in our understanding of box embeddings. Our contributions are twofold: first, we propose a new model, Query2Geom, which uses a geometrically exact mechanism to determine the intersection of boxes with fewer trainable parameters, fewer high-level learning components, and a more simple human-interpretation of results. By comparing Query2Geom to Query2box, we find that Query2Geom performs nearly as well, and in some cases better, than Query2box. These results imply that the use of an attention mechanism has few benefits that go beyond simply approximating a geometric solution, while at the same time leading to a higher-parameterised and less-interpretable system.

Second, through an analysis of the difference between attention-based and geometric-based methods, we argue that attention outperforms geometric calculation in some cases because it can attend to latent properties of boxes rather than to their geometry alone. We suggest that the underlying node embeddings used by Query2box do not allow fully expressive box embeddings, and that attention compensates for this inexpressivity in a way geometry cannot. We formalise this description in terms of the concept of “semantic-geometric mismatch”. Our results show that semantic-geometric mismatch can simultaneously explain the slight performance loss in Query2Geom compared to Query2box, while also leading directly to the creation of a novel metric of embedding expressivity: the Attention Deviance Metric.

We define the Attention Deviance Metric as the difference between attention-based performance and geometry-based performance, which thus quantifies semantic-geometric mismatch. In this case, embeddings are not fully expressive and are better modelled through attention, which results in greater deviance between the performance of attention-based and geometric-based solutions. When box embeddings are fully expressive this suggests that geometry and semantics of latent space align, removing the deviance between attention and geometric solutions. We call this case semantic-geometric alignment. We then propose future directions for how to more explicitly model for, and take advantage of, this knowledge to further improve box-based QA systems. Finally, we discuss the implications of these findings for the interpretation and formal analysis of attention systems outside of box embeddings alone.

The rest of this paper is organised as follows. Section 2 outlines our methods, notably how we perform geometric box intersection, and how we compare Query2Geom to Query2box. Section 3 presents our results. Section 4 provides

a discussion of our findings and our future work. Finally, Sect. 5 concludes the paper.

2 Methods

This section is structured as follows: Sect. 2.1 gives an introduction to the box intersection problem. It then describes in detail our method for geometric calculation of the intersection box centre point and offset. Section 2.2 explains our methods for comparing Query2Geom with Query2box.

2.1 The Box Intersection Problem

The box intersection problem can be modelled as follows. A box (or hyper-rectangle) exists in R^n . It is defined by two vectors: a centre point and an offset, which is a vector made of strictly non-negative values that describes the translation from the centre of the box to one of its vertices. An example of this in R^2 is shown in Fig. 1.

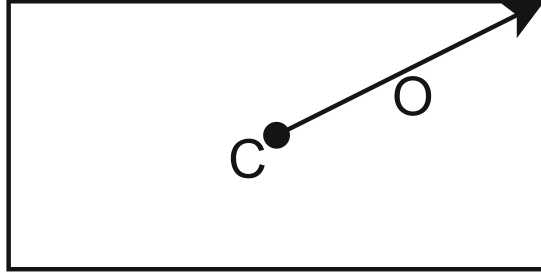


Fig. 1. A depiction of a box in R^2 . C is the box centre, and O is the offset vector.

The Box Intersection Problem is then defined as follows. Start with two boxes, A and B , which have centre points C_A and C_B and offsets O_A and O_B respectively. Box intersection attempts to find the so-called intersection box C that is formed by the overlap of boxes A and B . An illustration of this in R^2 is shown in Fig. 2.

In this paper, we will adopt the following mathematical notations for boxes. For a box A , C_A is the vector describing its centre and O_A the vector describing its offset. $C_{A,i}$ is used to describe the i^{th} element (i.e., the i^{th} dimension) of C_A ; $O_{A,i}$ is defined identically for the offset vector. For examples in R^2 , $C_{A,x}$ and $C_{A,y}$ will be used for the values in the x and y dimensions of the centre point, and $O_{A,x}$ and $O_{A,y}$ for the x and y dimensions of the offset vector.

It should be noted that since the intersection of any two boxes must be another box, the intersection box is therefore also fully described by a centre point and an offset vector.

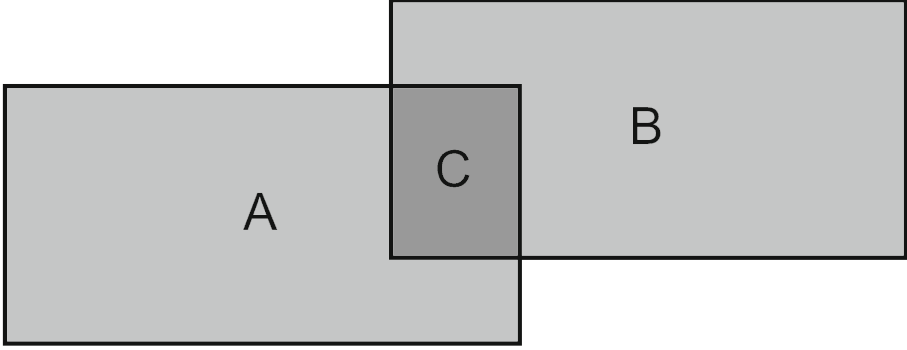


Fig. 2. The box intersection problem in R^2 . Boxes A and B are the boxes being intersected, and C is intersection box formed by their overlap.

Centre-Point Calculation. To calculate the intersection of two boxes A and B , we start by finding the centre point of the intersection box C . To do this, we first note that in each dimension $1..i..n$ of the centre point C_C in R^n , that the value of $C_{C,i}$ can be calculated independently of the other values in C_C . Thus, we decompose the n -dimensional box intersection problem into n 1-dimensional box intersection problems. We call this operation 1-D projection, and it is shown (in the R^2 case) in Fig. 3.

We now note that, in the case that $C_{A,i} < C_{B,i}$ the extent of the intersection box in dimension i (i.e., the distance from one hyper-edge to another along the i -axis) is bounded by two points: $C_{A,i} + O_{A,i}$ and $C_{B,i} - O_{B,i}$. Note that When $C_{A,i} > C_{B,i}$, these bounds simply reverse to $C_{A,i} - O_{A,i}$ and $C_{B,i} + O_{B,i}$.

Once we have the two end points in dimension i , finding the center point in dimension i is trivial: it is simply the arithmetic mean of the endpoints. Thus, the center point $C_{C,i}$ is given by

$$C_{C,i} = \begin{cases} ((C_{A,i} + O_{A,i}) + (C_{B,i} - O_{B,i}))/2, & \text{if } C_{A,i} < C_{B,i} \\ ((C_{A,i} - O_{A,i}) + (C_{B,i} + O_{B,i}))/2, & \text{otherwise} \end{cases} \quad (1)$$

Offset Calculation. Once the centre point of the intersection box, C_C is known, calculation of the offset is trivial. For any value $C_{C,i}$, we know the corresponding lower and upper bound that were used to calculate it as given in the previous section. The offset is simply the positive distance between this centre point and either of the endpoints. To be exact, the offset of the intersection box in dimension i , $O_{C,i}$, is given by

$$O_{C,i} = \begin{cases} C_{C,i} - (C_{B,i} - O_{B,i}), & \text{if } C_{A,i} < C_{B,i} \\ (C_{B,i} + O_{B,i}) - C_{C,i}, & \text{otherwise} \end{cases} \quad (2)$$

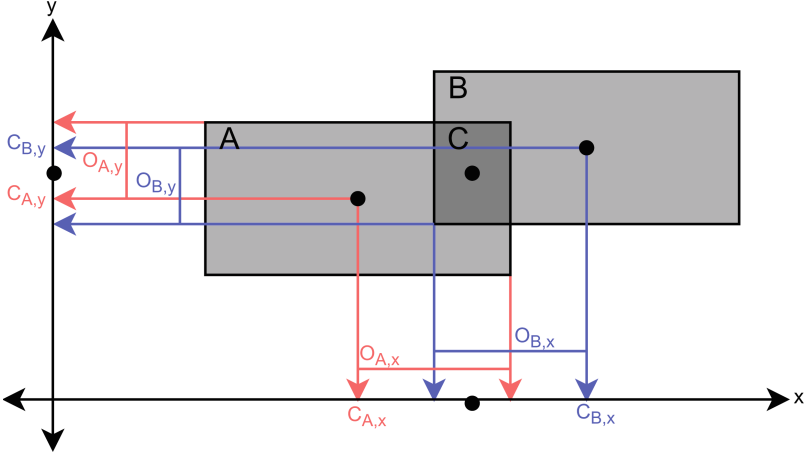


Fig. 3. 1-D projections to calculate the box centre in each dimension. $C_{A,x}$ is the x-coordinate of the centre point of box A , and $O_{A,x}$ is the x-coordinate of the offset vector of box A ; $C_{B,x}$ and $O_{B,x}$ are defined likewise for box B , and analogous variables are shown in the second dimension along the y-axis.

2.2 Comparison Against Query2box

In order to evaluate the performance of our model, we ran both Query2Geom (which uses our geometric box intersections) and Query2box (using the attention-based mechanism for calculating intersection centres and offsets, as in [5]). Both experiments were run with identical hyper-parameters and training configurations, using the setting determined in the original Query2box paper [5]. It is important to note that this includes the fact that both models were trained only on the $1p$, $2p$, $3p$, $2i$, and $3i$ query types (see the description below), and that remaining query types were only seen in testing. Our experiments were run on the same benchmarking datasets used in the original Query2box paper: FB15k, FB15k-237, and NELL995 [5].

Since Query2box evaluated its performance on a number of different types of queries input to it, we also report performance in all these various types of queries. A brief summary of these query types is given here; for a full explanation, see [5].

Query2box considers 9 types of queries, named $1p$, $2p$, $3p$, $2i$, $3i$, ip , pi , $2u$, and up . In this notation, p represents the “projection” operator, which is how Query2box models traversing relationship embeddings of the KG. i represents the intersection operator, which is how the logical conjunction is modelled. u represents the union operator, which is how logical disjunction is modelled. Numbers indicate the number of each operator applied (i.e. $2u$ means two union operations), and the order of elements describes the order in which the operators are used (i.e., pi means that a projection is followed by an intersection).

In the remainder of this paper, this notation will be used when discussing query types.

3 Results

This section is structured as follows: Sect. 3.1 compares the overall model complexity of Query2Geom and Query2box. Section 3.2 gives an analysis of the comparative performance of both models. Section 3.3 introduces our Attention Deviation metric and the concepts of semantic-geometric alignment and mismatch.

3.1 Model Complexity

The sizes of both models in terms of the number of parameters are summarised in Table 1. The relative decrease in the number of parameters used in Query2Geom compared to Query2box is also given.

Table 1. A summary of the number of model parameters when trained with an embedding dimension of 400.

Dataset	#params (Query2box)	#params (Query2Geom)	% reduction
FB15k	8772400	8132400	7.30%
FB15k-237	6821200	6181200	9.38%
NELL995	26304400	25664400	2.43%

It is critical to note that the simplifications in Query2Geom are not simply reductions in parameter usage. The parameter reduction comes from entirely replacing two learning components of the model – namely intersection box centre calculation and offset calculation – with fixed geometric formulas. It is a reduction in the number of higher-level learning components of the original model, which translates to simplification at the conceptual, interpretability, and architectural levels.

3.2 Model Performance

Performance was measured by four different scores: MRR (mean reciprocal rank), Hits@1, Hits@3, and Hits@10. All of these scores are calculated based on the ranking of correct responses among incorrect ones. MRR is the mean reciprocal rank of the correct answer among incorrect answers; the three Hits@k measures are the proportion of correct answers in the top k elements of the ranking.

The scores of both models by each of these metrics are summarised in Table 2, Table 3, and Table 4.

In general, Query2Geom matches or nearly matches the performance of Query2box on all datasets and query types examined. The two exceptions to this

Table 2. The performance of Query2Geom vs Query2box on each query type for FB15k. Performance is measured in terms of MRR (mean reciprocal rank), Hits@1, Hits@3, and Hits@10.

FB15k					
Query type	Model	MRR score	Hits@1 score	Hits@3 score	Hits@10 score
1p	Query2Geom	0.650267	0.489999	0.780970	0.903410
	Query2box	0.660733	0.504797	0.787674	0.905719
2p	Query2Geom	0.374285	0.275038	0.411344	0.572442
	Query2box	0.378315	0.278593	0.415397	0.579878
3p	Query2Geom	0.273310	0.182618	0.302931	0.447608
	Query2box	0.277408	0.186537	0.307702	0.452624
2i	Query2Geom	0.466711	0.324910	0.555033	0.720269
	Query2box	0.492412	0.343777	0.590193	0.756878
3i	Query2Geom	0.559498	0.417586	0.658477	0.809164
	Query2box	0.604431	0.459966	0.710518	0.852715
ip	Query2Geom	0.183677	0.111511	0.199075	0.322121
	Query2box	0.192262	0.117338	0.211144	0.337771
pi	Query2Geom	0.335959	0.217770	0.394403	0.558723
	Query2box	0.349359	0.226694	0.410136	0.583978
2u	Query2Geom	0.460892	0.269328	0.600964	0.805769
	Query2box	0.476807	0.289031	0.613005	0.811203
up	Query2Geom	0.298903	0.202786	0.326896	0.498826
	Query2box	0.303698	0.209297	0.328150	0.502811

Table 3. The performance of Query2Geom vs Query2box on each query type for FB15k-237. Performance is measured in terms of MRR (mean reciprocal rank), Hits@1, Hits@3, and Hits@10.

FB15k-237					
Query type	Model	MRR score	Hits@1 score	Hits@3 score	Hits@10 score
1p	Query2Geom	0.402108	0.278309	0.468142	0.634959
	Query2box	0.403250	0.279491	0.468547	0.638436
2p	Query2Geom	0.225566	0.144491	0.246415	0.386744
	Query2box	0.228106	0.147005	0.249112	0.392738
3p	Query2Geom	0.175492	0.108264	0.189519	0.312062
	Query2box	0.176689	0.108015	0.190522	0.314173
2i	Query2Geom	0.237886	0.124677	0.283483	0.465320
	Query2box	0.274497	0.157053	0.324284	0.513665
3i	Query2Geom	0.338280	0.216228	0.401429	0.571145
	Query2box	0.378354	0.248346	0.452466	0.621043
ip	Query2Geom	0.099940	0.052975	0.100593	0.188812
	Query2box	0.106850	0.056463	0.111404	0.201593
pi	Query2Geom	0.171607	0.095063	0.193016	0.315974
	Query2box	0.182645	0.100906	0.205410	0.340416
2u	Query2Geom	0.201138	0.087873	0.242313	0.432366
	Query2box	0.206530	0.094251	0.247073	0.434787
up	Query2Geom	0.176320	0.103061	0.188334	0.329051
	Query2box	0.180102	0.105682	0.191140	0.328722

Table 4. The performance of Query2Geom vs Query2box on each query type for NELL995. Performance is measured in terms of MRR (mean reciprocal rank), Hits@1, Hits@3, and Hits@10.

NELL995					
Query type	Model	MRR score	Hits@1 score	Hits@3 score	Hits@10 score
1p	Query2Geom	0.411444	0.227585	0.550640	0.710378
	Query2box	0.414124	0.229345	0.556521	0.711783
2p	Query2Geom	0.225108	0.128063	0.263442	0.418847
	Query2box	0.228444	0.131157	0.266658	0.423136
3p	Query2Geom	0.204901	0.123397	0.230799	0.360781
	Query2box	0.208871	0.128412	0.234453	0.362386
2i	Query2Geom	0.271019	0.152069	0.319716	0.524850
	Query2box	0.289603	0.162182	0.348049	0.553331
3i	Query2Geom	0.398097	0.275297	0.453691	0.643913
	Query2box	0.421173	0.290373	0.488008	0.678520
ip	Query2Geom	0.117654	0.066895	0.123042	0.215243
	Query2box	0.124954	0.072257	0.131605	0.226891
pi	Query2Geom	0.200642	0.120546	0.225809	0.354733
	Query2box	0.192567	0.116442	0.213335	0.339977
2u	Query2Geom	0.261958	0.099772	0.367293	0.576311
	Query2box	0.264654	0.100538	0.371798	0.579765
up	Query2Geom	0.154615	0.078754	0.163667	0.322732
	Query2box	0.156213	0.079159	0.167033	0.329692

are the $2i$ and $3i$ query types, in which case Query2box outperforms Query2Geom by a wider margin relative to other query types on each dataset. On the other query types using intersections, ip and pi , Query2Geom has a much smaller performance gap. That gap is almost always present, with the one major exception being Query2Geom out-performing Query2box on pi queries on NELL995. Overall, this suggests that the attention mechanism employed in Query2box is able to find slightly more performant intersections than the exact geometric values by attending to the properties of latent space.

On the other hand, Query2Geom uses between 2.5% and 10% fewer parameters than Query2box on the datasets tested, and in most cases performs almost identically to Query2box; notably, see $1p$, $2p$, $3p$, $2u$, and up queries on all datasets.

3.3 The Attention Deviation Metric and Semantic-Geometric Alignment and Mismatch

Our final result is the derivation of two critical ideas: the Attention Deviation Metric (ADM) and the concept of Semantic-Geometric Alignment and Mismatch.

In cases where there was a discrepancy between the performance of Query2Geom (which attempts to enforce semantic-geometric alignment) and Query2box (which allows semantic-geometric mismatch), we note that Query2box generally performed slightly better. This is clear evidence for the presence of semantic-geometric mismatch in Query2box: if its attention mechanism only served to approximate a geometric solution, then it would have been perfectly matched or outperformed by Query2Geom’s exact geometric solution, as that exact solution would have less error than the solution found by the attention mechanism. The deviation between these two scores can then be explained by the attention mechanism performing a “latent intersection” rather than a geometric one, using the hidden properties of boxes to yield more accurate results.

This leads us to the concepts of semantic-geometric alignment and mismatch. When box intersections are geometrically exact, then a box represents all answers to a query, and an intersection of two boxes represents all answers that satisfy both queries they represent. At a mathematical level, this means that the model attempts to enforce perfect alignment between the latent properties of the box embeddings and the geometric properties of embedding space. In the case that the embedding system is able to fully capture the semantics of the knowledge graph, it should produce such an alignment. Since the latent properties of the embeddings are representations of the semantics of the Knowledge Graph, we call this case “semantic-geometric alignment”. In other words, when there is semantic-geometric alignment, the semantics of the knowledge graph relevant to the question-answering task are contained within the geometry of their embeddings.

When attention is used rather than geometry to determine box intersection, semantic-geometric alignment is not enforced. Instead, the attention mechanism is encouraged to examine the latent features of box embeddings, and to give higher or lower weights to various elements of latent space that it finds correlate better or worse to the model’s training performance – even when that leads to geometrically inexact intersections. We call these intersections latent intersections to distinguish them from exact geometric intersections. The use of latent intersections leads to semantic-geometric mismatch. We note that the primary drive for using latent intersections would be in the case that the embeddings are not fully expressive, since semantic-geometric alignment would not be able to hold for such a system.

Looking back at the gap between Query2box and Query2Geom, we formally describe this deviation as one caused by semantic-geometric mismatch that drives Query2box’s attention mechanism to use slightly higher-performing latent intersections rather than approximating geometric ones. This leads us directly to the Attention Deviation Metric. ADM seeks to measure to what extent the latent intersections are able to learn more than geometric intersections: in other words, to quantify the extent of semantic-geometric mismatch. The Attention Deviation Metric is given by the following formula:

$$ADM = score(Attn) - score(Exact)$$

where *score* is a score function such as MRR, *Attn* is a model using attention-based calculation (such as Query2box) and *Exct* is a model replacing attention with an exact mathematical calculation (such as Query2Geom).

Here, using MRR as our score function, Query2box as *Attn* and Query2Geom as *Exct*, we can calculate the ADM between these two values. For example, looking at *2i* queries on FB15k-237, we can calculate

$$\begin{aligned} ADM &= MRR(Query2box) - MRR(Query2Geom) \\ ... &= 0.274497 - 0.237886 = 0.036611 \end{aligned}$$

Similarly, for *1p* queries on FB15k-237, the associated ADM value is 0.001142. This means that for FB15k-237, the use of latent intersections rather than geometric ones has a far (30x) higher impact on *3i* queries than on *1p* ones. This pattern generalises our previous observation that Query2box tends to perform better on queries with intersections compared to Query2Geom.

4 Discussion and Future Directions

We succeed in generally matching the performance of Query2box with a much simpler model, Query2Geom, that uses exact geometric calculations instead of an attention mechanism to calculate box intersections. Query2Geom has several benefits – many fewer trainable parameters and fewer high-level learning components. This results in a lighter demand on resources and a better ability to scale. However, it also means that the model is easier to interpret, since it is based on semantic-geometric alignment unlike Query2box, which is based on semantic-geometric mismatch.

The necessity of using a latent intersection fundamentally implies that the boxes constructed by Query2box are not fully expressive; i.e. that they do not fully capture the concept grouping they were designed to model. After all, if the boxes were fully expressive, then the power of the geometric intersection operator would approach that of the attention intersection operator, because perfect semantic-geometric alignment would hold in latent space. This effect is quantified by our Attention Deviation Metric, which captures the limits on expressivity of box embeddings through using semantic-geometric mismatch.

We propose that Query2Geom cannot reach full semantic-geometric alignment because the underlying node embeddings are not able to capture the full semantics of the Knowledge Graph. More expressive embeddings would result in semantic-geometric alignment, which would eliminate the small remaining benefit of attention. Creating such an aligned system and determining its properties is left as a future direction.

Beyond the realm of box embeddings, our work has a critical point to make about the function of attention in general. The clear presence semantic-geometric mismatch when attention is used implies that attention does not serve to simply approximate exact geometric (or other mathematical) functions. Instead, attention exists to learn how to use latent entity representations that are not captured

by geometry and exact formulas. We hypothesize that the ADM presented here will generalise to other attention mechanisms, and that the extent of semantic-geometric alignment or mismatch in a model can be calculated by ADM with attention-free alternatives. Exploring this hypothesis is left as a future direction.

5 Conclusion

In this paper, we introduced Query2Geom, a modification of Query2box that replaces its attention-based box intersection system with an exact geometric one. Our results indicate that both models perform very similarly, but that Query2box slightly outperforms Query2Geom because its attention mechanism allows it to attend to aspects of latent space that are not captured in a pure geometric model – a case we formalise as semantic-geometric mismatch. This led us to propose the Attention Deviation Metric, which models the expressiveness of a box embedding model by the performance lost when replacing attention-based intersection with precise geometric calculations of box intersections.

We leave as a future direction applying the Attention Deviation Metric to estimate the performance of other box embedding models, and other attention-based models more generally. Finally, we propose that research in this direction will not only lead to improvements in model performance, but also lead to increases in training resource- and time- efficiency.

Acknowledgements. This research was conducted with the financial support of Science Foundation Ireland D-REAL CRT under Grant Agreement No. 18/CRT6225 at the ADAPT SFI Research Centre at Trinity College Dublin, together with sponsorship of Sonas Innovation Ireland. The ADAPT SFI Centre for Digital Content Technology is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant # 13/RC/2106_P2.

References

1. Ali, M., et al.: Bringing light into the dark: a large-scale evaluation of knowledge graph embedding models under a unified framework. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(12), 8825–8845 (2021). <https://doi.org/10.1109/TPAMI.2021.3124805>
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data: the story so far. *Int. J. Semant. Web Inf. Syst.* **5**, 1–22 (2009). <https://doi.org/10.4018/jswis.2009081901>
3. Gu, K., Miller, J., Liang, P.: Traversing knowledge graphs in vector space (2015). <https://doi.org/10.18653/v1/D15-1038>
4. Hamilton, W.L., Bajaj, P., Zitnik, M., Jurafsky, D., Leskovec, J.: Embedding logical queries on knowledge graphs. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS 2018*, Curran Associates Inc., Red Hook, NY, USA, pp. 2030–2041 (2018)
5. Ren, H., Hu, W., Leskovec, J.: Query2box: reasoning over knowledge graphs in vector space using box embeddings. In: *International Conference on Learning Representations* (2020). <https://openreview.net/forum?id=BJgr4kSFDS>




Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





A Machine Learning Approach to Industry Classification in Financial Markets

Rian Dolphin¹(✉) , Barry Smyth^{1,2} , and Ruihai Dong^{1,2} 

¹ School of Computer Science, University College Dublin, Dublin, Ireland
rian.dolphin@ucdconnect.ie

² Insight Centre for Data Analytics, University College Dublin, Dublin, Ireland
{barry.smyth,ruihai.dong}@ucd.ie

Abstract. Industry classification schemes provide a taxonomy for segmenting companies based on their business activities. They are relied upon in industry and academia as an integral component of many types of financial and economic analysis. However, even modern classification schemes have failed to embrace the era of big data and remain a largely subjective undertaking prone to inconsistency and misclassification. To address this, we propose a multimodal neural model for training company embeddings, which harnesses the dynamics of both historical pricing data and financial news to learn objective company representations that capture nuanced relationships. We explain our approach in detail and highlight the utility of the embeddings through several case studies and application to the downstream task of industry classification.

Keywords: Machine learning · Latent space embeddings · Knowledge graphs · Financial markets

1 Introduction

Financial markets are an important but challenging machine learning domain when it comes to analysis and prediction [3, 9, 17]. Their stochastic nature reflects a complex network of interactions involving a web of hidden factors and unpredictable events. Though the financial literature spans many sub-domains, the application of machine learning and deep learning techniques to financial markets has often been narrowly focused on the problem of returns forecasting for *individual* assets [25]. As a result, many other problems facing the financial sector have been underrepresented or ignored.

For example, the challenge of classifying companies based on a taxonomy of industry types is not well covered by contemporary machine learning research, even though it is an important task in several settings. In government, private sector, academia, and even the broader public, industry classification schemes are an integral part of using business and economic information [20]. Additionally, research has shown that 30% of publications at the top-three finance journals

utilize industry classification schemes [27]. The ability to segment companies into market sectors is important for many types of financial and economic analysis—measuring economic activity, identifying peers and competitors, constructing ETF products, quantifying market share and bench-marking company performance—none of which would be possible without industry classifications [20].

The rise in popularity of sector-based investing has led to the development of new market-oriented industry classification schemes. However, despite their increased usage for choosing investments, many industry classification schemes have still not embraced the era of big data and remain a largely subjective undertaking. As a result, they have been shown to struggle with scalability [20], exhibit inconsistencies when determining the primary area of activity for a company [19], and they offer no way to quantitatively measure or rank similarity between companies. Other studies confirm significant disagreement between classification schemes when trying to categorize the same companies [12, 14].

Although research applying modern computational techniques to industry classification schemes, and the learning of asset relationships more broadly, has lagged, there has been marked progress within the computer science community on relevant areas like representation learning and machine learning on relational data. The model architecture presented in this paper takes inspiration from a class of modern language models that have proven to be very successful in the natural language processing (NLP) domain [10, 18].

In this work, we propose a novel training methodology for the learning of distributed representations of public companies, based on distributional similarities in both historical returns data and financial news content. We show how these *multimodal embeddings* can successfully capture the nuanced relationships that exist between companies and we demonstrate how they can be used to identify related companies. After discussing related work in the next section, we go on to explain the proposed approach before presenting several case-studies to highlight how the learned representations can be useful in financial applications. Before concluding, we present the results of an initial evaluation to demonstrate the effectiveness of these learned representations for the downstream task of industry classification.

2 Related Work

Finance has long been a pioneering industry in the application of machine learning techniques [25]. However, the literature applying modern computational techniques to financial markets overwhelmingly focuses on the forecasting of returns and volatility for *individual* stocks [16]. Although these applications have seen success, there are many other tasks within financial markets which have not received the same level of attention. In this paper, we look to address one of these – namely, the problem of industry classification.

Moving away from treating companies independently, and instead leveraging relationships, is key to tackling this task. Recent advances in areas such as representation learning and graph ML have encouraged research in this direction, with

the most relevant literature to this work being papers proposing novel embedding frameworks for financial assets. For example, [28] suggest using matrix factorization to learn latent representations of stocks based on a co-occurrence matrix obtained from financial news articles. [8] propose a framework for learning stock embeddings from the co-dynamics of historical returns data. The authors in [24] use network theory and machine learning to generate fund and ETF embeddings based on overlapping asset allocation, [23] apply Node2Vec to the stock correlation matrix to learn embedded representations of stocks, and [2] obtain embeddings by combining company information with knowledge from Wikipedia and relationships from Wikidata.

Other relevant applications of NLP in the financial domain include [26] where sentiment dynamics in related companies are assessed by building a network from financial news data. Authors in [15] tackle industry classification by using NLP to extract distinguishable features from business descriptions in financial reports. Also, authors in [13] extract company embeddings by using the output of the BERT [6] language model applied to annual reports, and then use these embeddings in the industry classification task. Though textual data has been used to inform company embeddings in prior research, there is often a reliance on the aggregation of pre-trained word embeddings rather than a tailored company embedding framework, as proposed in this work.

3 A Multimodal Embedding-Based Approach

This paper proposes a method for training embeddings of companies using a probabilistic neural framework. In this section, we first outline the link with the NLP domain. We then describe a methodology for applying the proposed framework to historical returns data in Sect. 3.2, and financial news data in Sects. 3.3. Following this, we describe the training process in Sect. 3.4.

3.1 Language Modelling Origins

Inspired by the use of distributional semantics in natural language processing, we propose a model architecture that uses the idea of *context companies* to train distributed representations of target companies. In linguistics, the distributional hypothesis captures the idea that “a word is characterized by the company it keeps” [10], i.e., words that occur in the same contexts tend to have similar meanings. In language modelling, the *context* of any given word has quite a natural interpretation as the words immediately before and after it. However, defining *context* in the case of financial assets is not as intuitive. But, before laying out the proposed approach to for financial assets, we first give some background on the Word2Vec [18] architecture to frame the discussion.

The distributional hypothesis underpins many modern language models like Word2Vec [18]. The goal of such language models is to construct a lower dimensional, dense representation of words that capture meaningful semantic and syntactic relationships [24]. The typical model architecture is a shallow two-layer



Fig. 1. Embedding pipeline for historical prices

neural network where the embeddings themselves are also the model parameters/weights. The embeddings are randomly initialized, as would be expected for weights in a neural model, and then trained by using *context words* (the model input) to predict the *center/target word* (the model output) and back propagating the loss¹. The result of these iterative updates is that words which commonly appear in the same context will have similar representations in the latent space. Further information about this architecture and training process can be found in [18, 22].

To adapt this modelling framework to non-textual data, we will now consider the idea of *context* in the case of the two data sources we consider in this work: financial news and historical stock returns.

3.2 Selecting Context Companies from Historical Returns

To select context companies from historical returns data, we treat companies with similar returns at the same points in time as related. This idea is supported by research showing that companies from the same business sectors tend to exhibit similar stock price fluctuations [11].

Consider a universe of public companies $U = \{a_1, \dots, a_{|U|}\}$ and for each company a_i we have a vector $\mathbf{p}_{a_i} = \{p_0^{a_i}, \dots, p_T^{a_i}\}$ containing its prices at discrete time intervals (daily or weekly for example). From the pricing data, we then compute a returns vector $\mathbf{r}_{a_i} = \{r_1^{a_i}, \dots, r_T^{a_i}\}$ using Eq. 1.

$$r_t^{a_i} = \frac{p_t^{a_i} - p_{t-1}^{a_i}}{p_{t-1}^{a_i}} \quad (1)$$

We generate *target:context sets* from these returns vectors. For a context size C , the context companies for target asset a_i at time t are simply the C companies which have the closest return at that point in time. The closest return is defined by the lowest absolute value difference in return for candidate company a_j , formulated as $|r_t^{a_i} - r_t^{a_j}|$. An example of this process is outlined in Fig. 1 with Apple Inc. (AAPL) as the context company and t as January 3rd 2000. We compute the absolute value difference between the return of AAPL on that day with the return of each other company on the same day. Then, we choose the C

¹ There are actually two Word2Vec architectures, we focus on CBOW and do not describe Skip-Gram here.

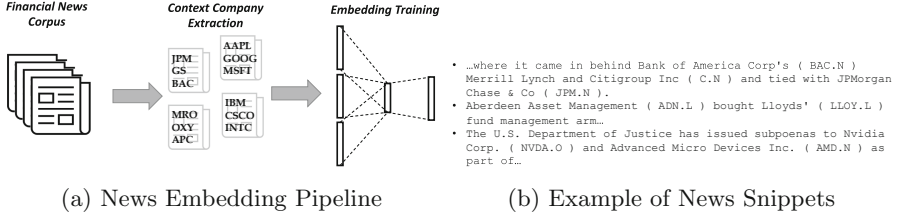


Fig. 2. Example of news snippets

companies with the lowest difference values as the context companies, excluding AAPL itself. More generally, we generate a target:context set for every company at each point in time, which results in a total of $|U| \times T$ sets for training.

An example of a target:context set for $C = 3$ might be $[MSFT : IBM, AAPL, ORCL]$. This tells us that, at some point in time, the three companies with the closest returns to Microsoft were IBM, Apple Inc. and Oracle.

Market data is notoriously noisy [5], and when looking at returns on the daily level, we see a bell shaped curve around that day's market average. This means that if the target stock has a return close to the market average on a given day, a lot of the corresponding context stocks are likely due to random chance. In an effort to isolate meaningful cases and reduce noise, a context set $\mathcal{S}(a_i, t)$, with target company a_i at time t , was deleted from the training data if the target stock return, $r_t^{a_i}$, was within the interquartile range (IQR) of returns on that day. As a result, only sets where the target stock had a movement outside the IQR of market returns on a given day were included in training.

The remaining target: context sets are then passed into the embedding training architecture, which will be described in more detail in Sect. 3.4.

3.3 Selecting Context Companies from News Articles

Unsurprisingly, movements in financial markets and financial news have been found to be intrinsically linked. For example, a positive correlation exists between the number of occurrences of a company in the Financial Times and the transaction volume of that company's stock both on the day before and the same day as the news is released [1]. Additionally, with the evolution of NLP techniques, the application of language modelling to financial news data for stock market forecasting on a stock-by-stock basis has become a popular area of research in recent years [29].

Focusing on individual companies/assets in isolation can mean that important relational information is missed. We hypothesize that companies co-mentioned in the same news articles are likely to be related [28], and that this can be leveraged to improve performance in tasks such as industry classification. As such, we want to learn embeddings in a way whereby companies which are commonly mentioned in the same news articles will end up having similar latent embeddings in terms of some suitable similarity metric, like cosine similarity for example. To do this, we use a corpus of over 100,000 financial news articles span-

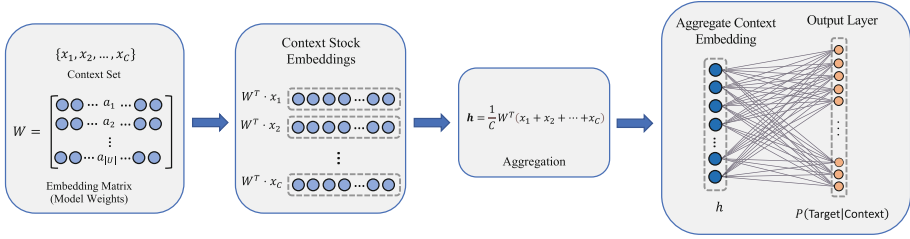


Fig. 3. Model architecture

ning 2006–2013 [7], and create target: context sets from every news article where more than one company is mentioned. Extracting the companies mentioned is helped by the fact that each company is followed by its associated stock ticker as shown in Fig. 2(b).

As an example, if a given news article mentions n companies, then each company will be put in a set as the target company with the remaining $n - 1$ companies listed as context companies in that set. Therefore, an article with n companies will result in n target:context sets for training. These sets are then all passed into the model framework, where the embeddings are trained. This process is shown in Fig. 2(a) and more detail will be given on the embedding training in Sect. 3.4.

3.4 The Training Process

The aforementioned shallow two-layer neural network model architecture is illustrated in Fig. 3. As previously mentioned, the model design is such that *the company embeddings are the model parameters*. As such, each row in the weight matrix W is a company embedding. In this section, we explain each step of this framework in detail and why the resulting embeddings capture the relationships of interest. Throughout this section, it is worth keeping in mind that two of these models will be used – one to learn the company embeddings for each data modality separately. The two independent embeddings are then concatenated to form the multimodal company embeddings.

The first step is to compute the hidden layer, which is simply an element-wise average of the context stock embeddings. To be more precise, the input to the model is a one-hot encoded version of the context set, and so, consists of C one-hot vectors $\{x_1, x_2, \dots, x_C\}$, one for each context stock. These vectors are used to extract the embeddings corresponding to the C context stocks. For example, computing $W^T \cdot x_1$ will extract a single row from W —the embedding corresponding to the first context stock. The hidden layer, h , is a simple element-wise average of the extracted embeddings, and is formulated in Eq. 2. We note that the use of a relatively simple average here is by design, since the mean function is agnostic to the number of inputs and so allows flexible context sizes during training.

$$\mathbf{h} = \frac{1}{C} \mathbf{W}^T (\mathbf{x}_1 + \mathbf{x}_2 + \dots + \mathbf{x}_C) \quad (2)$$

Thus, the hidden layer, \mathbf{h} , is an N -dimensional vector and can be thought of as an aggregate embedding representation of the context stocks, where N is the embedding dimensionality. The next step is to estimate the probability of the target company *given* \mathbf{h} by applying Eq. 3.

$$\mathbb{P}(\text{Target} \mid \text{Context}) = \text{softmax}(\mathbf{W}\mathbf{h}) \quad (3)$$

Ensured by using the softmax activation, the output is a posterior probability distribution expressing the probability of each stock in the universe being the target stock given the context stocks observed. Since the dot product represents a measure of similarity between vectors, the model assigns higher probability to stocks whose embeddings are similar to hidden layer embedding \mathbf{h} . In this way, when we apply back-propagation, stocks which commonly co-occur in target: context sets will end up closer in the embedding space. As a result, assuming our hypotheses are correct, the embeddings will capture nuanced relationships that are present in the historical returns data and financial news co-occurrence. We note that the ground truth here, y in Fig. 3, is a one-hot vector indicating the true target stock.

4 Evaluation

In this section, we will outline a number of interesting example case studies followed by an evaluation on the task of industry classification.

4.1 Datasets

For this analysis, we use publically available daily historical pricing data and over 100,000 financial news articles [7], both spanning 2006–2013. Included alongside the pricing data are two levels of industry labels for each stock, one high level like Finance, Technology etc., and the other a finer grained label like Major Bank or Semiconductors. The companies included in the analysis were selected based on the following inclusion criterion. Firstly, the company had to be publicly traded and have complete pricing data over the period in question. Secondly, we limited the dataset to companies mentioned in at least 50 news articles to ensure there was sufficient data for training. Additionally, the pricing data available contained only companies listed on the NYSE and NASDAQ exchanges, and so the dataset is also limited to these. After enforcing this inclusion criterion, we are left with 118 companies across seven industry sectors: Capital Goods, Consumer Non-Durables, Consumer Services, Energy, Finance, Health Care and Technology.

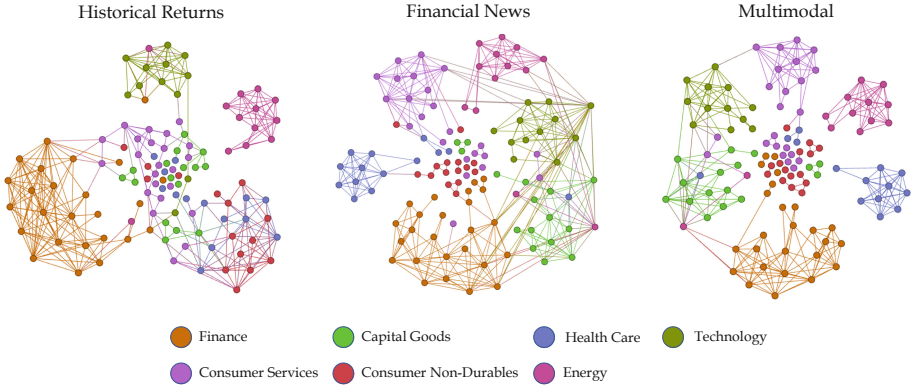


Fig. 4. Visualization of company embeddings colored by industry sector. Edges indicate high embedding similarity.

4.2 Company Knowledge Graph

Visualizing latent embeddings in two-dimensional space can often be a useful way of identifying relationships and clustering behavior. Figure 4 shows three knowledge graphs, where each node represents a company and nodes are colored by their industry sector label, with seven industries in total. Each graph is derived from different embeddings: one from the historical return embeddings, one from the financial news-based embeddings and one constructed from a concatenation of both embeddings. It is worth noting that the dimensionality of both types of embeddings, a hyperparameter, was chosen to be 20. As a result, the concatenated multimodal embeddings are 40-dimensional.

Firstly, to convert embeddings into knowledge graphs, a similarity matrix was computed using the cosine similarity between company embeddings. Then, if two companies had a similarity above a certain threshold, they would receive an edge between their nodes. The similarity threshold was chosen as 0.6, which resulted in approximately 5% of all possible edges being active. The plots are generated using a force-directed graph drawing algorithm in Gephi.

In each of the graphs in Fig. 4, we observe clear clustering of companies into industry sectors. Within each graph, edges tend to be present between nodes of the same industry sector, though there are some exceptions which will be discussed further in Sect. 4.4. This indicates that the proposed training framework successfully learns embeddings which pick up on relationships between companies in the case of both data modalities. In each case, using returns data or news co-occurrence, this is a very positive result because it suggests that it is possible to reconstruct important sectoral information from the embeddings, and indeed is likely to do so in a way that is more nuanced and objective than might be possible using simple sectoral labels.

Contrasting the three knowledge graphs, the graph constructed from the combined embeddings seems to best cluster the companies into industry sectors.

Table 1. Examples of top-3 nearest neighbors for given query companies

Query company Industry 1 Industry 2	Neighbor Company - Industry 1 - Industry 2	Similarity
JPMorgan Chase Finance Major Bank	Citibank - Finance - Major Bank	0.96
	Bank of America Corp - Finance - Major Bank	0.95
	Wells Fargo & Company - Finance - Major Bank	0.92
Intel Corporation Technology Semiconductors	Apple Inc. - Technology - Computer Manufacturing	0.84
	Texas Instruments - Technology - Semiconductors	0.83
	Hewlett-Packard - Technology - Computer Manufacturing	0.83
Walmart Consumer Services Department Stores	Target - Consumer Services - Department Stores	0.90
	Costco - Energy - Department Stores	0.85
	Best Buy - Consumer Services - Consumer Electronics	0.78

This indicates a benefit in combining the embeddings from both data modalities, which will be explored in further evaluations.

4.3 Identifying Related Companies

This first case study looks to use the learned embeddings to identify related companies through a nearest neighbors analysis, a natural first point of reference to sanity check the latent space representations. We would hope that companies with very high similarity in the latent space should be related somehow. In order to find the k -nearest neighbors (kNN) for a given query company, we first must define what exactly we mean by nearest. We implement kNN using cosine similarity as the similarity metric; note that a similar pattern of related companies results if we use euclidean distance or dot product similarity instead.

Table 1 shows the top-3 ($k = 3$) nearest neighbors for JPMorgan Chase, Intel Corporation, and Walmart, three well-known companies in very different sectors. In each case, the nearest neighbors pass the “sanity test” in that they belong to similar industry sectors and in many cases also agree on the finer-grained classification labelled as “Industry 2” in Table 1. For example, the three nearest neighbors of JPMorgan Chase, a major bank, are also all major banks. Remember, that no sectoral or industry information has been used in determining these nearest neighbors, and only daily returns and co-occurrence in news articles have been used to generate the distributed representations used for similarity assessment.

There is considerable scope for the use of nearest neighbor companies by investors. Firstly, we can develop a company recommendation system which, when given a target company – a novel company for the investor or one already in their portfolio – can generate a ranked list of similar companies based on their historical returns data and appearance in financial news. A system like this addresses a major pitfall of classic industry classification schemes, where no rank ordering is possible. This could have a variety of use cases, for example, investors and fund managers could consult this ranked list when conducting comparable

Table 2. Examples of high similarity mismatches—companies with very high similarity that have different sector labels

Company A Industry 1 - Industry 2	Company B Industry 1 - Industry 2	Similarity
General Electric Energy - Consumer Electronics	Boeing Capital Goods - Aerospace	0.87
Johnson & Johnson Consumer Non-Durables - Cosmetics	Colgate-Palmolive Healthcare - Major Pharma	0.81
3M Company Healthcare - Medical Instruments	Honeywell International Capital Goods - Auto Parts	0.88

company analysis or looking for alternative investment opportunities; it could be of use to sales representatives looking to recommend complementary investment opportunities to clients; investors could devise a tax loss harvesting strategy [24]; and asset managers could be assisted in the construction of market sector ETFs.

4.4 Analyzing High Similarity Mismatches

Though the vast majority of edges in Fig. 4 occur between nodes from the same industry, it is not true in all cases. In other words, there are some instances where two companies have a high embedding similarity, but their industry sector labels don’t match. Does this highlight a flaw in the embeddings, where companies achieve very high embedding similarity when they should not? To answer this, we provide some examples of these *high similarity mismatches*. We consider pairs of companies that have a high cosine similarity between embeddings and are members of different industry sectors. A number of examples are shown in Table 2.

The first example is that of General Electric, classified in the Energy (Consumer Electronics) sector, and Boeing, classified in the Capital Goods (Aerospace) sector. These two companies receive a high multimodal embedding similarity of 0.8 despite being classified in different industry sectors. However, upon closer inspection, we note that one of General Electric’s main business areas is the manufacturing of aircraft engines, with Boeing being one of their largest customers. As a result, they are commonly mentioned in news articles with Boeing and other aerospace companies, which results in the high embedding similarity.

Another example is that of Johnson & Johnson and Colgate-Palmolive, classified as Consumer Non-Durables (Cosmetics) and Healthcare (Major Pharma) respectively. These two companies again have a high embedding similarity, but have been classified into different industry sectors. The relationship could be explained by the large presence both companies have in the consumer healthcare market, resulting in exposure to similar idiosyncratic risk factors and the resulting headwinds.

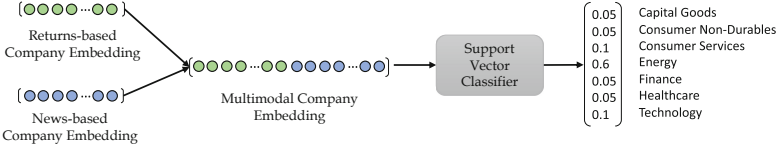


Fig. 5. Classifier framework

The final example in Table 2 is Honeywell and 3M. Again, they have relatively high similarity in their multimodal company embeddings, despite being classified in different industry sectors. The two companies are both multinational conglomerates operating in similar business sectors. In addition to this, both are constituents of three of the most popular indexes (Dow Jones Industrial Average, S&P 500 and S&P 100), and index inclusion has been shown to lead to more frequent news mentions and greater co-movement in returns [4].

Through these examples, we can see that current industry classification schemes will often segment quite similar companies into different industry sectors. The line is not always clear and, despite the increased usage of industry classification schemes for choosing investments, many have not embraced the era of big data and fail to utilize the countless data points being generated each day. Instead, the company allocation procedure remains a largely subjective task [20].

4.5 Using Multimodal Embeddings for Industry Classification

Using the embeddings generated by the proposed training framework, we can use a classification model to segment companies into business sectors in an objective manner. To do this, we train a support vector classifier [21] with embeddings as the input and industry sector label as the output. We used k -fold cross validation with $k = 4$ and account for a class imbalance in the data by using SMOTE.

There are a number of considerations here which will undoubtedly limit the accuracy of the classification model. Firstly, the embeddings themselves are derived solely from historical returns data and financial news, which are both influenced by a complex network of unpredictable factors. Secondly, as we saw in Sect. 4.4 when looking at high similarity mismatches, there are a number of companies with subjective ground truth labels that could be placed in a number of the industry categories. As a result, industry classification in the financial domain is a challenging problem.

Despite these hurdles, we see agreement of 90% with the traditional labels when using the pipeline in Fig. 5 to classify companies into industry sector. After an ablation study, we see an agreement of 85% when using the news-only embeddings, which is also above the baseline of 72% from the returns-only method Table 3.

Table 3. Results from industry classification task using k -fold cross validation with $k = 4$

Model	Precision	Recall	F1	Accuracy
Returns Embedding	0.74	0.72	0.72	72%
News Embedding	0.89	0.85	0.84	85%
Multimodal Embedding	0.91	0.90	0.90	90%

5 Conclusion

This work has focused on leveraging multiple sources of data to tackle the industry classification problem using machine learning. We proposed an approach for learning dense vector representations of companies that capture nuanced and interesting relationships between companies. The potential utility of these embeddings to financial analysts was discussed in relation to a number of tasks, and the evaluation results speak to the potential benefits of our approach and provide a useful starting point for further exploration and development. From examples in Sect. 4.3, we saw that the embeddings trained on each modality picked up on interesting relationships of different types. The benefits of the multimodal approach were also highlighted in the results of the industry classification model in Sect. 4.5, where combining modalities increased overall accuracy and F1 score.

In future work, we plan to adapt the proposed framework to generate multimodal embeddings optimized to capture dissimilarity, in addition to similarity, which is an important tool for effective portfolio optimization. We believe that these embeddings have the potential to be useful in the asset management space by informing successful diversification and risk management strategies. In addition, we plan to utilize other sources of data and introduce sentiment aware context company selection to assess the impact on performance.

Acknowledgements. This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183.

References

1. Alanyali, M., Moat, H.S., Preis, T.: Quantifying the relationship between financial news and the stock market. *Sci. Rep.* **3**(1), 1–6 (2013)
2. Ang, G., Lim, E.P.: Learning knowledge-enriched company embeddings for investment management. In: *Proceedings of the Second ACM ICAIF*, pp. 1–9 (2021)
3. Bachelier, L.: Théorie de la spéculation. In: *Annales scientifiques de l’École normale supérieure*, vol. 17, pp. 21–86 (1900)
4. Barberis, N., Shleifer, A., Wurgler, J.: Comovemen. *J. Financ. Econ.* **75**(2), 283–317 (2005)
5. De Long, J.B., Shleifer, A., Summers, L.H., Waldmann, R.J.: Noise trader risk in financial markets. *J. Polit. Econ.* **98**(4), 703–738 (1990)

6. Devlin, J., et al.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
7. Ding, X., et al.: Using structured events to predict stock price movement: an empirical investigation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1415–1425 (2014)
8. Dolphin, R., Smyth, B., Dong, R.: Stock embeddings: Learning distributed representations for financial assets. arXiv preprint [arXiv:2202.08968](https://arxiv.org/abs/2202.08968) (2022)
9. Fama, E.F.: The behavior of stock-market prices. *J. Bus.* **38**(1), 34–105 (1965)
10. Firth, J.R.: A synopsis of linguistic theory, 1930–1955. *Studies in linguistic analysis* (1957)
11. Gopikrishnan, P., Rosenow, B., Plerou, V., Stanley, H.E.: Identifying business sectors from stock price fluctuations. arXiv preprint cond-mat/0011145 (2000)
12. Guenther, D.A., Rosman, A.J.: Differences between COMPUSTAT and CRSP SIC codes and related effects on research. *J. Account. Econ.* **18**(1), 115–128 (1994)
13. Ito, T., et al.: Learning company embeddings from annual reports for fine-grained industry characterization. In: Proceedings of the Second Workshop on Financial Technology and Natural Language Processing, Kyoto, Japan (2020)
14. Kahle, K.M., Walkling, R.A.: The impact of industry classifications on financial research. *J. Financ. Quant. Anal.* **31**(3), 309–335 (1996)
15. Kim, D., Kang, H.G., Bae, K., Jeon, S.: An artificial intelligence-enabled industry classification and its interpretation. *Internet Res.* **32**(2), 406–424 (2021)
16. Li, W., et al.: Modeling the stock relation with graph network for overnight stock movement prediction. In: Proceedings of the Twenty-Ninth IJCAI (2021)
17. Malkiel, B.G., Fama, E.F.: Efficient capital markets: a review of theory and empirical work. *J. Finance* **25**(2), 383–417 (1970)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint [arXiv:1301.3781](https://arxiv.org/abs/1301.3781) (2013)
19. Parker, R.: Restoring the enterprise statistics program (esp) for the 2012 economic census. Reports of the Census Bureau (2012)
20. Phillips, R.L., Ormsby, R.: Industry classification schemes: an analysis and review. *J. Bus. Finance Librarianship* **21**(1), 1–25 (2016)
21. Platt, J., et al.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classifiers* **10**(3), 61–74 (1999)
22. Rong, X.: word2vec parameter learning explained. [arXiv:1411.2738](https://arxiv.org/abs/1411.2738) (2014)
23. Sarmah, B., Nair, N., Mehta, D., Pasquali, S.: Learning embedded representation of the stock correlation matrix using graph machine learning. arXiv preprint [arXiv:2207.07183](https://arxiv.org/abs/2207.07183) (2022)
24. Satone, V., Desai, D., Mehta, D.: Fund2vec: mutual funds similarity using graph learning. In: Proceedings of the Second ACM ICAIF, pp. 1–8 (2021)
25. Vachhani, H., et al.: Machine learning based stock market analysis: a short survey. In: Raj, J.S., Bashar, A., Ramson, S.R.J. (eds.) *ICIDCA 2019. LNDECT*, vol. 46, pp. 12–26. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-38040-3_2
26. Wan, X., et al.: Sentiment correlation in financial news networks and associated market movements. *Sci. Rep.* **11**(1), 1–12 (2021)
27. Weiner, C.: The impact of industry classification schemes on financial research. Available at SSRN 871173 (2005)
28. Wu, Qiong, e.a.: Equity2vec: end-to-end deep learning framework for cross-sectional asset pricing. In: Proceedings of the Second ACM ICAIF (2021)
29. Xing, F.Z., Cambria, E., Welsch, R.E.: Natural language based financial forecasting: a survey. *Artif. Intell. Rev.* **50**(1), 49–73 (2018)


Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





A Machine Learning Approach for Modeling and Analyzing of Driver Performance in Simulated Racing

Fazilat Hojaji^(✉) , Adam J. Toth, and Mark J. Campbell

Esports Science Research Lab, Lero Irish Software Research Centre, University of Limerick,
Limerick, Ireland

{Fazilat.Hojaji, Adam.Toth, Mark.Campbell}@ul.ie

Abstract. The emerging progress of esports lacks the approaches for ensuring high-quality analytics and training in professional and amateur esports teams. In this paper, we demonstrated the application of Artificial Intelligence (AI) and Machine Learning (ML) approach in the esports domain, particularly in simulated racing. To achieve this, we gathered a variety of feature-rich telemetry data from several web sources that was captured through MoTec telemetry software and the ACC simulated racing game. We performed a number of analyses using ML algorithms to classify the laps into the performance levels, evaluating driving behaviors along these performance levels, and finally defined a prediction model highlighting the channels/features that have significant impact on the driver performance. To identify the optimal feature set, three feature selection algorithms, i.e., the Support Vector Machine (SVM), Extreme Gradient Boosting (XGBoost) and Random Forest (RF) have been applied where out of 84 features, a subset of 10 features has been selected as the best feature subset. For the classification, XGBoost outperformed RF and SVM with the highest accuracy score among the other evaluated models. The study highlights the promising use of AI to categorize sim racers according to their technical-tactical behaviour, enhancing sim racing knowledge and know how.

Keywords: Telemetry · Sim racing · Artificial intelligence · Machine learning

1 Introduction

Esport has become more and more popular, and this trend has been going on for a while [1]. According to [2], more and more fans are anticipated to tune in to see some of the top players in the world compete in their favourite games. Over 640 million people are anticipated to watch esports worldwide by 2025. Due to such popularity, huge amounts of data on games and players is being produced. In recent years, many new data analysis techniques have been widely used for processing and analysing data to extract insights, which are of great significance for improving players' performance levels [3]. However, there is lack of tools that offer player performance feedback and suggestions for how to

improve [4]. This leads to many new opportunities for esports research to discover what makes a gamer deserving of winning.

Within data science, artificial intelligence (AI) has become a new technique for data analysis and sports performance prediction [5, 6]. AI is a branch of computer science that simulates human intelligence processes by machines specifically computer systems [7]. Machine learning is a form of artificial intelligence that automatically enhances the performance of computer systems by identifying data patterns [8]. The benefit of AI is that it can rapidly process huge volumes of data, and data analysis techniques are continually evolving, enabling users to gain crucial information that is challenging to obtain manually [9].

The application of AI in simulated racing (sim racing), which is relevant to the research reported in this work, leads to technological enhancement in the computer-based simulator and contributes to the direct improvement of the team and sim racer performance [10]. In this case, solutions, and strategies for becoming the best and the fastest driver are of utmost importance, with various methods of data analysis and data collection tools being used for sim racers. In terms of prediction and analytics, most of the existing studies rely exclusively on the in-game data analysis [11–13]. However, using only in-game data for estimating the driver's performance is a limitation for giving the team and drivers optimal feedback. Although it can give the fundamental information on the characteristics and behaviour of the driver, a huge amount of data that can be gathered from the physical world and sensors is neglected. In sim racing, the physical and control parameters of the simulation may be tracked and saved as telemetry files [10]. It allows sim racers to gather all the information provided by the vehicle and to analyze the data captured during a race or session [14]. Insights from telemetry data lead to a better understanding of the corresponding strengths and weaknesses of the car and the drivers' behaviour and can improve their performance by accurately tuning their car setup as well as informing on driving strategies and techniques [15]. Such information can supplement logs obtained from in-game data, providing additional information for the design of predictive models.

In this study, we report on predicting the driving performance in sim racing using the telemetry data collected from Assetto Corsa Competizione (ACC) and different ML methods for data analysis. While there are a few studies dealing with the prediction of a driver performance in general [14, 16], there is still a lack of research on the evaluation of driver performance relying on sim telemetry data. The abundance of telemetry data produced by sim telemetry tools enables the execution of fundamental analysis via ML methods. However, the existing research utilized only the limited number of parameters (i.e., steering wheel, throttle, pedal and brake pedal) in their analysis and most of the telemetry in-game data are omitted. To the best of our knowledge, this is the first study that applies AI techniques to telemetry data obtained from web data sources and relies on different features of telemetry to predict the performance level of the driver in sim racing. This approach may cover the lack of sufficient training data and achieve better accuracy as opposed to existing approaches which only rely on the small amounts of data collected in lab.

2 Data and Methods

In this section, we describe the data used in this research, data pre-processing, and analysis helping predict the driver's performance.

2.1 Telemetry Data

From the time a virtual gaming session is started until it is over, data which is known as game telemetry, is generated, and drivers may use this data to analyze and understand in-game behaviour [17]. Several sim racing telemetry tools have been developed and they can log, display and analyze data from control vehicle systems. For this study, we chose MoTec i2 Standard (v1.1.2.0473, Melbourne, Victoria, Australia), as it is a professional telemetric data analysis application, well known in all kinds of actual motorsport competitions and more data is available on-line with this tool.

The basis of this work is a dataset from ArisDrives MoTec Server¹, an online repository of MoTec data with different car/track combinations, freely available for anyone to upload and download files. All telemetric data have been obtained from the ACC simulator and logged through the MoTec data analysis package. We describe the analysis of the Brands hachs track in this paper mainly because we have access to more data for this track. Besides data downloaded from ArisDrives MoTec Server, we included data from 710 GTRL Racing (<https://disboard.org/>), a sim racing server hosting for AC and ACC races. The data were gathered from servers prior to September 2022, the time of preparing this article. These data are totally de-identified, available to everyone, and simply retrieved from the public domain. Additionally, all General Data Protection Regulations (GDPR) requirements have been met.

2.2 Data Processing

To extract telemetry data from MoTec log files, we have used MoTec i2 Pro (V1.1.5) available on MoTec website². Following the guideline to setup ACC workspace on MoTec [18], we configured the software and defined particular settings for section time, channel and row data using built-in maths and filter functions. To gain a better insight of data, we created three data files from each MoTec log file, exported as.csv files: 1) time report including sector/lap time in a tabular form, 2) channel report containing match statistics of different channels, and 3) time series data containing general descriptions of the event (e.g., venue, track name, vehicle, duration) as well as 84 columns corresponding to driver and vehicle in-game metrics. We used Python (3.9) as our programming language on Anaconda 3 (Spyder 5.2) platform for the implementation of the pre-processing and analysis steps. We have provided a brief description of these steps below.

The first data pre-processing step involved removing invalid laps (zero lap times caused by MoTec disconnection, and pit laps (i.e., in-lap and out-lap). A total of 802

¹ <http://motec.ascaroht.de>.

² <https://www.motec.com.au/i2/i2downloads/>.

laps remained that were subjected to additional criteria for outlier removal using the z-score normalization method [19], and those laps were temporally isolated. To determine the optimal z-score threshold, data were analysed by applying different range of values (± 1.0 , ± 2.0 , ± 3.0), and finally chose Z-score = $+3.0$ as we observed better results with such value. Note that we did not eliminate laps with Z-scores lower than -3.0 because those laps represent the very fast laps. After removing the outliers, 782 laps remained for the further analysis. In addition, we made some general descriptive analysis to find the distribution of data and to identify trends in data. Moreover, graphical and correlation analyses were performed to find highly co-related features.

3 Results

Resulting from the pre-processing step, 782 laps were used for extensive analysis. Table 1 summarizes the statistics per lap. There are some slow laps, as seen by the variance between the maximum and median (slow laps). These laps are not eliminated in the analysis that follows; instead, they are taken into account such that a certain number of slow laps is also present.

Table 1. Overall lap statistics

Number of laps	Min	Max	Mean	Std	Median
782	82.353	169.009	108.277	18.04	111.778

3.1 Performance Level Analysis

In order to identify the most important metrics affecting racing performance, first we attempted to categorize the laps into the performance levels. To do this, we used two different data sets resulting from pre-processing step, 1) laptime data; 2) channel data including laptime plus all channel data. We analysed two K-value selection algorithms, namely Elbow method [20], and Silhouette Coefficient [21] to determine the optimal number of clusters for a given track data set, then used k-means method, the most commonly used clustering algorithm in both sport science (e.g., [14, 17]) and the research outside of the context of gaming literature [22, 23]. The K-means algorithm selects the number of clusters (k) and initializes each cluster centroid in a different location within the dataset. Following initialization, centroids iteratively move and begin clustering the data based on the Euclidian distance between the data and the cluster’s mean until no further movements are required and the clusters are established [24]. The results from each dataset were consistent, representing that lap time is the most important indicator in sim racing performance.

Table 2 presents the results of clustering as well as the statistics of the corresponding groups. The cluster names refer to the lap-time, i.e., the SLOW means the slow lap-time and FAST means a fast lap-time. Violin plots displaying the means and distributions of

the three clusters are shown in Fig. 1. The thick line in the centre of each plot represents the median, while the two red lines represents interquartile range. On each side of the red line is a kernel density estimation to show the distribution shape of the data. Wider areas of the violin plot represent a higher density of laps in the cluster for the given value; whilst a less population is represented by smaller sections. All groups have a normal distribution as we observe that the values of mean and median are approximately close.

Table 2. Lap time statistics for performance levels for Brands hatches track

Group	Number of laps	Mean	Std	Min	Max	Median
SLOW	91	147.685	11.398	119.0915	169.099	146.265
MIDDLE	219	117.272	6.934	107.031	132.350	116.966
FAST	475	96.580	5.157	82.353	106.990	96.099

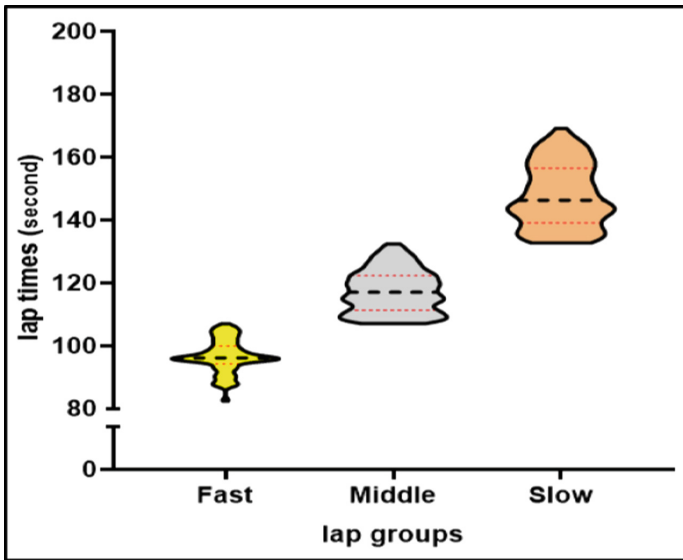


Fig. 1. Violin plots displaying the means and distributions, within each group

3.2 Feature Selection

Feature selection is the process of creating a subset from an initial feature set using ML algorithms, which removes the redundant and irrelevant features and picks the relevant features of the dataset [25]. Here, we relied on telemetry data retrieved from MoTec that contained 84 channels, including math channels (e.g., lane deviation) which we defined using built-in maths functions in MoTec. Considering the result of pre-processing step on correlation analysis, 38 channels were eliminated. The 46 remaining channels were

used for applying ML algorithms to find the most important metrics that have significant impact on the final performance. To do so, we calculated mean, median, max, standard deviation for each data channel, then conducted a bootstrapping analysis among various supervised machine learning algorithms using scikit-learn Python library. The algorithms we used are Extreme Gradient Boosting (XGBoost) [26], Support Vector Machine [27], and Random Forest [28]. These algorithms are effective for various classifications, and depending on various datasets, they each have unique attributes and performances. Figure 2 shows scatterplots depicting the accuracy of different classification methods. In comparison to other algorithms, the Extreme Gradient Boosting delivered the highest precision score among three methods in terms of mean absolute error. In addition, all algorithms got better accuracy across all included ranks (blue color in Fig. 2) compared to the results of individual rank groupings, which is reasonable given that more data yields more accurate classification outcomes.

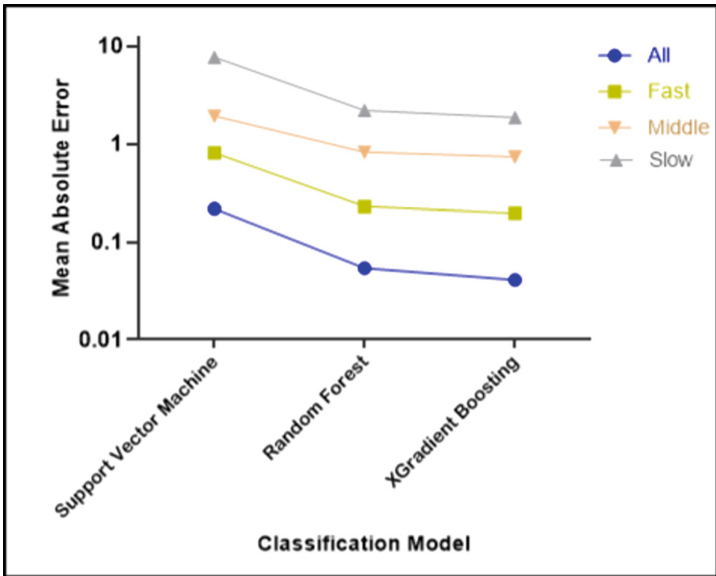


Fig. 2. Scatterplots depicting the accuracy of the classification methods for feature selection

For the classification, we divided the data into a training set (70%) and testing set (30%) in order to train the model. The model was trained using the training sets, and the accuracy of the predictions was assessed using the testing sets. The XGBoost model was able to predict the lap time with an absolute accuracy of 92.2% and an absolute error of 7.8%. A backward elimination method was used to compare classification accuracy before and after each feature was eliminated in order to assess the contribution of each feature to the classifier. The chosen features were utilized to train the classifier in phase two on the same dataset, increasing classification accuracy.

Figure 3 shows the bar graph of feature ranking for the ten most important metrics. The weights of each feature demonstrate how each feature affects the predicted lap time.

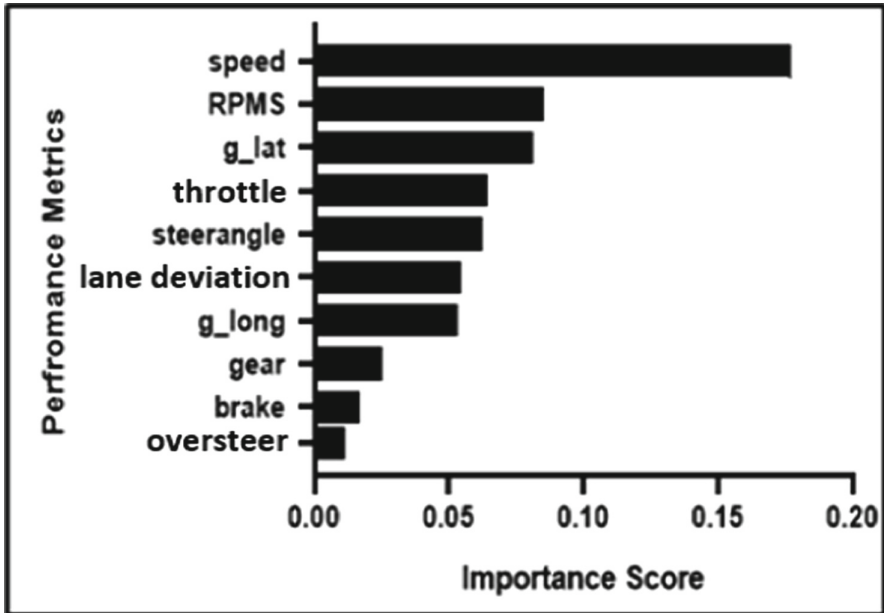


Fig. 3. Metrics found to be significant to the classification model created to predict the driver performance in sim racing

Importantly, we show the order of importance that each metric has for the prediction of lap time within our model. Here we see that all metrics that significantly were important in our classification model, are the parameters that the driver can directly control. Speed, throttle, brake and gear refer to the vehicle features. The engine RPMS describes the engine's rotations per minute and is a function of the gear used by the driver as an indication of when to change gear. Steer angle displays the angle of the steering wheel that is being input into the car at any given time. Steering error is calculation of speed, steering angle, g lateral, oversteer, brake, and throttle, indicating understeer, negative, or oversteer, positive. Lastly **g lat** and **g long** indicate the level of acceleration of the car in a specific direction, Longitudinal (forward and back) or lateral (side to side). More accurately, the higher the longitudinal g-forces are the more extreme acceleration the car has undergone which means the car has more grip when accelerating. The same can be applied through a corner using the lateral g-forces, the higher the g-force the more cornering grip the car has. From the results we observe that speed, RPMs and acceleration are the most important factors for predicting performance. These findings aid in our analysis of the different categories of drivers' driving styles. It would be also possible to focus on a specific segment rather than the data for the full lap to estimate the lap time. We defer this work as the future work.

3.3 Analysing Driving Patterns

A typical chart for analyzing driving behavior in racing is shown in Fig. 4. The figure shows the **speed**, **steering angle**, **brake pedal position**, **throttle** and **g lat** as a function of

lap distance travelled by combining the fastest and slowest laps. In order to make a better sense of the track, we incorporated sector lengths of the track into the telemetry data obtained from MoTec. To do so, we used the built-in MoTec Track Editor to determine the official sector and corner division. The sector names for each lap were then determined by processing the timeseries lap data for each lap. The vertical grey lines in Fig. 4 depict the sectors' boundaries. It is clear that all groups of drivers race in the same manner on straight sectors, while performing differently in corners. As we observe, the FAST drivers accelerate earlier and more quickly after each corner, with a sharp throttle and higher brake and stable steering control. We can observe how quickly the steering decreases when the throttle is increased in the fast laps. Additionally, how little turning is done while the brakes are fully applied. It is also obvious that fast drivers press the throttle earlier and stronger while releasing the brake later. A significant consistency can be shown when comparing driving behaviours with the feature rankings shown in

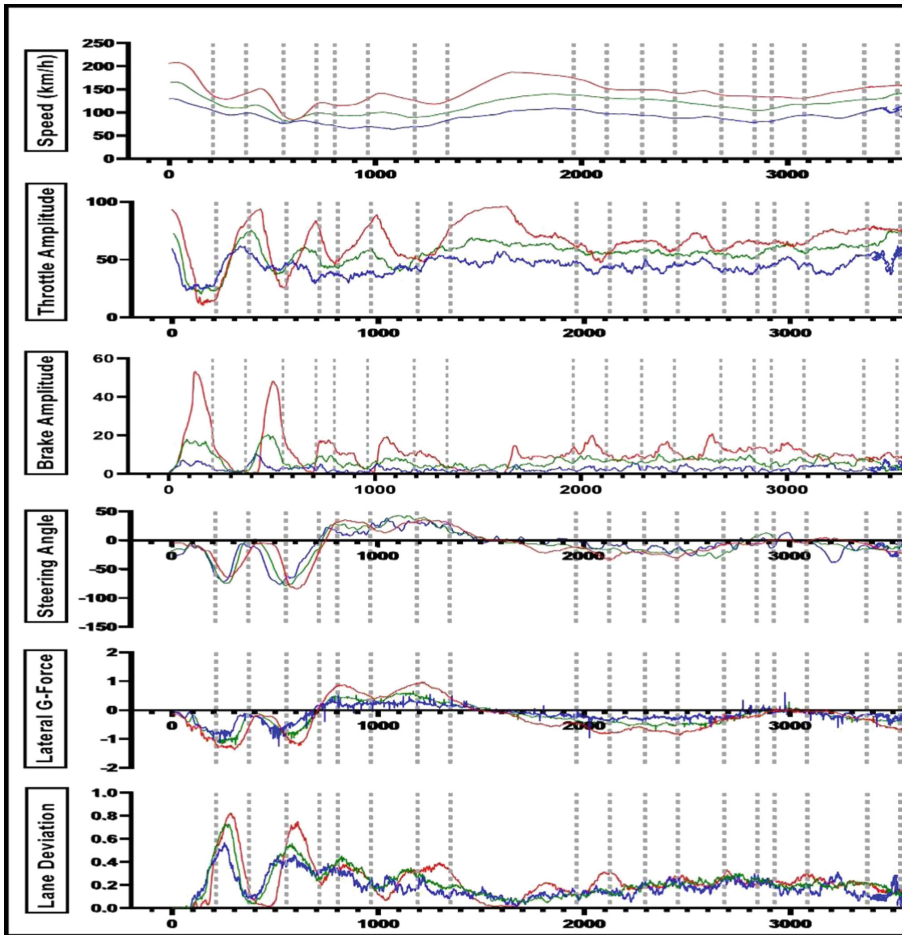


Fig. 4. Different features of driving behaviour for Fast, Slow and Middle performance group.

Fig. 4. It demonstrates that a fast driver maintains the maximum throttle for a longer period of time (producing a higher mean value) and brakes less frequently (producing a lower mean value). The brake maximum and median are greater for laps with shorter time. The similar trend is followed by the brake mean. There was no discernible trend in the acceleration characteristics (i.e., g_{lat}).

A deeper investigation needs to be carried out about the connections between all metrics that define the parameters to describe driver behaviour. It would be interesting to determine the maximum turning angle, the maximum brake, the length of full braking and the gap between the first application of the brake and the release of the throttle. This kind of analysis would also be very helpful in determining corner segments. *The breaking part of a corner*, where the car must sufficiently slow down to prepare for the turn-in point; *the racing line at a corner*, which is the segment between the turn-in point; *the apex point*, which is the inside midpoint of the corner, and *the outside apex*, where the driver must gradually accelerate out of the corner, can all be identified by studying the driving styles of professional racers.

4 Conclusion

In this work, we provided an AI enabled solution for predicting sim racing performance using telemetry data. Given telemetry data from different sources, a cluster analysis was used to divide the resulting laps into three groups based on the performance (lap-time) and then XGBoosting model was used to determine the key metrics that have more impact on the driver's performance. Overall, speed, level of acceleration, the angle of the steering wheel, RPM and number of times the driver failed in vehicle control-related (steering), were all identified as important factors that impacted driving performance across all ranked laps. The findings from our analysis provides researchers with key metrics to develop more efficient training tools and techniques to improve sim racing performance.

Further research should seek to understand more deeply the analysis of the driving style to help metrics that impact lap-time. Moreover, it would be interesting to predict the lap-time by only examining the telemetry data for a specific segment rather than the data for the entire lap. For instance, it would be interesting to explore the possibility of determining whether some parts of the lap are essential for the performance across the entire lap.

References

1. Kovács, J.M., Szabó, Á.: Esport and simracing markets—the effects of COVID-19, difficulties and opportunities. *Soc. Econ.* **44**, 498–514 (2022)
2. Statista. Esports market revenue worldwide. <https://www.statista.com/statistics/490522/global-esports-market-revenue/>
3. Chu, W.C.-C., et al.: Artificial intelligence of things in sports science: weight training as an example. *Computer* **52**(11), 52–61 (2019)
4. Roose, K.M., Veinott, E.S.: Leveling up: using the tracer method to address training needs for esports players. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. SAGE Publications Los Angeles (2020)

5. García-Aliaga, A., et al.: In-game behaviour analysis of football players using machine learning techniques based on player statistics. *Int. J. Sports Sci. Coach.* **16**(1), 148–157 (2021)
6. Mittal, H., et al.: A study on machine learning approaches for player performance and match results prediction. arXiv preprint [arXiv:2108.10125](https://arxiv.org/abs/2108.10125) (2021)
7. Lichtenthaler, U.: *Integrated Intelligence: Combining Human and Artificial Intelligence for Competitive Advantage, Plus E-Book Inside* (ePub, Mobi Oder Pdf): Campus Verlag GmbH (2020)
8. Russell, S.J.: *Artificial Intelligence a Modern Approach*. Pearson Education, Inc. (2010)
9. Li, B., Xu, X.: Application of artificial intelligence in basketball sport. *J. Educ. Health Sport* **11**(7), 54–67 (2021)
10. de Frutos, S.H., Castro, M.: Assessing sim racing software for low-cost driving simulator to road geometric research. *Transp. Res. Procedia* **58**, 575–582 (2021)
11. Cristina de Angelo, J., et al.: Video game simulation on car driving: analysis of participants' gaze behavior and perception of usability, risk, and visual attention. *Strateg. Des. Res. J.* **12**(3), 312–322 (2019)
12. van Leeuwen, P.M., et al.: Differences between racing and non-racing drivers: a simulator study using eye-tracking. *PLoS ONE* **12**(11), e0186871 (2017)
13. Shechtman, O., et al.: Comparison of driving errors between on-the-road and simulated driving assessment: a validation study. *Traffic Inj. Prev.* **10**(4), 379–385 (2009)
14. Remonda, A., Veas, E., Luzhnica, G.: Comparing driving behavior of humans and autonomous driving in a professional racing simulator. *PLoS ONE* **16**(2), e0245320 (2021)
15. Sim Racing Telemetry. <https://www.simracingtelemetry.com/>
16. Bugeja, K., Spina, S., Buhagiar, F.: Telemetry-based optimisation for user training in racing simulators. In: 2017 9th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games). IEEE (2017)
17. Odierna, B.A., Silveira, I.F.: MMORPG player classification using game data mining and K-means. In: Arai, K., Bhatia, R. (eds.) *FICC 2019. LNNS*, vol. 69, pp. 560–579. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-12388-8_40
18. Forum, K.: MoTeC telemetry and dedicated ACC workspace. <https://www.assettocorsa.net/forum/index.php?threads/motec-telemetry-and-dedicated-acc-workspace.55103/>
19. Smiti, A.: A critical overview of outlier detection methods. *Comput. Sci. Rev.* **38**, 100306 (2020)
20. Géron, A.: *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Unsupervised Learning Techniques*. O'Reilly Media, Incorporated (2019)
21. McCallum, A., Nigam, K., Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2000)
22. Abdullah, D., et al.: The application of K-means clustering for province clustering in Indonesia of the risk of the COVID-19 pandemic based on COVID-19 data. *Qual. Quant.* **56**(3), 1283–1291 (2022)
23. Ashari, I.F., et al.: Application of data mining with the K-means clustering method and Davies Bouldin index for grouping IMDB movies. *J. Appl. Inform. Comput.* **6**(1), 07–15 (2022)
24. Maheshwari, A.: *Data Analytics Made Accessible*. Amazon Digital Services, Seattle (2014)
25. Cai, J., et al.: Feature selection in machine learning: a new perspective. *Neurocomputing* **300**, 70–79 (2018)
26. Chen, T., et al.: XGBoost: extreme gradient boosting. *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4 (2015)

27. Pisner, D.A., Schnyer, D.M.: Support vector machine. In: Machine Learning, pp. 101–121. Elsevier (2020)
28. Prasad, R., et al.: Designing a multi-stage multivariate empirical mode decomposition coupled with ant colony optimization and random forest model to forecast monthly solar radiation. Appl. Energy **236**, 778–792 (2019)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Rapid Quantification of NaDCC for Water Purification Tablets in Commercial Production Using ATR-FTIR Spectroscopy Based on Machine Learning Techniques

Hamzeh Asadi^{1(✉)}, Tom O'Mahony², Julie Lambert², and Kenneth N. Brown¹

¹ School of Computer Science and Information Technology, University College Cork, Cork, Ireland

{hamze.asadi,k.brown}@cs.ucc.ie

² Medentech Limited, Wexford, Ireland

{tom.omahony,julie.lambert}@kersia-group.com

Abstract. Accurate, fast and simple quantitative analysis of solid dosage forms is required for efficient pharmaceutical manufacturing. A spectroscopic analysis in ATR-FTIR (Attenuated Total Reflection-Fourier Transform Infrared) mode was developed for NaDCC (Sodium dichloroisocyanurate) quantification. This fast and low-cost method can be used to quantify NaDCC solid dosage forms using ATR-FTIR in absorbance mode in conjunction with partial least squares. A simple sampling procedure is included in the proposed experiment by just dissolving the samples in deionized water. An algorithm pipeline is also included for data cleaning, such as outlier removal, scatter correction, scaling, and mapping of the sample's spectrum to a NaDCC concentration. In addition, a simple model based on Beer's law was evaluated on a sub-range of $1220\text{--}1830\text{ cm}^{-1}$. Furthermore, a variable selection algorithm shows minimum excipient interference from the sample matrix in addition to visual analysis. A statistical analysis of the proposed method shows that it demonstrates a promising result with a regression coefficient of 0.996 ($R^2 = 0.996$) and recovery range of 95.5%–107%. As a result of the positive correlation of ATR-FTIR with NaDCC concentration, and in conjunction with the proposed method, this can serve as a clean, fast, affordable and eco-friendly method for pharmaceutical analysis.

Keywords: Machine learning · ATR-FTIR · Chemometric

1 Introduction

One third of people worldwide lack access to safe drinking water, with significant consequences for health [26]. Ensuring the availability of water is one of the United Nations Sustainable Development Goals [25]. Beyond systemic problems of service provision, water may be contaminated or temporarily restricted

during disasters, and providing emergency supplies and short-term purification treatments are essential. Water treatment and disinfection can be accomplished by methods including boiling, filtration, distillation and chlorination [24]. Chlorination is fast and effective, and can be delivered as soluble tablets or chemicals. Water disinfection tablets, with NaDCC (Sodium dichloroiso-cyanurate) as their main chemical component, have been shown to outperform iodine tablets for biocidal and cysticidal treatments [11, 12].

Data-driven techniques for spectroscopy analysis are now common in chemometrics, and classical machine learning approaches are competitive with data-heavy neural network methods [19]. In this paper, we propose a new method which exploits machine learning and data analytics methods to quantify NaDCC, as a replacement for the slow and expensive laboratory techniques of titration [19]. Partial Least Squares regression (PLS) [10] is applied to the tablet formula solution spectra. The method is fast enough to perform during processing as part of a batch failure prevention test, and does not require significant additional expertise on behalf of the operators. Further, we show that its accuracy is within the same 90% to 110% recovery rate as the current method.

1.1 Pharmaceutical Background

Tablets are the most common solid dosage form for pharmaceutical products due to inexpensive manufacturing, packaging, transportation costs and popularity [13]. Active pharmaceutical ingredient (API) quantification in quality control is an integrated part of the tablet manufacturing life-cycle [6]. Aside from the formula, the concentration of each component in a solid dosage form is determined by other process factors, including powder flow, particle size distribution, dosing depth and turret speed. Blend uniformity is an important quality test that checks for uniform distribution of content in the mixture which has a direct impact on the average quantity of API per tablet [2].

HPLC and capillary electrophoresis [9] are extensively used and considered as standard methods for quantitative testing of different pharmaceutical formulas. These methods require significant amounts of sample preparation and analysis time, in addition to the very high cost of these instruments. Therefore, the need for quick, cost-effective, and easy-to-use technologies for quality control, such as FTIR spectroscopy arises [15]. ATR-FTIR (Attenuated Total Reflection-Fourier Transform Infrared) can be used instead of standard methods to assay API. Different materials absorb infrared light in different patterns depending on whether they have a covalent bond that vibrates at a specific frequency [17]. This enables it to detect molecular vibrations and identify specific chemicals [1]. Infrared spectroscopy has been widely investigated for both qualitative and quantitative analysis of pharmaceutical analysis [8, 15].

In a batch manufacturing process, to evaluate the concentration of NaDCC in a batch, a sample of tablets is normally taken during production and after it is completed. In case the concentration is not within specification, three more samples will be collected, and if the concentration is still out of specification, the batch will be rejected. The time required to perform any of these assays is around 30 min, which makes it impossible to use them for in-process quality

control. Since the tablet formulation is the same for each product, and sometimes batch failures occur, we should investigate the production process to determine the root cause. The two main steps in the process are blending and compression, and since the blending configuration does not change during a batch, compression is the likely cause of any batch failure. The rotary tablet press machines used for compression must be clean and undergo regular maintenance. There may be times when a rotary tablet press is used for another product, in which case the configuration will change according to the new product's requirements. In addition, depending on the product, there is also quality control during the manufacturing process. A change of configuration will be made to the rotary tablet press if the results of these quality tests fail to meet specifications. These in-process quality checks are entangled with each other. Turret speed has a positive association with weight variation and a negative correlation with die filling, resulting in weight and hardness that are both out of specification. Also, hardness has a negative correlation with paddle speed and positive correlation with die depth [21]. In order to bring all of these in-process tests into specification the operator might need to configure the rotary tablet press so that some of these metrics are at their specification boundary. In addition, some of these in-process quality checks directly influence the concentration of active biocides such as NaDCC. Weight and NaDCC concentration are positively correlated, for instance. For all of these reasons, it would be beneficial to find an alternative approach to NaDCC quantification as part of the manufacturing process control.

The ATR-FTIR spectrum of solution of water purification tablets and chemometrics techniques were used to explore NaDCC quantification in this work. In simple terms, samples are prepared by dissolving solid dosage forms in deionized water, spectrum recordings are made from that solution, and the concentration is quantified based on the pipeline proposed for the prediction algorithm. The proposed economical and quick approach has potential as an alternative to the current techniques which are slow, require detailed method development and tedious sample preparation techniques.

2 Experiment

Medentech, Wexford, Ireland, supplied three excipients and one API: sodium bicarbonate, sodium carbonate and adipic acid as excipients, and NaDCC as an active biocide. For a successful measurement, various factors such as humidity, content distribution uniformity and temperature must be taken into account. To circumvent the difficulties noted in Sect. 1.1, samples were dissolved in deionized water obtained from an Elix[®] Advantage 3 Water Purification System. The deionized water used had a conductivity lower than $0.2 \mu\text{S}/\text{cm}$ at 25°C , the resistance was greater than $5 \text{ m}\Omega\text{-cm}$, and the organic carbon content was less than 30ppb. After each sample recording, ESEPT[®] alcohol-based Isopropanol 70 v/v was used to clean the surface of the ATR accessory.

A basic sampling approach was employed to establish a quick and easy procedure which can be used for in-process quality control. Samples were prepared in disposable plastic containers. The scale was calibrated to zero while the sampling

container was on the scale. Each component was removed from its bag and placed in the container with a clean spatula until the needed amount was attained. All of the components were weighed using the same approach. Samples were then gently placed into a 500 ml beaker containing 200 ml of deionized water. The samples were then dissolved thoroughly in deionized water in order to form a homogeneous solution. This was achieved by sonicating them for 2 to 10 min, depending on the sample. For both formulas, the above steps were followed to prepare the sample. Samples begin with zero concentration of NaDCC, and the quantity of excipients was decreased while gradually increasing the amount of NaDCC, so that the overall amount of the blend (combination of excipients plus NaDCC) remained constant. Following sonication and homogenization, one drop of solution was taken with a pipette for examination. Next, the same sample was diluted with 100 ml of deionized water and was processed as an independent sample. The dilution process was repeated three times, and each diluted sample was considered as a separate sample. The beakers and tools were thoroughly cleaned after each sample and re-used only after they were completely dry. Twenty samples, each diluted four times, were collected.

In this experiment, a Compact Alpha P FT-IR Spectrometer (Billerica, Massachusetts, United States), equipped with a diode laser with spectral stability and high wavenumber accuracy was used. All measurements were taken by a high-performance Platinum-ATR accessory featuring a monolithic trapezoid shape diamond crystal. Three spectra were acquired for each sample, and each spectrum was scanned 24 times. The spectral range was $4000\text{--}400\text{ cm}^{-1}$ and the Spectral Resolution was 2 cm^{-1} . Each spectrum gives 1776 data points on 2.04 wave-number intervals. Dissolution of the samples was performed using an ultrasonic bath, Decon FS200. Each sample is sonicated for 2–10 min to produce a homogenous solution. A magnetic stirrer was used for between three and twelve minutes, depending on adipic acid concentration.

Blends were created based on the formula for water purification tablets produced by Medentech. These blends were prepared on a small scale, so each sample weighed approximately 20 g. There are three excipients examined within each blend (sodium bicarbonate, sodium carbonate, and adipic acid) and one active pharmaceutical ingredient (NaDCC). Having performed this step, the concentration of NaDCC in each sample was established. The next step was to dissolve the blends in deionized water. Each blend (20 g) was dissolved in 200 ml deionized water in a 500 ml beaker. A homogeneous solution was obtained by sonicating the samples for 2–10 min. Additionally, each sample was diluted three times with 100 ml of deionized water each time in order to collect more data. The spectrometer prism was cleaned with Isopropanol alcohol (IPA) after recording each sample. The sampling surface of the spectrometer needs 15 s to completely dry out. Before sampling, the spectrometer was set at a resolution of 4 cm^{-1} and a range between 4000 cm^{-1} and 400 cm^{-1} . A disposable plastic pipette was used to place one drop of solution on the prism of the spectrometer to record sample spectra. Three scans were conducted on each sample. The recording of any sample was preceded by a background scan. There were 24 scans each for the background spectrum and the sample spectrum.

3 Results and Discussions

Excipients are added to an active pharmaceutical component (NaDCC) in Medentech's water purification solid dosage forms for a variety of purposes. The basic goal is to enhance the formulation's volume, make packaging and transportation easier, and impart desirable qualities [23]. The proportion of each excipient varies depending on the product's use case. In Fig. 1, four different sample spectra are shown: water, pure NaDCC, a normal sample based on Medentech product formulation, and a sample containing only excipients. In the range 1000 cm^{-1} to 2000 cm^{-1} , both NaDCC and the excipients show two additional peaks. In particular, NaDCC shows a unique peak at approximately 1250 cm^{-1} . Therefore, the following methods focus on that $1000\text{--}2000\text{ cm}^{-1}$ range.

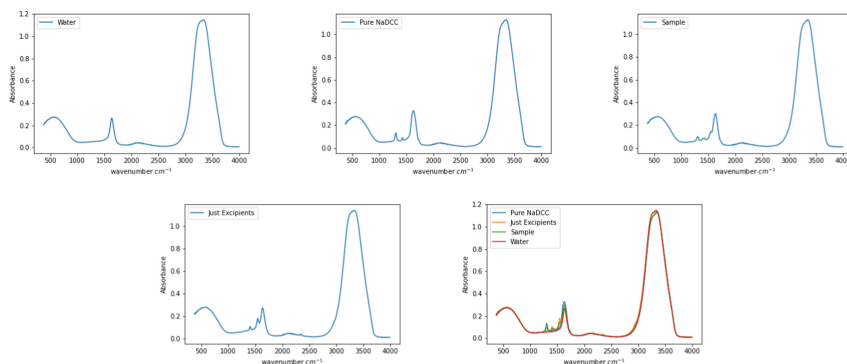


Fig. 1. Different concentrations of NaDCC and excipients in deionized water.

3.1 ATR FTIR Region Selection for Calibration Models

Figure 2 illustrates various sample solution spectra with different NaDCC concentrations. We see two peaks at approximately 1400 and 1550 where the absorbance seems to be inversely proportional to the NaDCC concentration. This is because the absorbance of the excipients at these wavenumbers is higher than for NaDCC (as seen in Fig. 1). Peak intensity of the spectrum (A) depends on molar absorptivity (ϵ), path length (b) and concentration (c) ($A = \epsilon bc$). If two substances have absorption at the same wavenumber and they are both present in a sample the response of the instrument depends on the concentration of each substance and their molar absorptivity. Since the amount of powder dissolved in the deionized water was held constant at 20g, when the concentration of NaDCC is increased, the concentration of excipients was decreased, and so the overall absorbance of the sample decreases at those wavenumbers. We therefore focus on the outer two peaks, where the absorbance of NaDCC is higher than for the

excipients, again as seen in Fig. 1. For these two peaks, absorbance is approximately linear with respect to concentration, and so Beer’s law [22] can be applied. Figure 2 illustrates where the height fluctuates with NaDCC and there are four bands evident. This is confirmed through formal analysis, both univariate, in the spectrum range 1220–1830, and multivariate, in range 400–4000.

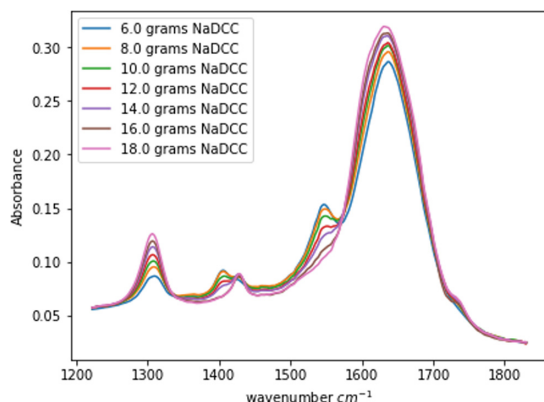


Fig. 2. ATR FT-IR spectrum of NaDCC concentrations in 200 ml of deionized water.

The response of a calibrated univariate Beer’s law model in the spectra in the range 1220–1830, according to peak height - with or without baseline modifications - is shown in the Table 1. The average recovery column of this summary shows the average recovery of at least 5 samples with the same concentration of NaDCC. In the table, we can see that the baseline correction [14] slightly enhances the accuracy and precision of Beer’s law calibrated models. The results confirmed the visualization assessment shown in Fig. 2, which shows a high correlation between wavenumbers in the range 1220–1830 and the concentration of the target analyte. Wavenumber 1281 has the highest correlation coefficient (0.9971) and an average recovery rate of 100.93.

3.2 PLS Calibration Model

Partial least squares regression plays an important role in chemistry [10] and high-dimensional collinear data processing. This technique resolves the multicollinearity problem associated with most spectroscopy data sets by mapping the acquired data into a set of latent variables [4] of much smaller size. A data preprocessing step is typically incorporated into statistical analysis and modeling along with the main prediction model, so preprocessing steps are vital to getting the most out of machine learning algorithms. Some algorithms might require all of these preprocessing steps, while others might require only a subset.

Table 1. The peak height selection of FTIR area and wavenumber.

Wavenumber	Baseline criteria	Regression	Corr. coefficient	Recovery Rate
1283	None	$4.210x + -0.257$	0.9959	100.67
1285	None	$3.595x + -0.22$	0.9962	100.93
1523	None	$-2.338x + 0.3$	0.9964	99.58
1534	None	$-1.492x + 0.233$	0.996	100.45
1536	None	$-1.403x + 0.226$	0.9961	100.76
1279	400–4000	$5.2352x + 0.1171$	0.9967	101.34
1281	400–4000	$4.4754x + 0.0998$	0.9971	100.93
1283	400–4000	$3.8356x + 0.0854$	0.9965	100.82
1536	400–4000	$-1.4613x + 0.1403$	0.9967	100.62
1538	400–4000	$-1.3854x + 0.1394$	0.9963	100.97

We use an algorithm pipeline of outlier detection, smoothing, scatter correction, variable selection, and PLS, to produce the NaDCC quantification from the input spectrum. Standard procedures for developing algorithms in machine learning and data analysis are used in the development of this pipeline. Three main steps comprise the general pipeline: data collection, preprocessing, and model prediction. An ATR-FTIR sample scan was done as a first step, based on the prepared sampling method, to gather raw data. Following that, general and specific data preprocessing steps are undertaken such as normalization, artifact removal, smoothing, and variable selection. In this study, PLS was calibrated using k-fold cross validation, and evaluated using R-square for unseen samples.

When a data point does not fit the general trend, it is usually considered an outlier. A model is not able to explain outliers well because they are associated with large errors in the cost function. There are several causes of outliers, including measurement error, sampling error, inaccurate recording, or incorrect assumptions about the distribution. In the outlier identification approach, the Q-residual, Hotelling T-Square, which is capable of reducing the computation time without compromising accuracy [16], and a 95% confidence interval were employed. According to Fig. 3, 19 cases were detected as outliers. Additionally, an approach can be used for eliminating outliers simply by examining samples visually. Figure 4 shows the effect of one poor quality scan (with three scans per sample), where the curve can be seen to deviate significantly from the curves for the other sample scans. Any given model will contain a data point that has a high Q-residual in comparison to the corresponding residuals of other data points, so there will always be some data points with a large residual error. In this study, model calibration by removing outliers based on the output of Q-residual and Hotelling T-Square approaches was applied to avoid the proposed model under-performing.

Signal smoothing is just as critical to the pre-processing of spectral data as removing outliers from the data points. The random noise in the data can be

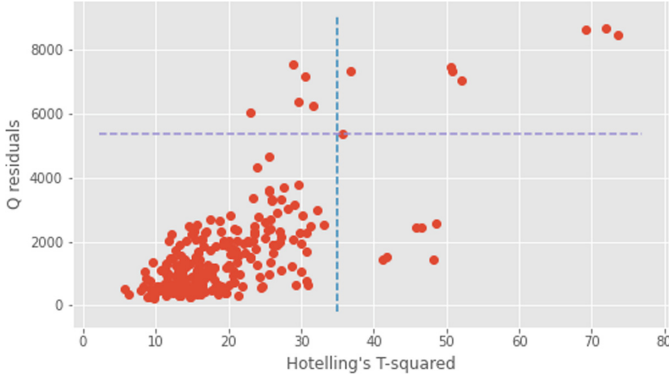


Fig. 3. Q-residual and Hotelling T-Square metrics Outlier Detection

reduced using a smoothing technique. Many methods can be employed to accomplish this task, such as Savitzky-Golay, and Fourier spectral smoothing [27]. In this model, for each point in the sample, a neighbourhood of points were selected, which is called the window size, and then a polynomial model is fitted to the selected data points in the window. This data point is replaced with the corresponding value of the fitted curve at that point in order to provide the smooth version of the data point. The Savitzky-Golay results in the suggested pipeline design were calculated using a window size of 11 and a polynomial degree of 3 (Fig. 5).

Specular reflections and diffuse reflections constitute a spectrum. Due to the sample's chemical composition, different wavelengths of incident light are absorbed differently by the sample, resulting in different spectral shapes. Additionally, particle sizes and path lengths also affect spectra. Scattering can be used to eliminate errors caused by sample geometry and morphology which have no connection to chemical composition. In essence, removing all these undesirable effects before computing the quantity of interest produces a better model. There are two tools that can be used in spectroscopy to correct scatter data—the standard normal variations (SNVs) and the multiplicative scatter correction (MSCs) [7]. The particle size and path length effects are expected to have a zero mean normal distribution in each sample, and these scatters should be reduced significantly by averaging all samples. An average spectrum is calculated from all the samples and a linear regression model is fitted to the calculated spectrum as an independent variable and to each sample as a dependent variable. Equation 1 illustrates the general procedure for the MSC.

$$X_m = \frac{1}{N} \sum_{i=1}^N X_i \quad X_i = b_i + m_i \times X_m \Rightarrow X_i^{msc} = \frac{X_i - b_i}{m_i} \quad (1)$$

In the SNV method, used in this study (Fig. 5), there is no reference to regress the input spectrum against. We use $X_i^{snv} = \frac{X_i - \bar{X}_i}{\sigma_i}$.

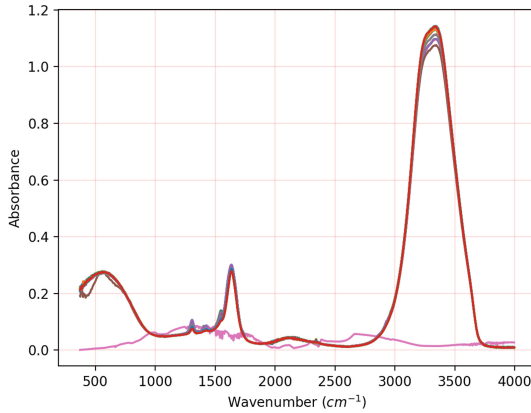


Fig. 4. Examples of a misaligned sample

Pharmaceutical laboratories heavily rely on instruments that generate large amounts of data. It is fairly common for a laboratory instrument to generate data that has thousands of variables; for example a typical FTIR instrument records absorbance at more than 10000 frequencies. However the full amount of this data is not useful in many of scenarios and normally there is considerable redundancy and correlation among these variables. Since high dimensional data leads to problems related to the curse of dimensionality in machine learning, extracting and compressing these variables in such a way that keeps essential information is vital. This compression or extraction may be achieved by combining different variables to get a more informative variable (such as Principal component analysis-PCA) or by selecting a variable from a set of variables that provide more information for the task in hand.

Due to the simplicity of the forward variable selection algorithm, it is applied to this problem in order to ensure that important wavelengths are separated from less informative wavelengths within spectral measurements [18]. The wavelength bands containing most of the signal related to the analyte can often be hard to predict in advance, especially in visible and infrared spectroscopy. A measurement of all bands that the instrument is capable of will be made in the first step, followed by a determination of vital bands. In other words, the wavelength bands will make better-quality models stand out.

Preprocessing discards one wavelength at a time. An entire spectrum calibration model will be created, then the wavelength associated with the regression coefficient with the smallest absolute value will be eliminated. By calculating the model's mean square error in each iteration, the performance will be evaluated. A given number of wavelengths will be discarded in conjunction with the minimum model mean square error. A total of 1292 wavelengths were discarded and the optimised MSE was 0.006929.

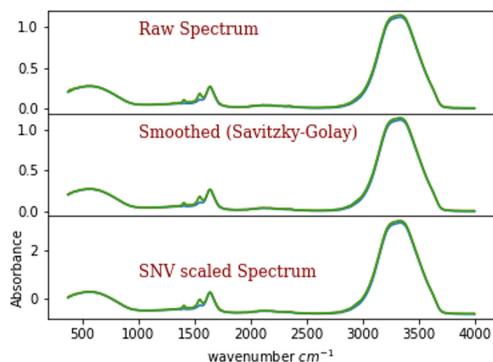


Fig. 5. Spectrum smoothing and scatter correction. The top graph displays raw spectrum in absorbance mode with no processing; the middle graph is obtained by applying a savitzky golay filter with window size 11 and degree 3; the bottom graph illustrates the spectrum after applying the standard normal variations scatter correction.

The estimation part of the pipeline consists of partial least square regression (PLSR) as its core algorithm. The concentration is predicted by this algorithm after the preprocessing stage, where the spectrum is mapped to concentration. In partial least square regression, multiple linear regression is performed, which builds a linear model, $Y = XB + E$, which maps latent variables (LV) onto dependent variables. The method is designed to maximize correlation between the selected LVs and the target variable. The reason that PLSR is superior to PCR (principal component regression) is that it simultaneously extracts variability from input data (X) and correlates it with target data (Y). In PCR, the LVs of PCA are used to account for variation in independent variables, but they may not affect the dependent variable directly.

In spectroscopy analyses, three variables are involved: X, Y, and E. X represents spectra, Y represents quantities or quantity sets, and E represents errors. In mathematical terms, PLSR can be considered as an optimization problem with the objective: $\arg \max_{w_i} \text{cov}(XW, Y)$ $i = 1, \dots, A$. The analysis of the collected data was based on the PLSR model. Since 80 samples were gathered and each sample was scanned three times, a total of 240 samples were collected. Data was split as a typical procedure in machine learning and chemometric analysis, ten percent for testing, and ninety percent for calibration. During variable selection 1776 variables were reduced to 484, which were used as input for model calibration, reserving 24 samples as the test set. Variables were selected based on the magnitude of mean square error produced by the PLSR model on the calibrated dataset. The optimal number of variables was associated with the minimum error of the model. The model was calibrated based on 10-fold cross-validation in the selection of variables as well as in the training process for predicting the interest target. In addition, the optimal number of PLS components for the PLS model had to be estimated. The search for a model includes all potential component combinations between 1 and 30. On the other hand,

according to Fig. 6, PLS models with 15 components produce the smallest average square error, which aligns with the optimal number of components in the variable selection stage.

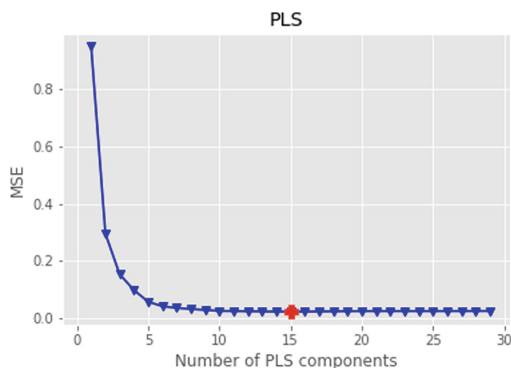


Fig. 6. Optimal number of component for PLSR

The minimum error of the calibration model, illustrated in Fig. 6, is responsible for the good dispersion of predicted values of NaDCC concentration around the regression line. In addition to determining how well the calibration model fits, a second factor to consider is the square of correlation coefficient (R^2). This is a measure of how well the independent variables can explain the variation in dependent variables. The R^2 value, 0.9961, is shown in Fig. 7. The value is over 0.99, which is representative of a high degree of linear correlation between the predicted and the ground truth values.

NaDCC Quantification Result. This is the last step of the algorithm pipeline, which evaluates the algorithm’s performance capabilities and statistical analysis. Seven test groups are used to perform this evaluation. Each group consists of several samples with the same NaDCC concentrations, but each group’s number of samples varies. Occasionally, after dilution of a sample, the concentration of API in two different samples was equal. The purpose of the diluting procedure was to create additional data. A summary of the results of our model is presented in Table 2. Within the 7 test groups, the recovery average ranged from 95.46% to 107%, which is in complete agreement with the baseline (titration) that we are comparing to which has a target recovery range of 90% to 110%. Another important evaluation metric in chemometric analysis is the limit of detection (LoD) and limit of quantification (LoQ). LoD is the least quantity of analyte that can be consistently distinguished from zero concentration, whereas LoQ is the smallest value of analyte that can be quantified [5, 20].

The LoD in relation to partial least square regression has been calculated using equation 3 from Franco et al [3]. In their proposed LoD formula, excipient-containing samples will be treated as samples with zero concentration. We will

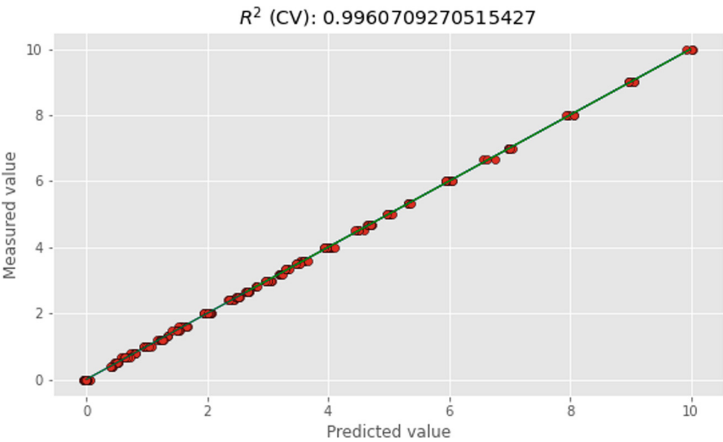


Fig. 7. Test Coefficient of determination ($R^2 - Score$)

calculate LoD using $LoD = (t_{\alpha,\nu} + t_{\beta,\nu}) \times \sqrt{var(y_0)}$, in which y_0 represents a sample with a concentration of zero, and $t_{\alpha,\nu}$ and $t_{\beta,\nu}$ represent the parameters of a t distribution that has ν degrees of freedom. The limit of detection has been calculated at 0.0849 mg/ml, while the limit of quantification (LoQ) has been calculated at 0.283 mg/ml.

Table 2. Samples

Sample	NaDCC measured	NaDCC predicted	Recovery average	Standard deviation
1	1.000	0.955	95.462%	0.006
2	1.500	1.616	107%	0.068
3	1.600	1.536	96%	0.095
4	2.000	1.985	99.237%	0.015
5	3.200	3.360	105.012%	0.066
6	5.000	4.909	98.188%	0.057
7	6.000	6.245	104.076%	0.028

4 Conclusion

This study proposes applying data analysis techniques directly to the FTIR spectrum of chemical compounds (ATR-FTIR) to quantify APIs of interest. The method is to eliminate the complicated traditional, time-consuming titration methods, simplify sampling procedures, and expedite result extraction for in-process quality control. The method’s low cost and the elimination of toxic chemicals from titration methods makes it environmentally friendly. By simply

dissolving NaDCC tablets in deionized water and using it as an ATR-FTIR sample, the proposed method successfully quantifies NaDCC concentrations. According to an evaluation of the proposed pipeline with $R^2 = 0.996$ and recovery range of 95.5%–107%, which completely aligned with the recovery range required in the case study. Additionally, the process can be completed in less than 3 min, making it suitable for use as an in-process quality control method. The technique could potentially replace the existing labor-intensive and time-consuming titration technique for analysis of NaDCC concentrations.

Acknowledgements and Data Availability. This publication has emanated from research supported in part by Science Foundation Ireland under Grant number 16/RC/3918 which is co-funded under the European Regional Development Fund. The datasets generated during the current study are available in [google drive: NaDCC.db](#).

References

1. Abd El-Rahman, M.K., Eid, S.M., Elghobashy, M.R., Kelani, K.M.: Inline potentiometric monitoring of butyrylcholinesterase activity based on metabolism of bambuterol at the point of care. *Sens. Actuators B: Chem.* **285** (2019)
2. Akseli, I., et al.: A practical framework toward prediction of breaking force and disintegration of tablet formulations using machine learning tools. *J. Pharm. Sci.* **106**(1), 234–247 (2017)
3. Allegrini, F., Olivieri, A.C.: IUPAC-consistent approach to the limit of detection in partial least-squares calibration. *Anal. Chem.* **86**(15), 7858–7866 (2014)
4. Boulesteix, A.L., Strimmer, K.: Partial least squares: a versatile tool for the analysis of high-dimensional genomic data. *Brief. Bioinform.* **8**(1), 32–44 (2006)
5. Currie, L.A.: Nomenclature in evaluation of analytical methods including detection and quantification capabilities (IUPAC recommendations 1995). *Pure Appl. Chem.* **67**(10), 1699–1723 (1995)
6. Davidson, I.E.: 11 - setting up specifications. In: Ahuja, S., Scypinski, S. (eds.) *Handbook of Modern Pharmaceutical Analysis, Separation Science and Technology*, vol. 3, pp. 387–413. Academic Press (2001)
7. Dhanoa, M., Lister, S., Sanderson, R., Barnes, R.: The link between multiplicative scatter correction (MSC) and standard normal variate (SNV) transformations of NIR spectra. *J. Near Infrared Spectrosc.* **2**(1), 43–47 (1994)
8. Eid, S.M., Soliman, S.S., Elghobashy, M.R., Abdalla, O.M.: ATR-FTIR coupled with chemometrics for quantification of vildagliptin and metformin in pharmaceutical combinations having diverged concentration ranges. *Vib. Spectrosc.* **106**, 102995 (2020)
9. ElBagary, R.I., Azzazy, H.M., ElKady, E.F., Farouk, F.: Simultaneous determination of metformin, vildagliptin, and 3-amino-1-adamantanol in human plasma: application to pharmacokinetic studies. *J. Liquid Chromatogr. Relat. Technol.* **39**(4), 195–202 (2016)
10. Geladi, P., Kowalski, B.R.: Partial least-squares regression: a tutorial. *Anal. Chim. Acta* **185**, 1–17 (1986)
11. Gerba, C.P., Johnson, D.C., Hasan, M.N.: Efficacy of iodine water purification tablets against cryptosporidium oocysts and giardia cysts. *Wilderness Environ. Med.* **8**(2), 96–100 (1997)

12. Kgabi, N., Mashauri, D., Hamatui, N.: Utilisation of water purification “tablets” at household level in Namibia and Tanzania. *Open J. Appl. Sci.* **4**, 560–566 (2014)
13. Kottke, M., Rudnic, E.: Tablet dosage forms. In: Banker, Rhodes (eds.) *Modern Pharmaceutics*, pp. 458–532. CRC Press (2002)
14. Liland, K.H., Almøy, T., Mevik, B.H.: Optimal choice of baseline correction for multivariate calibration of spectra. *Appl. Spectrosc.* **64**(9), 1007–1016 (2010)
15. Mallah, M.A., Sherazi, S.T.H., Bhanger, M.I., Mahesar, S.A., Bajeer, M.A.: A rapid Fourier-transform infrared (FTIR) spectroscopic method for direct quantification of paracetamol content in solid pharmaceutical formulations. *Spectrochim. Acta Part A Mol. Biomol. Spectrosc.* **141**, 64–70 (2015)
16. Mashuri, M., Ahsan, M., Lee, M.H., Prastyo, D.D.: Wibawati: PCA-based hotelling’s T² chart with fast minimum covariance determinant (FMCD) estimator and kernel density estimation (KDE) for network intrusion detection. *Comput. Ind. Eng.* **158**, 107447 (2021)
17. Mayerhöfer, T.G., Popp, J.: Beer’s law - why absorbance depends (almost) linearly on concentration. *ChemPhysChem* **20**(4), 511–515 (2019)
18. Mehmood, T., Liland, K.H., Snipen, L., Sæbø, S.: A review of variable selection methods in partial least squares regression. *Chemom. Intell. Lab. Syst.* **118**, 62–69 (2012)
19. O’Connell, M.L., Howley, T., Ryder, A.G., Leger, M.N., Madden, M.G.: Classification of a target analyte in solid mixtures using principal component analysis, support vector machines, and Raman spectroscopy. In: *Opto-Ireland 2005: Optical Sensing and Spectroscopy*, vol. 5826, pp. 340–350. SPIE (2005)
20. Saadati, N., Abdullah, M.P., Zakaria, Z., Sany, S.B.T., Rezayi, M., Hassonizadeh, H.: Limit of detection and limit of quantification development procedures for organochlorine pesticides analysis in water and sediment matrices. *Chem. Cent. J.* **7**(1), 63 (2013)
21. Schomberg, A.K., Kwade, A., Finke, J.H.: The challenge of die filling in rotary presses-a systematic study of material properties and process parameters. *Pharmaceutics* **12**(3) (2020)
22. Swinehart, D.F.: The beer-lambert law. *J. Chem. Educ.* **39**(7) (1962)
23. Tekade, R.: *Basic fundamentals of Drug Delivery*. Academic Press, an Imprint of Elsevier, London, San Diego (2019)
24. Torricelli, A.: Drinking water purification. *Adv. Chem.* **21**, 453–465 (1959)
25. United Nations Department of Economic and Social Affairs: Goal 6: Ensure availability and sustainable management of water and sanitation for all
26. World Health Organization: 1 in 3 people globally do not have access to safe drinking water. <https://tinyurl.com/3dkfupct>
27. Zhao, A.X., Tang, X.J., Zhang, Z.H., Liu, J.H.: The parameters optimization selection of Savitzky-Golay filter and its application in smoothing pretreatment for FTIR spectra. In: *2014 9th IEEE Conference on Industrial Electronics and Applications*, pp. 516–521. IEEE (2014)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





WiFi-Based Human Activity Recognition Using Attention-Based BiLSTM

Amany Elkelany^(✉) , Robert Ross , and Susan McKeever

Technological University Dublin, Dublin, Ireland

amany.elkelany@adaptcentre.ie

Abstract. Recently, significant efforts have been made to explore human activity recognition (HAR) techniques that use information gathered by existing indoor wireless infrastructures through WiFi signals without demanding the monitored subject to carry a dedicated device. The key intuition is that different activities introduce different multipaths in WiFi signals and generate different patterns in the time series of channel state information (CSI). In this paper, we propose and evaluate a full pipeline for a CSI-based human activity recognition framework for 12 activities in three different spatial environments using two deep learning models: ABiLSTM and CNN-ABiLSTM. Evaluation experiments have demonstrated that the proposed models outperform state-of-the-art models. Also, the experiments show that the proposed models can be applied to other environments with different configurations, albeit with some caveats. The proposed ABiLSTM model achieves an overall accuracy of 94.03%, 91.96%, and 92.59% across the 3 target environments. While the proposed CNN-ABiLSTM model reaches an accuracy of 98.54%, 94.25% and 95.09% across those same environments.

Keywords: WiFi · Channel State Information (CSI) · Human Activity Recognition (HAR) · Deep learning · Convolutional Neural Network (CNN) · Long Short Term Memory (LSTM)

1 Introduction

With the rise of the Internet of Things, human sensing techniques have gained prominence with numerous persuasive applications such as human activity recognition, smart healthcare, safety surveillance, and ubiquitous interaction all gaining popularity. Previously established research approaches to HAR have fallen into three categories of techniques: vision-based, low-cost radar-based and wearable sensor-based approaches [11]. However, there are many limitations to using traditional techniques for HAR. Vision-based approaches can only operate in a limited number of line-of-sight (LOS) environments. They are vulnerable to lighting conditions, obstacles, as well as the problem of dead angles. Vision systems also raise many issues related to human privacy. Meanwhile, low-cost radar-based systems often circumvent the problems of privacy but have short operational distances, typically in the tens of centimetres. Although wearable

sensor-based solutions achieve fine-grained behaviour awareness, their high cost and limitations with respect to real-time nature make them unsuitable for some applications (e.g. survival applications).

From a different technology perspective, WiFi technology has opened the way for numerous technological revolutions. It has an impact on almost every aspect of modern life, especially in indoor environments. It does not require wearable sensors, yet it has important features that make it a desirable choice when compared to other sensing technologies. WiFi signals travel freely in the atmosphere due to the unguided property of radio signal propagation and may be reflected by the wall and/or other objects in an environment. Antennas at the receiver may thus receive signals from two or more paths, a phenomenon known as a multi-path phenomenon. The core idea of human activity recognition using WiFi signals is that moving bodies affect multi-path propagation and that different moves have different discernible effects. WiFi-based technology typically includes two types of wireless signals: RSSI (Radio Signal Strength Indicator) and CSI (Channel State Information) [11]. RSS describes coarse-grained information about the communication link, whereas CSI describes fine-grained information about the state of the communication channels. RSS is less stable than CSI because it cannot capture dynamic changes in the signal while the activity is being performed. In recent years, CSI has received more attention than RSS as a more informative specification of WiFi signals for different sensing applications such as human activity recognition, human presence detection, fall detection, gesture recognition, people counting, and so on. Such models take advantage of the fact that when a person moves between the transmitter and receiver, the reflected wireless signals from the body create a distinct pattern [12]. CSI is a complex value representing the amplitude and phase information of multiple paths. From this, we can then analyze the received signals in terms of amplitude and phase shift to recognize human activities. *In this paper, we propose a WiFi CSI-based human activity recognition approach using two deep learning models to recognize predefined different human activities.*

Despite the numerous advantages of using CSI to recognize human activities, there are some drawbacks. Firstly, the complicated relationship between CSI measurements and human activities is challenging to model precisely using statistical models or traditional machine learning algorithms. Secondly, WiFi network settings may affect network performance. Thus, some WiFi sensing applications necessitate a high CSI measurement frequency in order to achieve optimal performance. This could result in increased overhead for WiFi communications, resulting in decreased sensing performance and efficiency.

In light of the above, the contributions in this paper are summarized as follows: First, we propose a human activity recognition framework using Attention-based Bidirectional Long Short Term Memory (ABiLSTM) and Convolutional neural network ABiLSTM (CNN-ABiLSTM). Second, we conduct experiments on three different indoor environments and evaluate the proposed framework across these environments. The performance evaluation shows that our system is significantly robust and generic enough to be used in training for other environments. Third, we investigate the impact of the spatial environment on the

trained models and their ability to transfer across environments using the transfer learning approach.

The rest of the paper is organized as follows. Section 2 discusses work related to human activity recognition using WiFi Technology. Section 3 describes the proposed methodology. Later, in Sect. 4 we describe the evaluation and results of the proposed approach. Finally, we conclude the work and explain the future work in Sect. 5.

2 Related Work

WiFi-based human activity recognition has gained tremendous attention recently due to its ubiquitous availability in indoor areas. This section provides a brief literature review of the existing works related to WiFi-based human activity recognition.

The RSSI can be used to measure the distance as well as the channel condition between the transmitter and receiver as an indication of the power level being received at the receiver. Most of the previous work proposed the use of RSS changes to recognise human activities by analysing a specific space. For example, the authors in [18] captured RSSI values from WiFi signals to recognize four activities: lying down, crawling, standing, and walking, and achieved recognition accuracy of over 80%. In [7], the authors proposed an approach based on the slow fading component of the received RSSI and the SVM algorithm that achieves an overall accuracy of 94%. However, because RSSI only provides coarse-grained information about channel variations, it is frequently influenced by multi-path effects and noise.

Unlike RSSI, CSI can capture the combined effects of scattering, fading, and even power delay as a function of distance. Fine-grained changes in wireless channels can be detected by CSI. It is a common communication link channel property. Because of the use of the release of the Linux 802.12n CSI tool [9], a significant amount of research has been conducted to use CSI measurements in HAR tasks.

In [22], Wang et al. gathered 1440 CSI samples for six daily activities such as walking, sitting and standing. They proposed a multi-task 1D convolutional neural network (CNN) with a ResNet-based basic architecture [10] and a simple aggregate loss function. This architecture achieved an average accuracy of 88.13% which is somewhat low accuracy with respect to the number of activities included in the dataset.

A framework called CSITime was proposed by Yadav et al. [23]. CSITime is a generic neural network architecture for CSI-based HAR. They treated the HAR problem as a multivariate time series classification problem, using time-series methods as inspiration. They evaluate the model on three different datasets: ARIL [8], StanWiFi [24], and SignFi [14]. CSITime achieved an accuracy of 98.20%, 98%, and 95.42%, on the ARIL, StanWiFi, and SignFi datasets, respectively. However, due to the lack of a standard dataset that uses the CSI data collected in such settings, they were unable to assess the performance of the proposed model in an environment with high interference.

Memmesheimer et al. [15] used Efficient-Net [20] to classify the multi-variate signal sequences after encoding them as images. They evaluated the model's performance on ARIL [8] and found accuracy to be 94.91%. The time-consuming nature of the computational process required to convert time-series data into image sequences is however one of the limitations of this work. Additionally, adopting a very complex model like Efficient-NET also increases the time needed for the computational process. In our own work, we directly extract helpful features from the time series data to avoid the requirement for this expressive encoding of time series sequences.

Damodaran et al. [5] proposed two different learning models for CSI-based HAR in indoor settings. One model uses sophisticated preprocessing and feature extraction methods based on wavelet analysis with a support vector machine (SVM) as a classifier, and the other model uses raw data directly with a long short-term memory (LSTM) network. In that work, the LSTM-based algorithms performed similarly to SVM-based algorithms despite requiring less preprocessing.

The work of Dempster et al. [6] addresses the trade-off between computational load versus accuracy of time series data-driven models. Their work transforms and classifies time series data using a random convolutional kernels method called ROCKET. They found that using a large number of random kernels was effective for capturing discriminative patterns in time series data, achieving state of the art classification results with a much reduced computational load. Their work is not focused in the domain of WiFi signals, but is of interest as a mechanism to address the complexity of time-series input.

Based on the previous achievements and limitations, we propose two deep learning models to automatically extract features from CSI measurements collected from multiple sub-carriers simultaneously in three different environments. The first model is mainly built based on attention-based bidirectional LSTM (BiLSTM). In the second model, we adopt the first model with a CNN layer before the BiLSTM layer to improve the recognition accuracy. The proposed models overcome some of the limitations of the mentioned related work such as the high cost of encoding time series sequences into image sequences and the difficulty of evaluating the proposed model in an environment with high interference.

3 Design and Methodology

In this section, we present the methodology of the proposed system for HAR including the data collection phase for model train and test, the methods adopted for data pre-processing and a description of the architecture of the deep neural network models. An overview of the proposed system is illustrated in Fig. 1. Each phase of the proposed model will be explained in detail as follows:

3.1 Data Collection

A publicly available dataset [2] is used to train our models. As the dataset is collected at the German Jordanian University, we named this dataset "GJWiFi".

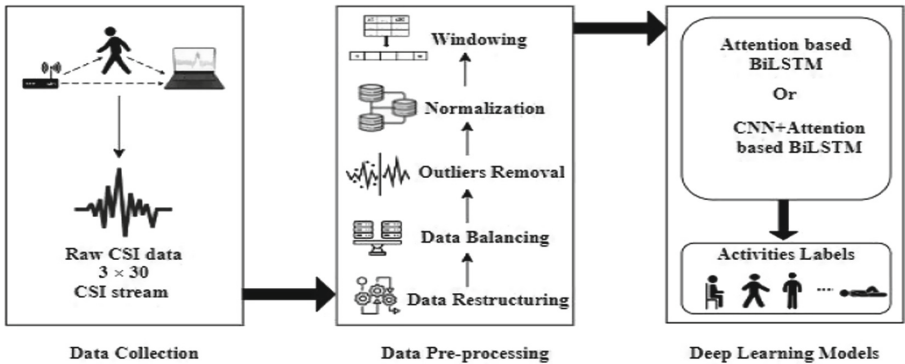


Fig. 1. The Overview of the proposed HAR system

The dataset was collected in three separate spatial environments: Laboratory is denoted E1, Hallway is denoted E2 and a Hybrid (Laboratory, Hallway and between them there is a barrier of 8 cm thickness) which is denoted E3. The settings of the three different environments are summarized in Table 1. Figure 2 shows a sketch of the three environments where Tx stands for transmitter and Rx stands for the receiver [2]. E1 and E2 are in LOS configurations while E3 is in NLOS configurations. For each environment, 10 subjects voluntarily participated in the data collection process. Each subject performed each activity a total of 20 times. The dataset contains 12 activities with different numbers of samples (i.e. the total number of CSI values for each activity) as illustrated in Table 2. Each environment contains 3000 data files. Each file contains measurements for 1 trial for 1 activity performed by 1 subject. The number of measurements in files is unequal. A Direct URL to the data: <https://data.mendeley.com/datasets/v38wjnz6f6/1> is provided by the original authors.

Table 1. Environments settings

Setting	Laboratory (E1)	Hallway (E2)	Hybrid (E3)
Propagation type	LOS	LOS	NLOS
Area	4.7 m x 4.7 m	7.95 m x 3.6 m	Not determined
Distance between Tx and Rx	3.7 m	7.6 m	5.44 m

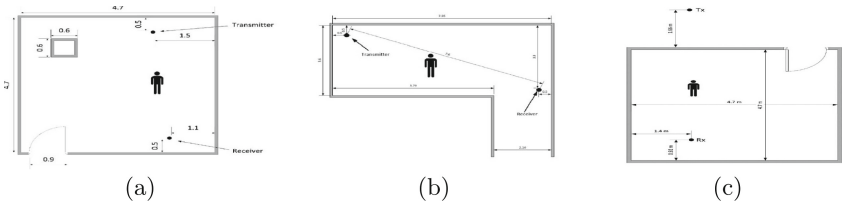


Fig. 2. (a) The Laboratory (E1) (b) The Hallway (E2) (c) Hybrid (E3)

Table 2. Description for activities included in the dataset

ID	Activity description	Number of samples	ID	Activity description	Number of samples
A1	Sitting on chair	501454	A7	Turning near the Rx	251545
A2	Falling down from sitting position	251199	A8	Walking from Rx to Tx	250136
A3	Lie down	501955	A9	Turning near the Tx	248912
A4	Standing	501084	A10	Standing up	200258
A5	Falling down from standing position	250798	A11	Sitting down	200414
A6	Walking from Tx to Rx	249984	A12	Pick a pen from the ground	274271

3.2 Data Preprocessing

The data preprocessing phase consists of 5 stages:

- **Data Restructuring:** The WiFi signals are stored in .csv files. We extract the CSI values only. The extracted CSI values are in complex form. So, we convert these complex CSI values to real values. Thus, amplitude measurements can be easily used in the training process.
- **Data Balancing:** The number of instances for each activity is unequal as shown in Table 2 because the timing of performing each activity is different [2]. Classification based on an unbalanced dataset performs poorly [13]. To balance our dataset, we used under-sampling [21], balancing class distribution by removing the majority of class samples randomly. This is repeated until the instances of the majority and minority classes are balanced to 200258 samples for each activity.
- **Outliers Removal:** Outlier values may result from noise in the environment, incorrect device reading, and an unexpected change in the environment. The Hampel filter [16] is used to eliminate these outliers by removing any reading that is greater than three times the standard deviation by replacing that value with the mean of the values within the sliding window. Since the transmitter is sending packets with a sampling rate of 320 packets/second. we apply the Hampel filter with a window size of 32 data points which is equivalent to 0.1 s.
- **Data Normalization:** Because time series data can have a wide range of values, it should be scaled to a similar range of values to speed up the learning process [17]. For this, we used the robust-scalar technique. It scales data by subtracting the median and scales the data according to the Interquartile Range (IQR). The IQR is the range between the 1st quartile and the 3rd quartile as shown in Eq. (1), where X is the value of the feature, $Q_1(X)$ is the 1st quartile, and $Q_3(X)$ is the third quartile [19].

$$X_{scaled} = \frac{X - median(X)}{Q_3(X) - Q_1(X)} \quad (1)$$

- **Data Windowing:** Classification algorithms cannot be applied directly to raw time-series data. So, the windowing approach [4] was applied to transform the raw time-series data from the shape of the 2-D vector (samples, features) into a 3-D vector (timestamp, samples, features). At each time instance, $1 \times 3 \times 30$ CSI streams are recorded during the measurements. A sliding window of 1.0 s is applied for all the instances to construct the CSI vector used in the training process. Then, new features were generated by aggregating the raw samples within each window. The activity with the highest number of occurrences (i.e. the most frequent) in that window was used to assign a class label to the transformed features. Moreover, we consider overlapping windows with 10% overlap instead of taking discrete windows. This ensures that each subsequent row in the transformed vector contains information from the previous window.

3.3 Deep Learning Models

After preprocessing, the collected activity data may still be very complex. The literature shows that it is difficult to effectively analyze CSI data using traditional methods such as SVM, decision trees, and fuzzy rule-based classifiers. The “GJWiFi” dataset is composed of time series measurements for human activities. To address the issue of extracting hidden patterns and dependencies from the complex data flow, we employ deep learning techniques that are effective in generating discriminative representations from complex data. We implemented two deep learning models for activity classification. The first model is built mainly using Attention-based Bidirectional LSTM (BiLSTM). The BiLSTM is a special LSTM that can extract both forward and backward long-term time dependencies on the time series sequences to make predictions more accurate. In the second model, a CNN is added to attention-based BiLSTM to take advantage of CNN in the detection of the most important local features without any human supervision and weight sharing that minimizes the cost of computing.

Attention-Based BiLSTM (ABiLSTM): The conventional LSTM can only process sequential CSI measurements in only forward direction. Thus, only past CSI information has been considered for the current hidden state. However, future information is extremely important when learning representative features for these similar activities. For example, laying down and sitting both require the human body to be lowered first, but the final positions for the two activities are different. As a result, we employ a BiLSTM network to learn effective features from raw CSI measurements. The BiLSTM is a bidirectional variant of LSTM that connects two hidden layers of opposite directions to the same output, so the network can remember information from forward and backward direction. This adds more context for recognition in memory as compared to LSTM. An attention layer is added to the BiLSTM model to give a different focus to the information extracted from the forward hidden layer and the backward hidden layer of BiLSTM. The learned features from BiLSTM are used as inputs for

attention layer to generate an attention matrix that indicates the importance of features and time steps. The attention layer is implemented with normalised weights for each feature as input and time step as outputs. Then, learned features are combined with the attention matrix via element-wise multiplication, producing the modified feature matrix. In this model, the input layer takes the preprocessed CSI sequence as the input vector. The BiLSTM layer is configured with 64 hidden units using tanh as the activation function and the return sequence variable is set to true to use the complete feature matrix as input to the next layer. The attention layer has 64 hidden units. It is implemented as a softmax regression layer. The attention layer is followed by the dropout layer to decrease the probability of overfitting. The output layer is a dense layer which consists of 12 neurons and uses softmax as its activation function to calculate the likelihood of 12 different activities.

CNN and Attention-Based BiLSTM (CNN-ABiLSTM): This second model is an adaptive version of the ABiLSTM model. A CNN layer and max-pooling layer are added to the architecture of the ABiLSTM model to improve the accuracy. The CNN layer is added after the input layer directly. The CNN layer has 32 filters with a kernel size of 3 and uses rectified linear unit (relu) as an activation function. The CNN layer is followed by a max-pooling layer with a pool size of 2 to reduce the dimensions of the feature maps. The pooling layer summarises the features in a region of the feature map produced by a convolution layer. This makes the model more resistant to changes in features. As a result, it reduces the number of parameters to learn as well as the amount of computation done in the network.

Training of Learning Models: To develop our proposed models, we used a **Google Colab** (a web IDE) that executed commands written in pro-version Python. We train the proposed models using the three environments E1, E2 and E3 separately. Because of limited RAM resources in Google Colab, we took 12800 samples for each activity per subject (i.e. the total number of measurements for each activity class is 128000). Thus, 1536000 samples in total for each environment. We apply stratified splitting to split the samples of each environment into 80% for the training dataset and 20% for the testing dataset. We took 20% of the training dataset for validation. In each environment, samples for each subject are represented in training, validation and testing data. Then we evaluate the performance of the proposed models for the three environments E1, E2 and E3 independently in terms of precision, recall and f1-score. The models were trained using the Adam optimizer with a batch size of 64 and a learning rate of 10^{-3} to minimise data sample loss. We also use early stopping during training to prevent overfitting. The validation loss was monitored and the training is stopped when the validation loss worsened from one epoch to the next using a patience value equal to 1. After evaluating the proposed models on E1, E2 and E3 separately, we want to discover whether the proposed models are applicable to other environments with different settings. So, we apply the transfer learning approach using the proposed models trained in one environment and tested in the other two environments.

4 Results and Discussion

We compare the performance of the proposed model based on three aspects: first, we evaluate the overall performance of the proposed models in terms of precision, recall, f1-score, and accuracy for each activity; second, we evaluate the proposed models that were trained in one environment and tested using the other two environments before and after applying transfer learning [3]; third, we compare the proposed deep learning models with hand-crafted state-of-the-art models.

4.1 Overall Performance

Because of the distinctive multi-path distribution, the spatial environment in which the experiments were carried out is an important factor for the WiFi-based human activity recognition system. Therefore, we evaluate the activity recognition performance of ABiLSTM and CNN-ABiLSTM models for the three environments E1, E2 and E3 independently in terms of precision, recall, and f1-score as shown in Table 3. From Table 3, it is shown that the performance improves for the three environments while using the CNN-ABiLSTM learning model than the ABiLSTM learning model indicating the good effect of adding the CNN. A decrease in the performance is observed in the E2 and E3 environments when compared to the experiments performed on in the E1 environment. This may be due to two factors: (i) the distance between the transmitter and the receiver in the E2 and E3 environment is longer than the distance between the transmitter and receiver in the E1 environment. and (ii) the experiments conducted in E2 and E3 involves the hallway which is considered a public area in the university that may be affected by surroundings and cause fading effects for the signals.

Table 3. Precision%, Recall%, and F1-Score% for E1, E2 and E3

Proposed models	Environments	Precision	Recall	F1-score
ABiLSTM	E1	94.54	94.05	94.03
	E2	92.11	91.96	91.93
	E3	93.31	92.59	92.58
CNN-ABiLSTM	E1	98.57	98.54	98.5
	E2	94.30	94.25	94.23
	E3	95.35	95.09	95.08

4.2 Impact of Environment

To verify the effectiveness of the proposed models, we propose to evaluate the performance of each model trained in one environment and tested using the other two environments before and after applying transfer learning. Before applying transfer learning means testing the proposed models trained in one environment and testing using the other two environments. While after applying transfer learning means that we freeze all the layers of the pre-trained model and add a

fully connected layer and an output layer. As shown in Table 4, the accuracy of testing the pre-trained models without transfer learning gives very low accuracy. This is to be expected as using WiFi signals in HAR for different spatial environments with different configurations affects the signals in various ways. Moreover, different human subjects with individual characteristics such as heights, weights, ages, genders, and body shapes will affect the signals in different ways, even if they are doing the same activity. When the transfer learning approach is applied, the recognition accuracy increased again after several epochs. Thus, the pre-trained models in one environment can adapt to the new environments after a few epochs of training. Therefore, the use of transfer learning to minimise retraining of CSI-based HAR models is promising.

Table 4. Recall % of pre-trained models on different environments

Training data	Testing data	Before transfer learning		After transfer learning	
		ABiLSTM	CNN-ABiLSTM	ABiLSTM	CNN-ABiLSTM
E1	E2	2	3	61.3	62.5
E1	E3	13.6	13	51.6	59.9
E2	E1	10	12.6	66.3	66.3
E2	E3	27	17	54.8	46
E3	E1	3.5	7.8	59.6	66.6
E3	E2	14.5	13.7	50.2	51.4

4.3 Comparison with State-of-Art Approaches

We compare the proposed models with state-of-the-art approaches using the “GJWiFi” dataset to illustrate that the proposed deep learning-based approaches earlier outperformed hand-crafted approaches. We compare the proposed models with the support vector machine (SVM) model applied to the same dataset. The authors in [1] apply the SVM model for 6 activities instead of 12 activities as they combine some activities and consider them as one class. Specifically, they combine stationary activities like lying down, sitting and standing into one class. Moreover, they conducted experiments in the laboratory and hall-way environments only. Table 5 summarizes a comparison between the proposed ABiLSTM, CNN-ABiLSTM and state-of-art SVM model. The results show that the recognition accuracy of the deep learning models is higher than the SVM model for the environments E1 and E2 except for ABiLSTM for E1 which has the same accuracy as SVM. However, the ABiLSTM model for E1 gives 94% accuracy for 12 activities instead of 6 activities. So it is better than the SVM model that gives the same accuracy for 6 activities only.

Table 5. Accuracy % of SVM, ABiLSTM and CNN-ABiLSTM

Environment	SVM	ABiLSTM	CNN-ABiLSTM
E1	94	94	98.54
E2	89.07	92	94

5 Conclusion and Future Work

This paper proposes a complete WiFi-based human activity recognition workflow from data collection to model evaluation. We developed two deep learning models: BiLSTM and CNN-BiLSTM. We evaluate the proposed model on different environment configurations separately using different performance metrics. We also evaluate the proposed models trained in one environment and tested using the other two environments before and after applying transfer learning. These experiments show a significant increase in the recall after applying transfer learning. Also, the recognition accuracy of the proposed deep learning models is significantly high compared to the state-of-art handcrafted models.

The potential areas for improvement in future work will include (i) extending the proposed model to recognize complex human activities, (ii) using transfer learning and fine-tuning approaches to improve the accuracy of the proposed models when tested in different environments, (iii) combine the three environments in one dataset and build a discriminator for the environment and propose a deep learning model to recognize the activity, human and in which environment the activity is performed, and (iv) examine approaches for treatment of time-series data input with a view to reducing computational load and/or increasing model accuracies.

Acknowledgement. This research was conducted with the financial support of Science Foundation Ireland under Grant Agreement No. 13/RC/2106_P2 at the ADAPT SFI Research Centre at Technological University Dublin ADAPT, the SFI Research Centre for AI-Driven Digital Content Technology, is funded by Science Foundation Ireland through the SFI Research Centres Programme.

References

1. Alsaify, B.A., Almazari, M., Alazrai, R., Alouneh, S., Daoud, M.I.: A CSI-based multi-environment human activity recognition framework. *Appl. Sci.* **12**(2), 930 (2022)
2. Alsaify, B.A., Almazari, M.M., Alazrai, R., Daoud, M.I.: A dataset for Wi-Fi-based human activity recognition in line-of-sight and non-line-of-sight indoor environments. *Data Brief* **33**, 106534 (2020)
3. Alshalali, T., Josyula, D.: Fine-tuning of pre-trained deep learning models with extreme learning machine. In: *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018*, pp. 469–473 (2018)

4. Banos, O., Galvez, J.M., Damas, M., Pomares, H., Rojas, I.: Window size impact in human activity recognition. *Sensors* **14**(4), 6474 (2014)
5. Damodaran, N., Haruni, E., Kokhkharova, M., Schäfer, J.: Device free human activity and fall recognition using WiFi channel state information (CSI). *CCF Trans. Pervasive Comput. Interact.* **2**(1), 1–17 (2020)
6. Dempster, A., Petitjean, F., Webb, G.I.: ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Min. Knowl. Discov.* **34**(5), 1454–1495 (2020)
7. Dib, W., Ghanem, K., Ababou, A., Nedil, M., Eskofier, B.: Receive signal strength- based human activity recognition. In: 2021 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting, APS/URSI 2021 - Proceedings, pp. 365–366 (2021)
8. Federico, C., et al.: A public domain dataset for human activity recognition in free-living conditions. In: Proceedings - IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, pp. 166–171 (2019)
9. Halperin, D., Hu, W., Sheth, A., Wetherall, D.: Tool release. *ACM SIGCOMM Comput. Commun. Rev.* **41**(1), 53 (2011)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016–December, pp. 770–778 (2015)
11. Khalili, A., Soliman, A.H., Asaduzzaman, M., Griffiths, A.: Wi-Fi sensing: applications and challenges. *J. Eng.* **2020**(3), 87–97 (2020)
12. Liu, J., Teng, G., Hong, F.: Human activity sensing with wireless signals: a survey. *Sensors* **20**(4), 1210 (2020)
13. Luque, A., Carrasco, A., Martín, A., de las Heras, A.: The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognit.* **91**, 216–231 (2019)
14. Ma, Y., Zhou, G., Wang, S., Zhao, H., Jung, W.: SignFi. *Proc. ACM Interact. Mob. Wearable Ubiquit. Technol.* **2**(1), 1–21 (2018)
15. Memmesheimer, R., Theisen, N., Paulus, D.: Gimme signals: discriminative signal encoding for multimodal activity recognition. In: IEEE International Conference on Intelligent Robots and Systems, pp. 10394–10401 (2020)
16. Pearson, R.K., Neuvo, Y., Astola, J., Gabbouj, M.: Generalized hampel filters. *EURASIP J. Adv. Signal Process.* **2016**(1), 1–18 (2016)
17. Raju, V.N., Lakshmi, K.P., Jain, V.M., Kalidindi, A., Padma, V.: Study the influence of normalization/transformation process on the accuracy of supervised classification. In: Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020, pp. 729–735 (2020)
18. Sigg, S., Shi, S., Buesching, F., Ji, Y., Wolf, L.: Leveraging RF-channel fluctuation for activity recognition: active and passive systems, continuous and RSSI-based signal features. In: ACM International Conference Proceeding Series, pp. 43–52 (2013)
19. Singh, D., Singh, B.: Investigating the impact of data normalization on classification performance. *Appl. Soft Comput.* **97**, 105524 (2020)
20. Tan, M., Le, Q.V.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (2019)
21. Tumrate, S., et al.: Classification of imbalanced data: review of methods and applications. *IOP Conf. Ser.: Mater. Sci. Eng.* **1099**(1), 012077 (2021)

22. Wang, F., Feng, J., Zhao, Y., Zhang, X., Zhang, S., Han, J.: Joint activity recognition and indoor localization with WiFi fingerprints. *IEEE Access* **7**, 80058–80068 (2019)
23. Yadav, S.K., et al.: CSITime: privacy-preserving human activity recognition using WiFi channel state information. *Neural Netw.: Official J. Int. Neural Netw. Soc.* **146**, 11–21 (2022)
24. Yousefi, S., Narui, H., Dayal, S., Ermon, S., Valaee, S.: A survey on behavior recognition using WiFi channel state information. *IEEE Commun. Mag.* **55**(10), 98–104 (2017)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





A Data-Driven Analysis of Formula 1 Car Races Outcome

Ankur Patil¹, Nishtha Jain², Rahul Agrahari², Murhaf Hossari²,
Fabrizio Orlandi², and Soumyabrata Dev^{2,3}(✉)

¹ National College of Ireland, Dublin, Ireland

² ADAPT SFI Research Centre, Dublin, Ireland
soumyabrata.dev@ucd.ie

³ School of Computer Science, University College Dublin, Dublin, Ireland

Abstract. There are a range of factors that affect the outcome of Formula 1 (F1) car races. Today, it is reasonable to say that F1 races are first won at the factory, and then on the track. F1 teams accumulate enormous amounts of data during races. In this paper, we propose a data-driven approach to identify the most important factors that contribute to the overall points scored by each driver in a F1 season. We perform a correlation analysis along with a principal components analysis (PCA) to identify the factors that are closely related. Furthermore, using PCA, we efficiently reduce our 21 input variables into a lower-dimensional subspace, that can explain most of the variance in our data and which is easier to comprehend. We obtain 5 years (2015–2019) of data explaining the F1 car characteristics from a publicly available website <https://www.racefans.net/>. We use this web-scraped F1 race study to understand the impact of the different car features on the total points scored by a driver in the season. To the best of our knowledge, our work is the first of its kind in the area of F1 car races.

Keywords: Formula-1 · Feature analysis · Data analytics · Open-source code

1 Introduction

One of the most popular sports in the world is Formula 1 (F1). The speed thrill and nail-biting experience that fans get while watching the race is the result of a lot of engineering, data science, management, and of course lots of training on the tracks. What is often underappreciated is that races are won first at the factory and then on the circuit. The F1 teams work hard to maintain a constant balance between obtaining top speed and down force, here aerodynamics plays a major role [12]. The teams try to predict what position they will finish by using the massive datasets they have accumulated from the past seasons. It would be worthwhile to dig deeper into such a sport and analyze the associated analytics

A. Patil and N. Jain—Authors contributed equally.

© The Author(s) 2023

L. Longo and R. O'Reilly (Eds.): AICS 2022, CCIS 1662, pp. 134–146, 2023.

https://doi.org/10.1007/978-3-031-26438-2_11

to understand its impact on the total car race points accumulated by a F1 driver. In this work, we provide a data-driven framework to understand the various car race statistics, and check their impact on the performance of the F1 racers.

There are 4 basic strategies that the driver/team uses during the race. This will assist us in understanding the basic fundamentals of F1 race, before diving deep into the winning prediction methodologies.

- **Preparation:** The engineering team works on developing a strategy which is based on simulations and data that have been acquired from the various trial runs the driver takes and also based on the past races.
- **Practice:** The driver practices and uses the strategy provided. This is a great way to correct the shortcomings in the strategy. This acts as a stepping stone in fine-tuning the strategy for the qualifying and final race.
- **Qualifying:** The driver moves on to taking the qualifying rounds and the starting position. The previous practice and qualifying rounds feed very critical information to the engineering team to work on the final race's pit stops and race strategy.
- **Racing:** Technical difficulties are a part and parcel of F1 races, but there are a few other things that act as catalyst to the victory or loss of the team/driver. Weather conditions, traffic on the tracks, pit stops for tyre or oil changes or other quick fixes, and of course the safety car which limits the speeding cars from crashing when obstructions appear on the track.

Now that we are aware of the 4 important stages in devising a successful strategy for winning the race, it is time to dive deeper into our objective of improving the decision-making steps using statistical tools and techniques.

1.1 Related Work

In the literature, a lot of work is done by researchers in predicting the different sports results using machine learning techniques. In the work by Bishell [4], the author performed experiments on horse races data, and implemented a neural network for predicting the horse race outcomes. Bishell concludes that a simple neural network model gave more efficient and accurate results compared to other benchmarking models. The neural model managed to achieve an accuracy of 66% for the top three ranks. Another research conducted by William and Li [19] using the data collected from Caymans Race Track in Jamaica. The authors implemented the model by using a neural network and achieved an overall accuracy of 74% to predict the top three positions. Similar research was done in 2010 by Dacoodi and Khanteymoori [7], they acquired the data from Aqueduct Race Track in NY. Their work proposed a neural net that has an accuracy of 77%, when compared with other neural networks. In [13], Miljković *et al.* did research on predicting the outcomes of basketball matches using Naive Bayes, using the data acquired from the NBA website. The model achieved an accuracy of 77.97%. Also, [9] predicted the outcomes of the matches played by Tottenham Hotspurs in English Premier League. In [14], the author proposes a model developed from

Bayes networks to predict the expert knowledge in the game of football. The author concludes that Bayes network achieved an accuracy of 59.21% and outperforms other benchmarking algorithms. Recently, in 2017, [18] conducted a research analysis on predicting the football matches played in English Premier League using Bayes Networks and the prediction accuracy was 75.09% on an average across three seasons.

The majority of this prior research focused on developing predictive models with high generalization accuracy (as measured by performance on test sets) rather than on analyzing the factors that contribute to the outcome of a sports event. Furthermore, F1 races and its related analysis have been largely ignored. To the best of our knowledge, there is no publicly published work that provides a systematic analysis on race features that influences the outcome in F1 races. In this paper, we attempt to bridge this gap and provide a detailed analysis on F1 car analytics.

1.2 Contributions of This Paper

The main contributions of this paper are as follows:

- Firstly, we propose a novel and systematic analysis of the various F1 car race factors in our collected dataset, that govern the finishing position of a driver and the manner in which they are related to each other;
- Secondly, we successfully reduced the data space comprising 21 race features into 4 orthogonal dimensions that explain approximately 70% of the captured variance, using principal components analysis. This will facilitate us in identifying the key factors of F1 car race influencing the race outcome;
- Finally, in the spirit of reproducible research, we release all the code and associated dataset with this work. The data set for this domain of sports analytics is a bit difficult to obtain in the form of direct CSV files. We have web scraped the data using R language for this work. We subsequently converted this data set from multiple pages on the website into a re-usable CSV file.

The rest of the paper is organized as follows. Section 2 discusses the various factors associated with a F1 car race. Section 3 describes their inter-dependency in details. We perform a dimensional reduction of the original feature space, using PCA in Sect. 4. Subsequently, we analyze the impact of the different car race features in total race points in Sect. 5. Finally, Sect. 6 concludes the paper and discusses the future works.

2 Formula-1 (F1) Car Race Factors

In this section a brief discussion is done on data collection, data pre-processing and transformation of the input data.

2.1 Dataset

The dataset used in this paper has been acquired from a single source. This dataset is obtained after web scrapping using R studio. With the spirit of reproducible research, the dataset and code for this work is reproducible and is available online¹. The dataset was taken from <https://www.racefans.net/2018-f1-season/2018-f1-statistics/>. We collected the data for a period of 5 years (2015–2019).

The dataset provides information on the following attributes:

- Average number of pit stops taken by each racer across the board is represented by **Average.Pit.Stop**
- Information about % usage of each tyre type is represented by variables **Hard**, **Medium**, **Soft**, **Super.soft**, **Ultra.soft**, **Hyper.soft**, **Wet** and **Intermediate** - which denote their % use
- Laps each driver spent in each position during the season considering only first, second and third position is represented by the variables **FirstPosition**, **SecondPosition** and **ThirdPosition**
- # of races the driver started is represented by the variable **Started**, # of races the driver classified by completing 90% of the race is represented by the variable **Classified** and # of races the driver completed by covering 100% race distance in the season is represented by the variable **Completed**
- Full season laps led (represented by **Full.seasons.laps.led**) and driver's season laps led (represented by **Driver.s.season.laps.led**) explain the number of laps led as percentage during the season and all race laps covered by that driver respectively
- # of accidents by each racer in the season is represented by the variable **Accident**
- # of penalties attained to the team and driver for each driver are represented by **Penalties.due.to.team** and **Penalties.due.to.driver** respectively. Simultaneously, if there was no penalty given, it counts as a no action and that is represented by the variable **No.action**
- Average position where each driver started every race, after penalties were applied is represented by the **Average.pole.position**
- The total number of points scored by the driver during the season is denoted by **Total.Points**. These points eventually decide the winner of each season

2.2 Data Pre-processing

This section gives us insights on how missing values, data transformation and data pruning were dealt with in order to carry the analysis forward.

Data Pruning: Data pruning refers to getting rid of unwanted data which are not required for analysis. In our case we performed data pruning on the attributes which were outliers and had no significance on the analysis. The

¹ https://github.com/nshthj/F1_race_exploratory_analysis.

attribute **Withdrawn** (W) described all the drivers who had withdrawn from the race. Here all the racers did participate and there was no driver who withdrew. So, this attribute was removed. Also the attribute **Did Not Qualify** (DNQ) consists all the data for racers who did not qualify. However, all the drivers did qualify for the final race and hence this attribute was removed.

Handling the Missing Values: After data pruning missing values were detected and analyzed as to why they are absent. The missing values in the data is not because of faulty data entry or avoided data. It is because the driver has not been involved in that event. As an illustration, in the event of an accident, only a couple of drivers were affected. Hence, the missing values were replaced with zero.

2.3 Data Transformation

The variables that underwent transformation are as follows:

- Pit Stop data was mentioned according to each lap i.e. 22 laps. A mathematical average was calculated and average pit stop for each driver was created.
- Tyres data was in the form of a percentage. All the special characters were taken off and the percentage was normalized to a decimal format.
- Full season laps led and Driver's season laps led was in the form of percentage which was normalized to a decimal format.

The variables what we have in the dataset are all considered to be important. However, there are 22 variables so having a feature selection process in place to get more independent and uncorrelated input variable set becomes all the more important. Most classification algorithms thrive on input variables that are independent of each other in order to explain maximum variation and trends in the dataset. This paper essentially explores these different variable selection processes. We first talk about a rather straightforward correlation analysis and then move on to a more comprehensive principal components analysis.

3 Interdependency of Variables

In this section, we do a correlation analysis [5,6] of all the variables described in the aforementioned sections. We have used the R function `corrgram`². In our case, as mentioned all the attributes are considered important for the research and there was no manual removal of features. It is important to understand the correlation trend [1,17] between the different features before we perform any classification task. This is because if two features are perfectly correlated, then one feature can be efficiently described by the other [11,16]. Figure 1 depicts how attributes are correlated with each other.

² <https://www.rdocumentation.org/packages/corrgram/versions/1.13/topics/corrgram>.

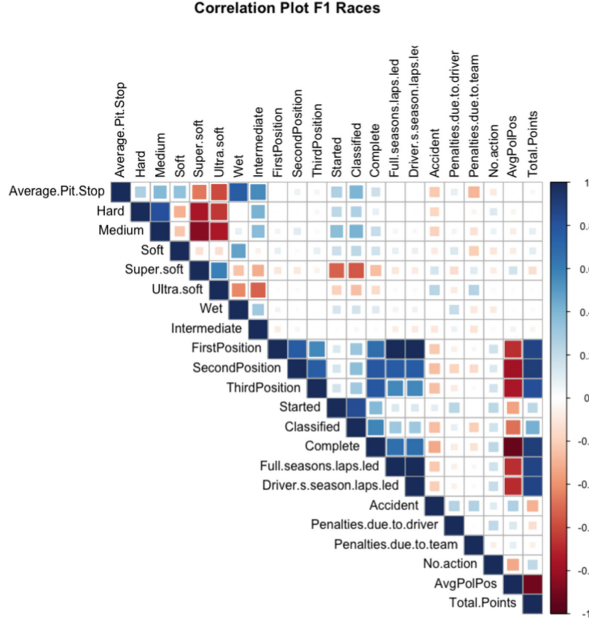


Fig. 1. Correlation between the various F1 car race variables (best viewed in color).

We observe that the average pole position is strongly negatively correlated with the first, second and third position. This makes sense as a higher average pole position would perhaps mean the racer didn't finish in the first, second or third position at the end of the race - also depicting that the average pole position is perhaps one of the key factors in determining the finishing position of the driver. Interestingly, we observe that team penalties appear to be related to the usage of soft tyres and hyper soft tyres – using soft tyres more often generate less penalties while usage of hyper soft tyres will generate more penalties. Moreover, hyper soft tyres are positively correlated with the occurrence of accidents - in line with the fact that they can cause more penalties. Additionally, the position features viz. first, second and third position are strongly positively dependent on the number of laps completed, the full seasons laps led by the drivers and whether the driver was classified or not. Another interesting relationship is the strong negative correlation between a driver classifying and the occurrence of accidents.

4 Principal Components Analysis

In addition to the inter-dependency of the different variables, we also use Principal Component Analysis (PCA) [3, 15] to understand the underlying structure of the dataset. Let us assume that our F1 race features are the column vectors \mathbf{v}_{1-22} (22 in our case), where $\mathbf{v}_j \in \mathbb{R}^{n \times 1}$ where $j = 1, 2, \dots, 22$, and n is the

total number of observations in the dataset. We stack the individual feature vectors \mathbf{v}_j to create the variable matrix $\mathbf{X} \in \mathbb{R}^{n \times 22}$:

$$\mathbf{X} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{22}]. \quad (1)$$

We normalize each of the feature vectors \mathbf{v}_j with the corresponding mean value \bar{v}_j and the standard deviation σ_{v_j} to compute the normalised matrix $\ddot{\mathbf{X}}$. We compute the matrix $\ddot{\mathbf{X}}$ as:

$$\ddot{\mathbf{X}} = \left[\frac{\mathbf{v}_1 - \bar{v}_1}{\sigma_{v_1}}, \frac{\mathbf{v}_2 - \bar{v}_2}{\sigma_{v_2}}, \dots, \frac{\mathbf{v}_j - \bar{v}_j}{\sigma_{v_j}}, \dots, \frac{\mathbf{v}_{22} - \bar{v}_{22}}{\sigma_{v_{22}}} \right]. \quad (2)$$

We thereby compute the covariance matrix of $\ddot{\mathbf{X}}$. Subsequently, we perform eigenvalue decomposition of the computed covariance matrix to obtain the eigen values and the eigen vectors. The eigen values describe the amount of variance captured by each of the principal components. The principal components are obtained from the eigen vectors.

4.1 Variation Explained by the Components

In this section, we analyze the variance captured by the most important principal components. Figure 2 describes the variance captured by each of the *orthogonal* principal components. We observe that the first two principal components capture 50% of the total variance. Furthermore, the cumulative variance captured by the first 4 principal components is $\approx 70\%$. This indicates that most of the race features are correlated with each other (as observed in Sect. 3), and the total information in the original feature space can be effectively reduced to a lower dimensional subspace without the loss of significant information.

4.2 Bi-plot Representation

We also represent the car race variables in the new subspace representation of the principal components. Figure 3 is the bi-plot representation [2, 8] of our race variables across the first two principal components in a two-dimensional space. We represent the different race observations in our dataset by points in the bi-plot figure. We represent the race car variables by vectors. The bi-plot figure provides us interesting insights on the F1 car race variables. We can observe the contribution of each of the race variables onto the principal components, and also the correlation between them. The position variables viz. **FirstPosition**, **SecondPosition**, **ThirdPosition** are correlated with each other and have a strong contribution to the second principal component. In addition to that, other variables related to the driver's position in the race are quite strongly contributing to PC1 - thus making it a PC that potentially explains the positional aspect of the driver. We also observe that accident and penalties due to team are correlated with each other. We don't see a similar dependence of variables on any other PCs, hence the other three components explain the variation in the input variables in a cumulative manner.

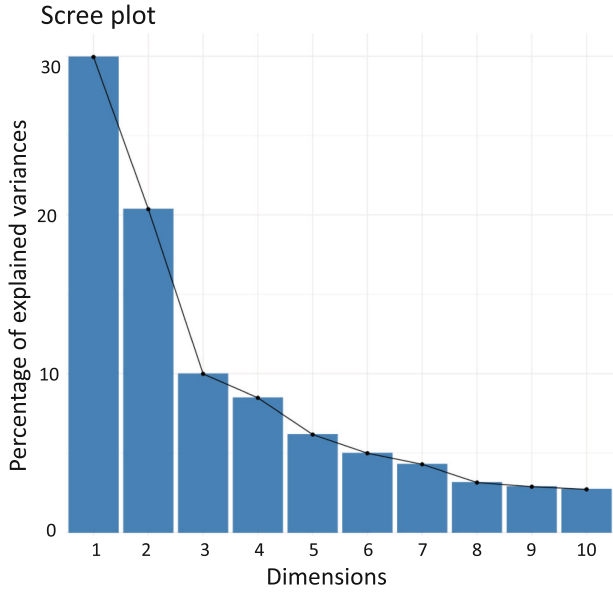


Fig. 2. Amount of variance captured by the individual principal components.

4.3 PCA Factor Loadings

The PCA factor loadings explain the loading that each variable has on each of the components. It also shows the range of loadings on each principal component from each variable [10]. Table 1 describes the loading factors of the various car race features onto the first four principal components. The bold loadings show the top 6 loading *magnitude-wise* on each principal component. It helps us understand what could each principal component potentially represent. For example, similar to the findings in the previous section, the first PC shows strong loadings for all position-related variables. Similarly, the third PC has maximum loadings on the tyre related variables, thus accounting for the variance based on the type of tyre used during the race. It is also possible for one variable to have high loadings on multiple principal components, as can be seen in the table as well.

5 Impact on Season’s Total Championship Points

We have discussed the relationship between the different factors that determine the final race outcomes. In this section, we run a linear regression on the data obtained from web-scraping. This data consists of information from 5 consecutive seasons of 2015 till 2019. The dependent variable in the linear regression is the total points scored by a driver in each season denoted by **Total.Points**. This is chosen as the dependent variable, because eventually the driver with

Table 1. Loading factors of the various features onto the first four principal components.

Race features	PC ₁	PC ₂	PC ₃	PC ₄
Average.Pit.Stop	-0.118	0.330	-0.346	-0.128
Hard	-0.102	0.333	0.387	-0.085
Medium	-0.147	0.363	0.313	-0.038
Soft	-0.066	0.039	-0.550	0.094
Super.soft	0.177	-0.360	-0.137	-0.110
Ultra.soft	0.114	-0.383	0.004	0.189
Wet	-0.072	0.208	-0.487	-0.040
Intermediate	-0.041	0.283	-0.021	-0.245
FirstPosition	-0.328	-0.181	0.003	-0.076
SecondPosition	-0.331	-0.155	0.026	-0.123
ThirdPosition	-0.289	-0.145	-0.010	-0.048
Started	-0.179	0.171	-0.009	0.520
Classified	-0.265	0.157	-0.040	0.305
Complete	-0.360	-0.069	-0.022	0.036
Full.seasons.laps.led	-0.327	-0.182	-0.001	-0.076
Driver.s.season.laps.led	-0.326	-0.187	-0.001	-0.065
Accident	0.149	-0.045	0.009	0.388
Penalties.due.to.driver	0.018	0.100	-0.125	0.413
Penalties.due.to.team	0.074	-0.100	0.172	0.162
No.action	-0.100	-0.023	0.153	0.327
Average.Pol.Pos	0.339	0.140	-0.041	-0.063

in **Total.Points**. The linear regression model has an R-squared value of 99% which means that the model was able to capture almost 99% of the variation in the data.

6 Conclusion and Future Work

In this paper, we have provided a systematic analysis of various variables associated with the F1 car race. We have identified the most important variables that assist in a favorable outcome of the car race. Using a set of statistical techniques, we concluded that most of the variables are strongly correlated with each other. We also surmised that the original feature space can be significantly reduced to a lower-dimensional subspace without a significant loss of information.

Future work include extending such systematic analysis for a larger statistical period of more than 5 years to gather more data and investigate the analysis further. Furthermore, we plan to investigate the linear regression model by

Table 2. We show the corresponding estimate and p-value for all the car race features, while estimating the total race points accumulated by a driver in a complete season. The significance codes are represented by ‘+’, where 0: ‘+++’, 0.001: ‘++’, 0.01: ‘+’, 0.05: ‘.’, and 0.1: ‘.’.

Race features	Estimate	p-value	Signif. codes
Average.Pit.Stop	−11.29	0.37	
Hard	1.09	0.35	
Medium	2.11	0.10	
Soft	2.13	0.08	
Super.soft	1.69	0.17	
Ultra.soft	2.11	0.09	
Wet	1.28	0.43	
Intermediate	3.62	0.05	+
FirstPosition	0.21	0.73	
SecondPosition	0.20	7.34e−07	+++
ThirdPosition	0.19	6.53e−07	+++
Started	−0.68	0.62	
Classified	0.03	0.99	
Complete	5.56	7.96e−05	+++
Full.seasons.laps.led	−2.15	0.82	
Driver.s.season.laps.led	3.39	0.30	
Accident	0.26	0.91	
Penalties.due.to.driver	−1.24	0.29	
Penalties.due.to.team	1.45	0.13	
No.action	0.44	0.77	
Average.Pol.Pos	−3.44	0.02	+

modifying it to use a selected set of race features by applying forward and/or backward step regression.

Acknowledgement. This research was conducted with the financial support of Science Foundation Ireland under Grant Agreement No. 13/RC/2106.P2 at the ADAPT SFI Research Centre at University College Dublin. ADAPT, the SFI Research Centre for AI-Driven Digital Content Technology, is funded by Science Foundation Ireland through the SFI Research Centres Programme. The authors would also like to thank Prof John D. Kelleher from Technological University Dublin, Ireland for helpful discussions on this work.

References

1. Alparslan, B., Jain, M., Wu, J., Dev, S.: Analyzing air pollutant concentrations in New Delhi, India. In: 2021 Photonics & Electromagnetics Research Symposium (PIERS), pp. 1191–1197. IEEE (2021)
2. AlSkaif, T., Dev, S., Visser, L., Hossari, M., van Sark, W.: A systematic analysis of meteorological variables for PV output power estimation. *Renew. Energy* **153**, 12–22 (2020)
3. Batra, S., et al.: DMCNet: diversified model combination network for understanding engagement from video screengrabs. *Syst. Soft Comput.* **4**, 200039 (2022)
4. Bishell, A.: Machine learning and New Zealand horse racing prediction. BSc. Report, Department of Computer Science, Massey University, New Zealand (2006)
5. Danesi, N., Jain, M., Lee, Y.H., Dev, S.: Monitoring atmospheric pollutants from ground-based observations. In: 2021 IEEE USNC-URSI Radio Science Meeting (Joint with AP-S Symposium), pp. 98–99. IEEE (2021)
6. Danesi, N., Jain, M., Lee, Y.H., Dev, S.: Predicting ground-based PM2.5 concentration in Queensland, Australia. In: 2021 Photonics & Electromagnetics Research Symposium (PIERS), pp. 1183–1190. IEEE (2021)
7. Davoodi, E., Khanteymoori, A.R.: Horse racing prediction using artificial neural networks. *Recent Adv. Neural Netw. Fuzzy Syst. Evol. Comput.* **2010**, 155–160 (2010)
8. Dev, S., Lee, Y.H., Winkler, S.: Color-based segmentation of sky/cloud images from ground-based cameras. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **10**(1), 231–242 (2017)
9. Joseph, A., Fenton, N.E., Neil, M.: Predicting football results using Bayesian nets and other machine learning techniques. *Knowl.-Based Syst.* **19**(7), 544–553 (2006)
10. Manandhar, S., Dev, S., Lee, Y.H., Winkler, S., Meng, Y.S.: Systematic study of weather variables for rainfall detection. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, pp. 3027–3030. IEEE (2018)
11. Manandhar, S., Dev, S., Lee, Y.H., Meng, Y.S., Winkler, S.: A data-driven approach for accurate rainfall prediction. *IEEE Trans. Geosci. Remote Sens.* **57**(11), 9323–9331 (2019)
12. Martins, D., Correia, J., Silva, A.: The influence of front wing pressure distribution on wheel wake aerodynamics of a F1 car. *Energies* **14**(15), 4421 (2021)
13. Miljković, D., Gajić, L., Kovačević, A., Konjović, Z.: The use of data mining for basketball matches outcomes prediction. In: Proceedings of IEEE 8th International Symposium on Intelligent Systems and Informatics, pp. 309–312. IEEE (2010)
14. Pariath, R., Shah, S., Surve, A., Mittal, J.: Player performance prediction in football game. In: Proceedings of Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 1148–1153. IEEE (2018)
15. Pathan, M.S., Nag, A., Dev, S.: Efficient rainfall prediction using a dimensionality reduction method. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, pp. 6737–6740. IEEE (2022)
16. Pathan, M.S., Nag, A., Pathan, M.M., Dev, S.: Analyzing the impact of feature selection on the accuracy of heart disease prediction. *Healthc. Anal.* **2**, 100060 (2022)
17. Pathan, M.S., Wu, J., Lee, Y.H., Yan, J., Dev, S.: Analyzing the impact of meteorological parameters on rainfall prediction. In: Proceedings of IEEE USNC-URSI Radio Science Meeting (Joint with AP-S Symposium), pp. 100–101. IEEE (2021)

18. Razali, N., Mustapha, A., Yatim, F.A., Ab Aziz, R.: Predicting football matches results using Bayesian networks for English Premier League (EPL). In: Proceedings of IOP Conference Series: Materials Science and Engineering, vol. 226, p. 012099. IOP Publishing (2017)
19. Williams, J., Li, Y.: A case study using neural networks algorithms: horse racing predictions in Jamaica. In: Proceedings of International Conference on Artificial Intelligence (ICAI 2008), pp. 16–22. CSREA Press (2008)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Brain Tumor Synthetic Data Generation with Adaptive StyleGANs

Usama Tariq¹, Rizwan Qureshi^{1,2,3}, Anas Zafar¹, Danyal Aftab¹, Jia Wu²,
Tanvir Alam³, Zubair Shah³, and Hazrat Ali³(✉)

¹ National University of Computer and Emerging Sciences, Karachi, Pakistan
K173810@nu.edu.pk

² Department of Imaging Physics, MD ANDERSON Cancer Center, The University
of Texas, Houston, TX, USA
{FRizwan, JWu11}@mdanderson.org

³ College of Science and Engineering, Hamad Bin Khalifa University, Qatar
Foundation, Doha, Qatar
{talam, zshah, haali2}@hbku.edu.qa

Abstract. Generative models have been very successful over the years and have received significant attention for synthetic data generation. As deep learning models are getting more and more complex, they require large amounts of data to perform accurately. In medical image analysis, such generative models play a crucial role as the available data is limited due to challenges related to data privacy, lack of data diversity, or uneven data distributions. In this paper, we present a method to generate brain tumor MRI images using generative adversarial networks. We have utilized StyleGAN2 with ADA methodology to generate high-quality brain MRI with tumors while using a significantly smaller amount of training data when compared to the existing approaches. We use three pre-trained models for transfer learning. Results demonstrate that the proposed method can learn the distributions of brain tumors. Furthermore, the model can generate high-quality synthetic brain MRI with a tumor that can limit the small sample size issues. The approach can address the limited data availability by generating realistic-looking brain MRI with tumors. The code is available at: <https://github.com/rizwanqureshi123/Brain-Tumor-Synthetic-Data>.

Keywords: Brain tumor · Deep learning · Generative models · Computer vision · MRI

1 Introduction

Due to the advancements in computational power and a large amount of high-quality datasets, deep learning has become the state-of-the-art technology in computer vision, natural language processing, and others [1]. Deep learning has also made remarkable progress in all areas of medical image analysis, including segmentation, detection, and classification [1]. However, deep learning models are

trained on large datasets, which may not be available in the medical domain due to privacy and ethical concerns [2]. Medical experts find it difficult to publicize the majority of medical images without patients' consent. In addition, the public datasets are also small and lack expert annotations, thus, hindering their use for training deep neural networks. Furthermore, most of the available datasets might contain unbalanced classes that may hinder the performance of deep learning models and may not produce critical biological insights.

To overcome the problem of data unavailability, many researchers use generative models [3] to generate realistic synthetic images with diverse distributions for training complex deep learning models for medical analysis. Generative Adversarial Networks (GAN), a type of neural network, comprises two neural networks, one of which focuses on image production and the other on discrimination. The training of GAN involves a contest between the generator G and the discriminator D . The discriminator D is a binary classifier that determines if the data generated by G belongs to the training set or not (real versus unreal). GANs can be used to create synthetic medical images, image captioning, and cross-modality image generation [4, 5]. Due to the adversarial training scheme's success in creating new image samples and utility in preventing domain shift, GANs have drawn great interest from the research community. However, a GAN with insufficient training data leads to over-fitting the discriminator. The feedback to the generator becomes meaningless, and the training starts to diverge [6]. A common approach to overcome over-fitting is data augmentation. For instance, training an image classifier by including images with rotation, noise, or scaling may increase the classifier's invariance to certain semantics-preserving distortions, which is a very desired quality. On the other hand, a GAN trained with comparable dataset augmentations learns to produce the augmented distribution [7].

Medical image analysis tasks such as brain tumor diagnosis [8] are critical where one would wish for minimum error from a computer model. Brain tumor refers to excessive growth of cells in regions of the brain. An early diagnosis of a brain tumor increases the effectiveness of the treatment and hence, the survival rates. Early diagnosis of a brain tumor is necessary in order to treat it properly; otherwise, it might cause severe damage to the brain that can eventually be fatal. Magnetic Resonance Imaging (MRI) is the most popular way to generate brain scans and detect tumors in different regions of the brain. Many deep learning models [9] have been introduced recently to detect tumors in brain MRIs. However, progress is generally hindered by the lack of enough data. Traditional data augmentation methods, such as rotation, translation, mirror, and lightning, are not sufficient to generate a diverse, realistic dataset for brain tumor diagnosis. Synthetic images can be generated for this purpose which can address the problems associated with data acquisition, such as; privacy concerns, class imbalance, and small sample size.

Generative Adversarial Networks (GANs) have been very popular for generating realistic diversified datasets. In 2018 StyleGAN [10] was proposed, with the main aim to improve the existing generator architecture G . StyleGAN mainly

improved the existing architecture of the generator network in ProGAN [11] for better performance and kept the discriminator D network and loss functions constant. The latent code (z) is transformed into an intermediate latent code (W) prior to feeding it into the network. The synthesis network (G) is supervised by affine transforms through an adaptive instance that adds random noise maps to the space W resulting in much entangled latent space. The proposed model is capable of generating realistic, high-quality images and offers control over the new style of the generated image.

StyleGAN2 [12] architecture was presented to overcome issues present in the initial images generated by StyleGAN, such as blob and phase artifacts. Two causes were identified for the artifacts introduced in StyleGAN such as; fixed positions of eyes and nose and water droplet effects. Upon investigating the cause of common bob-like artifacts, it was observed that it was generated by the generator in response to an architectural design defect. A new design was proposed for the normalization used in the generator, which removed artifacts.

In this paper, we used StyleGAN2 with adaptive discriminator augmentation (ADA) [6] for generating brain tumor MRI images of 512×512 resolution while utilizing a significantly limited amount of training data when compared to the existing approaches. Our proposed approach effectively addresses the problem of data limitation by generating realistic brain MRI with tumor samples and can learn different data distributions from brain tumor raw images. The experiments are conducted on the brain tumor dataset. We utilized pre-trained models trained on FFHQ dataset [10], BreCaHaD dataset [13], and AFHQ dataset [14]. The experimental results indicate that these models can generate superior quality superior MRI tumor samples that can be effectively utilized for medical analysis. The remaining paper is organized as: Sect. 2 provides a review of the related literature. Section 3 explains the methodology and the architecture. Section 4 presents experiments, and Sect. 4 presents the results and discussion for the synthesis of brain MRI having tumors. Finally, Sect. 6 concludes the paper.

2 Literature Review

The StyleGAN architecture generates style images while controlling different high-level attributes of the images [10]. The generator architecture in this research was designed in such a way that helps to control the image synthesizing process by learning on a constant input of $4 \times 4 \times 512$ and on each subsequent layer based on latent code for adjusting the style of the image. When noise is provided as an input to the network, this combined effect helps segregate high-level attributes from stochastic variation in generated images and allows for better style mixing and interpolation. The datasets used in the work are FFHQ [15], LSUN [16] and CelebA-HQ [17]. The concept of intermediate latent space was used, which significantly affected how variational factors are represented in the network and could be disentangled. Two metrics, i.e., the perceptual path length and linear separability, were used to estimate the degree of latent space disentanglement.

StyleGAN2 was introduced in [12] to address the characteristic artifacts and improve the output of StyleGAN. The first reason for the artifacts was the attempt of StyleGAN to evade a design flaw related to instance normalization used in AdaIN. The second type of artifact was related to progressive growth and was addressed in StyleGAN2 by changing the training method. A method for mapping low-resolution medical images to high-resolution medical images using generative models is presented in [18]. In [19], considering the limitation of GANs to generate high-quality images for domains that have very little data, one of the very recent breakthroughs in generative modeling is a text-driven method that allows domain adaptation capability to the generator model for generating images across a multitude of domains. A text-driven method for out-of-domain image synthesis is proposed. The domain shift was carried out by adjusting the generator's weights in the direction of images aligned with the driving text. The network architecture is dependent on StyleGAN2 and Contrastive-Language-Image-Pre-training (CLIP) [20,21].

CLIP model has been used for discovering semantically rich and meaningful latent manipulations in order to generate images with styles defined through text based interface. In the first stage, an optimization task has been applied using CLIP-based loss to manipulate a latent input vector. In the next stage, a latent mapper for an optimized text-based manipulation given an input image has been used. Effectively, mapping the text-based inputs in the direction of StyleGAN style space results in effective text-based image manipulation. Motivated by the potential of the StyleGAN2 architecture to generate improved images of human faces, we use the StyleGAN2-ADA architecture to synthesize brain MRI, as explained in the following sections.

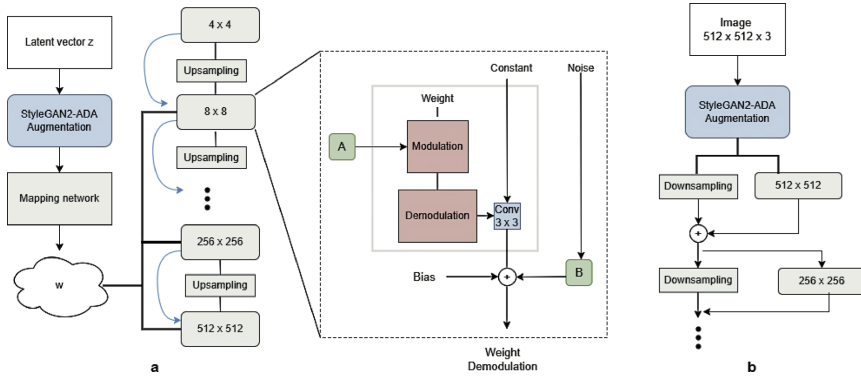


Fig. 1. StyleGAN2-ADA (a) **Generator.** Based on the incoming style, the modulation scales each input feature map of the convolution, and the demodulation module is used to remove the droplet artifacts. (b) **Discriminator.** After the input vectors of the components, StyleGAN2-ADA performs data augmentation.

3 Methodology

We have utilized StyleGAN2 with ADA methodology to generate high-quality MRI brain tumor images while using a significantly limited amount of training data. The proposed pipeline of StyleGANs with ADA is shown in Fig. 1. StyleGAN2 [12] introduced several changes in the architecture to overcome the issues in StyleGAN. Many viewers observed distinctive artifacts in StyleGAN images. Two key issues were identified in the output of StyleGAN, and changes were introduced in the architecture and the training method accordingly. Upon investigating the cause of common blob-like artifacts, it was observed that the blobs were generated in response to an architectural design defect. A new design was proposed for the normalization used in the generator, which helped in removing the artifacts. It was concluded that the artifacts related to progressive growing have been quite effective at stabilizing high-resolution GAN training. Overall, the following major improvements were made in the G network considering issues in StyleGAN:

- StyleGAN used a constant input c as the model input directly, it was modified to input C by adding noise and bias.
- Noise and bias were moved outside the style block.
- Only the standard deviation value of every feature map was modified instead of modifying both the standard deviation and the mean values.
- Demodulation module was introduced to overcome the droplet artifacts.

Weight Demodulation: Similar to StyleGAN, StyleGAN2 makes use of a normalization technique to provide styles from the W vector using learning to transform A into the source image. Here, the weight demodulation handles the droplet artifacts.

Lazy Regularization: StyleGAN2 computes regularization terms once after 16 mini-batches compared to StyleGAN, which computes both the main loss function and regularization for every mini-batch with heavy memory consumption and high computation cost. This change in approach is to compute the cost function, which has no major effects in terms of model efficiency but speeds up the training.

Path Length Regularization: Introducing path length regularization [22] allows the same displacement in the latent space that would produce the same magnitude change in the image space regardless of the value of the latent factor.

Removing Progressive Growing: Progressive growth in StyleGAN causes phase artifacts (location preference for facial features). StyleGAN2 overcomes the issue by using a different network design based on skip connections similar to that of ResNet architectures.

Adaptive Control Scheme: In order to have dynamic control over the augmentation strength parameter p to avoid over-fitting, an adaptive control scheme [23] has been used instead of manually tuning the augmentation strength. With the introduction of two heuristics to detect over-fitting in the discriminator,

we are going to increase the magnitude of the augmentation to have dynamic scheduling.

$$r_v = E[D_{train}] - E[D_{validation}] / E[D_{train}] - E[D_{generated}] \quad (1)$$

$$r_t = E[\text{sign}(D_{train})] \quad (2)$$

where r_v is the first heuristic which refers to the validation set results relative to the training set and images generated given in Eq. (1). r_t is the second heuristic that refers to the training set that generates positive discriminator outputs given in Eq. (2).

4 Experiments

Style transfer learning mechanism is used for model training. Transfer learning [24] is used to reduce the training data by using the weights of the model already trained on a dataset [7, 25–27].

4.1 Datasets

We applied the models to the brain tumor dataset [28, 29], as available via Kaggle [30]. This dataset includes 154 brain MRI samples and contains 3064 T1-weighted images with high contrast consisting of three kinds of brain tumors which are classified as Glioma, Meningioma, and Pituitary Tumor, as shown in Fig. 2.

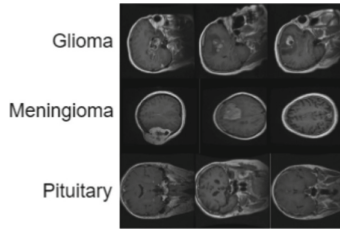


Fig. 2. Brain tumor dataset sample images. Each row represents one type of tumor.

4.2 Implementation Details

We resize all training images to 512×512 resolution. We used *Google Colab Pro* platform for the training model as it allows access to faster GPUs which helps in speeding up the training. The model was trained on a Tesla P100 GPU with 25 GB RAM. For monitoring and managing GPU resources, NVIDIA System Management Interface (nvidia-smi) driver version 460.32.03 and Cuda version 11.2 has been used for the management and monitoring of NVIDIA GPU devices. We converted all images into TFR records, enabling StyleGAN2 ADA to read data and improving the import pipeline’s performance. We utilized pre-trained models trained on FFHQ dataset [10], BreCaHaD dataset [13], AFHQ [14].

4.3 Pre-trained Models

FFHQ512 [10] pre-trained model is trained on Flickr-Faces high-quality images (FFHQ) dataset. The FFHQ is an image dataset containing high-quality images of human faces. It offers 70,000 PNG images at 512×512 resolution that display diverse ages, ethnicity, image backgrounds, and accessories like hats and eyeglasses.

BreCaHaD [13] pre-trained model is trained on a dataset consisting of 162 breast cancer histopathology images that are distributed into 1944 partially overlapping crops of 512×512 . The dataset is widely used by the biomedical and computer vision research community to evaluate and develop novel methods for tumor detection and diagnosis of cancerous regions in breast cancer histopathology images.

Animal FacesHQ [15] (AFHQ) pre-trained model is trained on a dataset of 15,000 high-quality animal face images at 512×512 resolution in three domains of cat, dog, and wildlife, with 5000 images per domain. AFHQ sets a more challenging image-to-image translation problem by having three domains and diverse images of various breeds. The images are vertically and horizontally aligned. The low-quality images were manually discarded. We used weights from the AFHQ (Cat) and AFHQ (Wild) pre-trained models.

4.4 Evaluation Metrics

Fréchet Inception Distance (FID) [31] is a metric for quantifying the distance between two distributions of images P_r and P_g where P_r is the probability distribution of real images, and P_g is the probability distribution of generated images. It is used to evaluate the quality of generated images and the performance of GANs. FID is defined as:

$$FID(\mu_r, \Sigma_r, \mu_g, \Sigma_g) = \|\mu_r - \mu_g\|_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \quad (3)$$

where (μ_r, Σ_r) and (μ_g, Σ_g) denote the mean vectors and covariance matrices of the Gaussian approximations for real and generated samples, respectively. The lower the FID value, the better the generated image quality. Kernel Inception Distance (KID) [32] is a metric which measures the dissimilarity between two probability distributions P_r and P_g using samples drawn independently from each distribution. KID is defined to be the squared maximum mean discrepancy (MMD) between the Inception features of real and generated images. A cubic polynomial kernel is used to map the real and generated images from the feature space of the Inception network, which is defined as:

$$KID(x, y) = \left(\frac{1}{d} x^T y + 1 \right)^3 \quad (4)$$

5 Results and Discussion

FID is a commonly used metric for assessing the quality of the images generated by a model. However, FID is prone to be dominated by the inherent bias when the

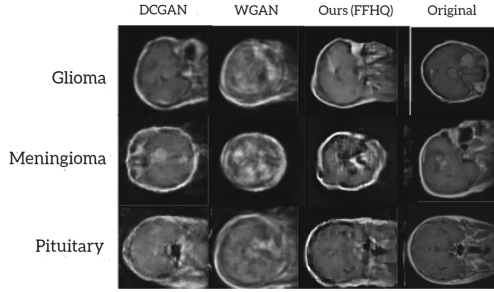


Fig. 3. Comparative analysis of generated brain tumor MRI samples using DCGAN [20], WGAN [33], FFHQ (Ours), and the original sample. Each row corresponds to images from three different classes, namely; Glioma, Meningioma, and Pituitary.

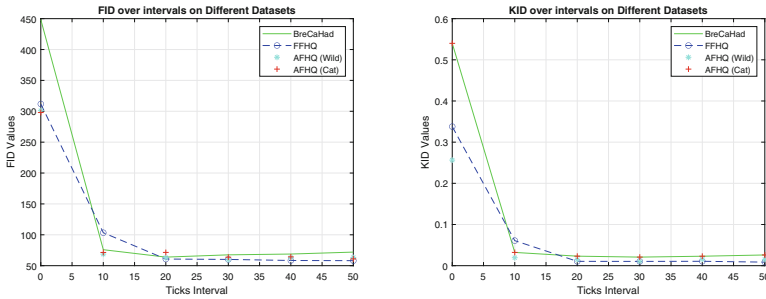


Fig. 4. Trend of FID and KID using transfer learning in generating synthetic brain MRI with tumor images of 512×512 resolution trained on StyleGAN2 ADA.

number of real images is not large enough. Hence, we used KID as an additional metric for evaluating our model performance. The trend for FID and KID using different pre-trained models is shown in Fig. 4. FID and KID values were recorded on every ten tick intervals for FFHQ, BreCaHaD, and AFHQ models, where tick interval refers to the number of iterations after the training snapshot has been taken. The results are summarized in Table 1.

The trend indicates a decrease in FID and KID values as tick intervals increase for FFHQ and AFHQ (Cat) models. For BreCaHaD and AFHQ (Wild) models, a decrease can be observed from 0–30 tick intervals. After that, an increase can be seen for both FID and KID values. Amongst the models evaluated, the BrecaHaD model had the worse performance, having the highest FID and lowest KID values.

Qualitative results of initially synthetically generated brain tumor images by different models are shown in Fig. 5. Using the best FID and KID of the pre-trained models, the brain MRI images generated by transfer learning are shown in Fig. 6. By analyzing our results, we find that FFHQ gives the lowest

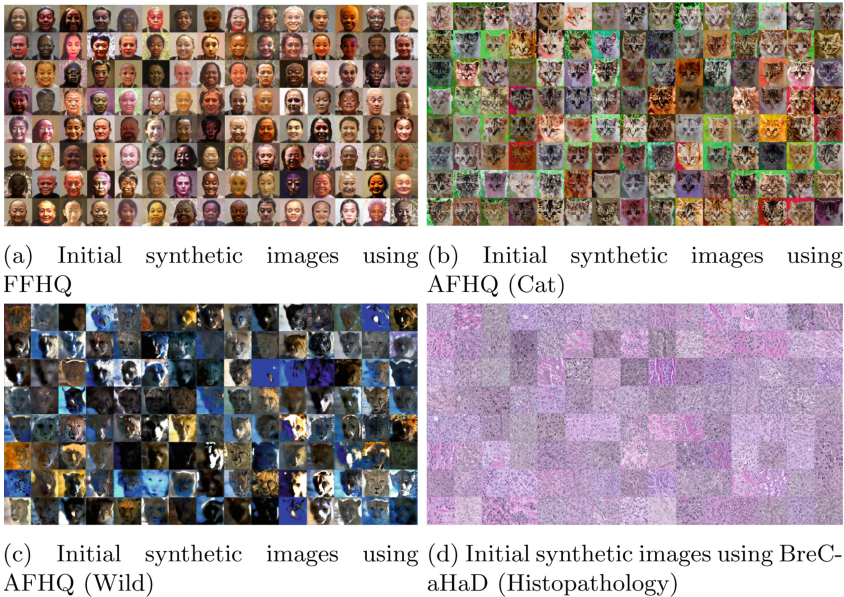
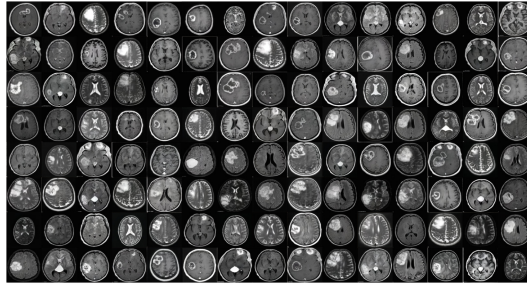


Fig. 5. Samples of initially generated images. Results show a visualization of the weights of the StyleGAN model trained on FFHQ, AFHQ (Cat), AFHQ (Wild), and BreCaHad images.

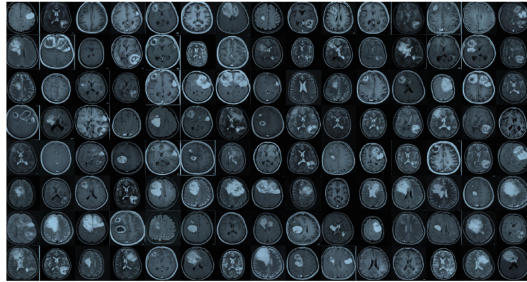
FID of 58.1097 and KID of 0.00862, and generates better quality images when compared with other pre-trained models. Figure 3 shows a comparison of the images generated using DCGAN [20], WGAN [33] and Ours (FFHQ) model using the brain tumor dataset. The results indicate Ours (FFHQ) generates better quality images when compared with the other GAN models.

Table 1. Results

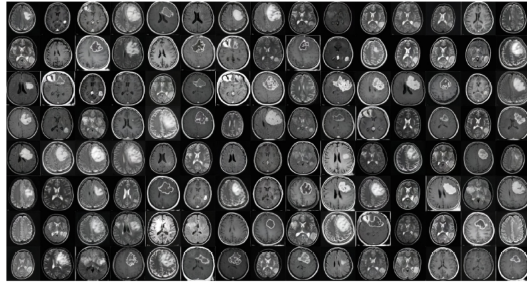
Pre-trained models	FID	KID
FFHQ	58.1097	0.00862692
AFHQ Cat	60.9486	0.01049849
BreCaHaD	67.5336	0.02081763
AFHQ Wild	59.7498	0.0109629



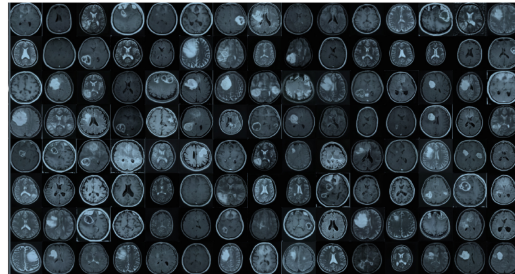
(a) Synthetic MRI brain tumor images on best FID and KID for the FFHQ pre-trained model.



(b) Synthetic MRI brain tumor images on best FID and KID values for the AFHQ (Cat) pre-trained model.



(c) Synthetic MRI brain tumor images on best FID and KID values for the AFHQ (Wild) pre-trained model.



(d) Synthetic MRI brain tumor images on best FID and KID values for the BreCaHaD pre-trained model.

Fig. 6. Synthetic images generated for the best FID and KID values.

6 Conclusion and Future Work

In this work, we presented a useful application of Adaptive StyleGANs for synthetic brain MRI images. Our results show that high-quality realistic MRI brain tumor images can be generated using pre-trained GAN models. By analyzing our results, we find that FFHQ gives the lowest FID and KID and generates better quality images when compared with other pre-trained models used in this research. This work will motivate other researchers to leverage the potential of StyleGAN in many applied domains of medical imaging research. For example, the models can be explored for modeling to detect the presence of tumors in body parts, perform tissue segmentation when training largely suffers due to the unavailability of high-quality data, and cross-modality medical image generation. The future work of this research is to explore the use of StyleGAN2-based architectures for the synthesis of high-quality medical images of other modalities such as Computed Tomography and histopathology images. It would be interesting to evaluate the model performance with other smaller medical imaging datasets. Similarly, an interesting direction is to explore the use of StyleGAN2 with StyleCLIP [19] for generating medical images from the text description.

References

1. Ker, J., Wang, L., Rao, J., Lim, T.: Deep learning applications in medical image analysis. *IEEE Access* **6**, 9375–9389 (2017)
2. An, G., Akiba, M., Omodaka, K., Nakazawa, T., Yokota, H.: Hierarchical deep learning models using transfer learning for disease detection and classification based on small number of medical images. *Sci Rep.* **11**(1), 1–9 (2021)
3. Goodfellow, I., et al.: Generative adversarial networks. *Commun. ACM* (2020)
4. Yi, X., Walia, E., Babyn, P.: Generative adversarial network in medical imaging: a review. *Med. Image Anal.* **58**, 101552 (2019)
5. Ali, H., Biswas, R., Ali, F., Shah, U., Alamgir, A., Mousa, O., Shah, Z.: The role of generative adversarial networks in brain MRI: a scoping review. *Insights Imaging* **13**(1), 1–15 (2022)
6. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. *Adv. Neural. Inf. Process. Syst.* **33**, 12104–12114 (2020)
7. Zhao, Y., Li, C., Yu, P., Gao, J., Chen, C.: Feature quantization improves GAN training. *arXiv preprint [arXiv:2004.02088](https://arxiv.org/abs/2004.02088)* (2020)
8. Işın, A., Direkoğlu, C., Şah, M.: Review of MRI-based brain tumor image segmentation using deep learning methods. *Proc. Comput. Sci.* **102**, 317–324 (2016)
9. Díaz-Pernas, F.J., Martínez-Zarzuela, M.: A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network. *Healthcare* **9**(2), 153 (2021)
10. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: *IEEE CVPR*, pp. 4396–4405 (2019)
11. Gao, H., Pei, J., Huang, H.: ProGAN: network embedding via proximity generative adversarial network. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1308–1316 (2019)
12. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: *IEEE CVPR*, pp. 8107–8116 (2020)

13. Aksac, A., Demetrick, D.J., Ozyer, T., Alhajj, R.: BreCaHAD: a dataset for breast cancer histopathological annotation and diagnosis. *BMC. Res. Notes* **12**(1), 82 (2019)
14. Kumari, N., Zhang, R., Shechtman, E., Zhu, J.-Y.: Ensembling off-the-shelf models for GAN training. In: *IEEE CVPR*, pp. 10651–10662 (2022)
15. Choi, Y., Uh, Y., Yoo, J., Ha, J.-W.: StarGAN v2: diverse image synthesis for multiple domains. In: *IEEE CVPR*, pp. 8185–8194 (2020)
16. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint [arXiv:1506.03365](https://arxiv.org/abs/1506.03365)* (2015)
17. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint [arXiv:1710.10196](https://arxiv.org/abs/1710.10196)* (2017)
18. Ahmad, W., Ali, H., Shah, Z., Azmat, S.: A new generative adversarial network for medical images super resolution. *Sci. Rep.* **12**(1), 1–20 (2022)
19. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: StyleCLIP: text-driven manipulation of StyleGAN imagery. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2065–2074 (2021)
20. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, vol. abs/1511.06434 (2016)
21. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: *ICML*, pp. 8748–8763. PMLR (2021)
22. Peebles, W., Zhu, J.-Y., Zhang, R., Torralba, A., Efros, A.A., Shechtman, E.: GAN-supervised dense visual alignment. In: *IEEE CVPR*, pp. 13470–13481 (2022)
23. Ma, X., Jin, R., Sohn, K.-A., Paik, J.-Y., Chung, T.-S.: An adaptive control algorithm for stable training of generative adversarial networks. *IEEE Access: Pract. Innov. Open Solut.* **7**, 184103–184114 (2019)
24. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. *J. Big data* **3**(1), 1–40 (2016)
25. Wang, Y., Gonzalez-Garcia, A., Berga, D., Herranz, L., Khan, F.S., Weijer, J.V.D.: MineGAN: effective knowledge transfer from GANs to target domains with few images. In: *IEEE CVPR*, pp. 9332–9341 (2020)
26. Iqbal, T., Ali, H.: Generative adversarial network for medical images (mi-GAN). *J. Med. Syst.* **42**(11), 1–11 (2018)
27. Wang, Y., Wu, C., Herranz, L., van de Weijer, J., Gonzalez-Garcia, A., Raducanu, B.: Transferring GANs: generating images from limited data. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 218–234 (2018)
28. Cheng, J., et al.: Enhanced performance of brain tumor classification via tumor region augmentation and partition. *PLoS One* **10**(10), e0140381 (2015)
29. Singh, P., Sizikova, E., Cirrone, J.: CASS: cross architectural self-supervision for medical image analysis. *ArXiv*, vol. abs/2206.04170 (2022)
30. Chakrabarty, N.: Brain tumor dataset. *Kaggle*
31. Nunn, E.J., Khadivi, P., Samavi, S.: Compound frechet inception distance for quality assessment of GAN created images. *arXiv [cs.CV]* (2021)
32. Knop, S., Mazur, M., Spurek, P., Tabor, J., Podolak, I.: Generative models with kernel distance in data space. *Neurocomputing* **487**, 119–129 (2022)
33. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: *34th ICML*, vol. 70, pp. 214–223. PMLR (2017)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Responsible and Trustworthy Artificial Intelligence



Challenges Associated with the Adoption of Artificial Intelligence in Medical Device Software

Karla Aniela Cepeda Zapata¹ , Tomás Ward³ , Róisín Loughran¹ ,
and Fergal McCaffery^{1,2} 

¹ Dundalk Institute of Technology, Dundalk, Ireland
{karla.cepada, roisin.loughran, fergal.mccaffery}@dkit.ie

² Lero, Limerick, Ireland

³ Insight SFI Research Centre for Data Analytics, Dublin, Ireland
tomas.ward@dcu.ie

Abstract. The utilization of Artificial Intelligence (AI) has changed and enhanced several industries across the world, such as education, research, manufacturing and healthcare. The potential of AI to create new and enhanced applications that can benefit patients and physicians has created interest and enthusiasm, especially in a Medical Device Software (MDS) context. Although, the adoption of AI in MDS has also brought concerns for regulatory agencies and policymakers. The complexity of AI has challenged the standard requirements set by regulatory agencies, especially in the context of the differences between traditional MDS and AI. Additionally, the unique capacity of AI to continuous learning for optimal performance in real-world settings may also bring potential harm and risk to patients and physicians. The challenges discussed in this paper are in relation to: (1) *Software Development Life Cycle (SDLC) frameworks*; (2) *learning processes and adaptability of AI algorithms*; (3) *explainability and traceability*; and (4) *conflictive terminology*. At the end of this paper, conclusions and future work are presented to contribute to the safety and methodical implementation of AI in health care settings.

Keywords: Artificial Intelligence · Medical Device Software · Healthcare · Challenges

1 Introduction

Artificial Intelligence (AI) is revolutionizing many fields of science and technology. Advances in technology have changed and evolved the definition of AI, bringing new discussions and, in some cases, confusion to the scientific community [1, 2]. In simple terms, AI refers to machines that *mimic human reasoning* for problem-solving [1, 3], although, the definition of AI has also been used as an *umbrella* term that covers other techniques, such as Machine Learning (ML) and Deep Learning (DL) [2–4].

AI has experienced massive growth in recent years. The global AI market size grew by 37% in 2020. Despite the fact that the global AI market size might slightly slow

down by the end of 2025, the European market will grow with a compound annual growth rate of 29.6% [5]. This massive growth of AI reflects the enthusiasm within institutions and businesses to embrace AI. The advances this technology has brought to society have changed and enhanced a wide range of industries worldwide, such as education, manufacturing, and healthcare. However, the adoption of this technology has also brought many challenges, and within this paper, we will discuss both the importance and challenges that AI has brought to the healthcare domain.

The paper is structured as follows: Sect. 2 discusses the importance of AI in Healthcare from a data and application perspective; Sect. 3 presents challenges in healthcare in relation to (1) *SDLC frameworks*, (2) *adaptability of algorithms*, (3) *explainability and traceability*, and (4) *terminologies*; Sect. 4 presents some of the efforts conducted by regulatory bodies; finally, Sect. 5 introduces a summary and directions for future work.

2 The Importance of Artificial Intelligence in Healthcare

The promises of AI in healthcare are aimed at improving and innovating different areas such as *medical practices*, research and management [6]. Some examples of high-value AI applications for medical practices could be easier detection of disease, fast action on urgent events, improved confidence in diagnosis, personalized treatments, drug discovery, and management of critical conditions [6, 7]. It must be noted that none of these applications would be possible without data.

Despite the fact that healthcare was found to have one of the smallest global dataspaces¹ in 2018, this industry will experience rapid growth, reaching a compound annual growth rate of 36% by the end of 2025 [8]. Possible reasons for this increment are due to the advances in digital care, healthcare analytics, and advances in the imaging technology industry [6, 8]. From this perspective, AI will be increasingly important in healthcare to exploit the vast amount of medical data generated daily and, therefore, empower the sector to provide better assistance for patients and physicians.

Within a Medical Device Software² (MDS) context, AI has been adopted to improve medical products and handle large volumes of data for interpretation [9, 10]. AI can be categorized into two major types of MDS: *Software as a Medical Device* (SaMD) and *Software in Medical Devices* (SiMD) [11]. Based upon the definition of the *International Medical Device Regulator Forum* (IMDRF), SaMD is software that is used on its own for one or more medical purposes, and it does not necessarily have to be part of the hardware to achieve the intended use. On the other hand, SiMD is *part of* a Medical Device (MD), which means that the software is utilized to assist an MD in performing the intended use [12]. In Europe, the Medical Device Regulations (MDR) covers both terminologies SiMD and SaMD by using the term MD³. AI could fall into either SiMD or SaMD, which is generally referred to as AI-enabled MD⁴ (AI-MD) [11]. A limited

¹ Global dataspaces refers to all data used for digital transformation. This data is created, captured, or replicated in datacentres, enterprise-hardened infrastructures, and endpoints [8].

² The use of the word *device* is implemented in this paper as a synonym of MDS.

³ In this paper, for general purposes, the term MDS is used to refer to either SaMD or SiMD.

⁴ The IMDRF use the term ML-enabled MD to specifically refer to ML techniques. Although, in this paper, AI-enabled MD is used to explicitly cover more branches of AI.

number of AI-MDs are already approved for the market by regulatory bodies. The Food and Drug Administration (FDA) approved 222 AI-MDs from 2015 to 2020 [13], and the agency has indicated that most of these devices were categorized as AI-enabled SaMD [7, 9]. Meanwhile, in Europe, Notified Bodies approved 240 MDs that contained AI between 2015 and 2020 [13].

In general terms, the MD industry, like aircraft, autonomous cars, and nuclear industries, is classified as safety-critical due to its consequences in terms of harm if *something goes wrong*, i.e., serious injuries or even potential loss of life to patients [14]. Historical examples of the lack of regulation and control resulted in devastating consequences. A particular event occurred decades ago in relation to SiMD, which was one of the starting points in *explicitly* enhancing software regulation procedures and requirements [15]. The main character of this unfortunate accident was the *Therac-25*, a software-controlled radiation machine for tumour treatment, late in the 80 s. It was discovered that the device affected patients as a consequence of the high-energy radiation delivery, causing severe injuries and death [16]. Consequently, this event triggered actions from policymakers on *how to regulate and ensure the software is safe* in MDs [15], which eventually included SaMD as a response to technological advances. Many lessons have been learned from this event. However, there is now an alarm that similar events like the *Therac-25* might occur again by enabling AI-MD, given the current uncertainty of regulatory guidance regarding adopting this technology in MDS. Even more, for AI to be adequately incorporated into the MDS industry, challenges must be considered as the *complexity* and *non-deterministic* behaviour of AI technology.

3 Challenges Introduced Through Artificial Intelligence in Healthcare

The adoption of AI in MDS has challenged the traditional regulatory framework. Manufacturers are facing new struggles related to the integration of AI in MDs. Within this paper, the challenges explored are in regard to various features between traditional MDS and AI, transparency, and terminology. First, we discuss the Software Development Life Cycle (SDLC) in general terms to illustrate the differences between AI and traditional MDS. Subsequently, we introduce adaptive AI algorithms as a *unique* feature and discuss the challenges this poses. Lastly, transparency and terminology are presented as challenges, although proper implementation may enhance the safety and trustworthiness of AI-MDs. The incorporation of AI in healthcare also magnified ethical and social issues such as fairness and bias. Despite the great importance of these ethical and social challenges, these are beyond the scope of this paper.

3.1 Different Software Development Life Cycle Frameworks in Traditional Medical Device Software and Artificial Intelligence

Traditional MDS and AI-MD have different characteristics [17, 18]. One possible reason for the struggle to regulate AI-MD is the difference in the structure of the Software Development Life Cycle (SDLC) process for AI in comparison to traditional MDS. Regulatory agencies have validated, cleaned and approved several MDS, although barriers have been encountered when it comes to AI due to its complexity [7].

In simple terms, traditional MDS has a defined and *deterministic* set of instructions that, based on specific inputs, a specific output is generated (see Fig. 1, Traditional MDS diagram) [3]. Different SDLC frameworks have been created and adopted to design, develop, and test traditional MDS, from plan-driven approaches, like *waterfall* and *v-model*, to more *adaptable* ones, like *Agile frameworks* [19]. However, AI has modified *the rules of this game*. AI models are fed with data containing features, i.e., inputs, and a target, i.e., output, to be trained and tested (see Fig. 1, AI diagram) [3]. In AI, *input(s)* and *output* comprise the dataset used to train and test a model, and the *AI technique* could be any ML technique operating via a supervised learning paradigm. These elements, *input + output + AI technique*, are used to build a *model* which represents the training dataset's patterns [3].

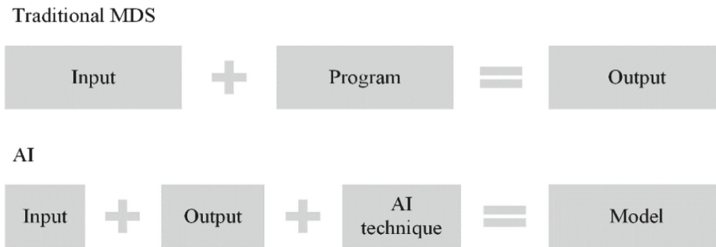


Fig. 1. Differences between traditional MDS and AI [3].

The functional differences between SDLCs for traditional MDS and AI analysed were related to (1) *data*, (2) *the set of skills required from practitioners*, and (3) *a lack of modular programming* [18]. To mainly focus on (1), when comparing the general SDLC tasks between traditional MDS and AI, it is possible to notice significant differences related to data: the need for data *to learn* and the *different SDLC* frameworks. *Data engineering* processes must be performed before training and testing a model to acquire high accuracy of the outcome. Although the *data engineering* stage is pre-conditioned by the data – with none or insufficient data, it would not be possible to build an AI application. The functional differences (2) and (3) are not discussed as these are out of the scope of this paper.

In addition, the implementation of AI in the MD industry may require a new SDLC framework structure due to the stochastic behaviour of AI algorithms. Despite the fact that there are several SDLCs for AI projects, these life cycle frameworks may not be suitable for the MDS industry and should be revisited given the possible absence of regulatory requirements such as quality control, documentation design, and monitoring procedures [17].

3.2 Risks Associated with the Adaptability of Artificial Intelligence

Another different feature between traditional MDS and AI-MD is the capability of *learning*. In high-level terms, the FDA has classified two types of algorithms: *locked* and *unlocked*. For those algorithms labelled as locked, these are not retrained over time once

the MDS has been deployed and approved; hence, these algorithms always provide the same outputs after feeding the same inputs. However, unlocked⁵ algorithms are those designed to continuously learn under post-market conditions, e.g., from real-world data, in an automated process [7]. In other words, the essential difference between locked and unlocked algorithms is that the unlocked ones are upgraded by themselves, i.e., the software, whereas locked algorithms are upgraded by human intervention via new software versions [12]. In the document *Machine Learning-enabled Medical Devices: Key Terms and Definitions* by the IMDRF, the learning process from unlocked algorithms is called *continuous learning*, while for locked algorithms it is *batch learning* [11]. It is essential to clarify that some AI-MDs are also categorized as *locked* devices because manufacturers do not have the intention to retrain the model during operation [7, 11].

Regulatory entities and policymakers have drawn attention to this ability of AI, as their *unique* position among traditional MDS and the benefits of optimizing performance through continuous learning [7], preserving the prediction accuracy of the AI model [20]. However, regulatory agencies also recognize AI models' potential risk from this stochastic behaviour. The fact that autonomous and continuous learning from real-time data may instruct the AI itself to perform differently could bring unpredictable consequences and potentially harm and endanger patients, and consequently, questioning the idea of request to manufacturers for another premarket submission [7].

As part of a proposed new framework for modifications, the FDA outlined potential future changes in the performance to support the development of unlocked AI-MDs [7]. The IMDRF additionally discussed future changes in the structure of AI algorithms. However, the IMDRF group also mentioned future changes related to external factors that may alter and modify the performance of AI-MDs, such as alteration of the *data* (e.g., quality of inputs affected) and the environment setting (e.g., system operation upgrades) [11]. It is essential not to forget that these external factors may also affect *locked* AI-MDs. The adoption of *unlocked* AI-MDs, employed in a regulated and safe manner, might be also beneficial for changing environments, to which locked AI-MDs are not able to respond.

The challenge that AI brought to healthcare, including the differences mentioned in Subject. 3.1, generates the need for a new and *adaptable* SDLC framework for AI-MDs. Moreover, this new framework must fit the regulatory requirements for MD purposes to ensure trustworthy and safe AI devices from the beginning of their development [9, 21]. Additionally, as a consequence of the *adaptability* of AI algorithms, this may challenge other regulatory requirements, such as *management* process, *risk* and *quality* management, *clinical evaluation*, *manufacturing facility*, *control design*, and *post-market surveillance* [22, 23]. In particular, *transparency* has taken a critical role in implementing *unlocked algorithms* in a safety-critical environment such as healthcare [9, 11, 24].

3.3 Achieving Explainability and Traceability in AI – Essential to Satisfy Regulators

Transparency is defined in various ways depending on the scenario and discipline. Generally speaking, this refers to the possibility of accessing information [25]. Although,

⁵ Also referred by the FDA as *adaptive*.

particularly in an Information Technologies (IT) environment, it was identified that the use of the word *transparency* refers to the degree to which the information and functionality of a system are *invisible* to users [25]. In the medical domain, the “*condition of being transparent*” is an essential element of end-to-end traceability that establishes a better relationship with patients, enhances services, reduces risk, and increases trust in physicians and the health care system [26]. Despite the fact that challenges to *transparency* remain in medical care practices [26], the implementation of AI in healthcare has magnified the current ones and raised new disputes in the area. In the context of trust, policymakers have agreed that *transparency* is one of the essential requirements for achieving *trustworthy AI applications* [9, 27, 28]. Moreover, the word *transparency* has been used to *encapsulate* the conditions of making AI more *visible*. These qualities are generally related to *explainability* and *traceability* [24, 29, 30].

Explainability. This quality of transparency is related to the structure of the AI algorithms and their visibility to users. Typically, AI models receive specific inputs, e.g., patient data or clinical images, and generate a prediction or classification based on *internal procedures* [31]. Often, these internal procedures are hidden from physicians, providing no *explanation* of the decision-making process of AI models [24], which may compromise trust in the prediction of the AI algorithms. Moreover, this provides an insufficient level of understanding of these algorithms to physicians and is referred to as the *black box problem* [30]. Some ML models are easier to explain, e.g., in Regression Analysis is possible to refer to the weights given to the variables to understand their relationship, whereas the visualization style provides an understanding of Decisions Trees [30, 32]. However, in the case of more *sophisticated* AI techniques, such as DL and Natural Processing Language, the explanation of AI decisions becomes more and more complex [9, 30]. There is a realization that there is a *trade-off*: between the *best* performance from the model (which is often the *least explainable* algorithm) and those models having *inferior* performance but being the *most explainable* [9]. Due to the complexity of AI algorithms, the challenge for explainability is to select the best approach to describe the AI-MDs [24].

Traceability. Regulatory agencies have recognized the crucial role of manufacturers in achieving transparency, in this case by designing proper traceability of the AI-MD [7]. *Traceability* in IT is the appropriate design of the life cycle of a system in terms of requirements in an onward and regressive sequence [33, 34]. In the MDS industry, traceability refers to the proper documentation of the system’s design, and it is critical as it is utilized as a *risk control* mechanism [33]. In short, *control design* aims to ensure a plan for the development process is designed, increasing the probability of correct translation of the user needs into an MDS, increasing the system’s quality and assuring safety before being placed on the market [35]. It has been suggested to document the entire process of AI SDLC and implementation [24]. Although, manufacturers are still struggling to document AI models due to the lack of mechanisms and guidance on how to do it [10]. Moreover, another challenge is that some AI applications are rarely delivered with complete *traceability* documentation due to the preference of manufacturers to keep the functionality, data, and algorithms private and confidential for Intellectual Property purposes [23].

The importance of *explainability* and *traceability* in AI-MD is not just to increase physicians' and patients' trust [23] but for troubleshooting (e.g., diagnose and trace incorrect outcomes) and liability purposes (e.g., who is responsible for mistakes?) to minimize risk and assist adoption of AI [9, 23, 24]. Additionally, *transparency* would play a significant role in clarifying functionality, learning approach (i.e., *batch* or *continuous*), and changes over time [7, 36]. However, challenges in the selection of approaches to explain AI algorithms and the lack of guidance to document the life cycle of AI-MDs remain, which may require adjustments including the introduction of best practices in the documentation of AI projects in the MDS industry. Even more, the lack of transparency aggregates more challenges to other areas, such as *cybersecurity*, *validation*, and *verification* procedures [37]. Furthermore, the erroneous use of *terminologies* in documentation may limit the explainability of the AI-MD.

3.4 Conflict Use of Terminologies

Another challenge exists in terms of the *terminology* and *taxonomy* of AI [21]. This complication arises as there are different fields working together in the MDS industry, such as Artificial Intelligence, Data Science, Computer Science, Healthcare, and Regulatory agencies. Most of these disciplines have adopted different terminologies, with similar words but different meanings, leading to conflict and confusion. A simple example of this is the word *validation*, which is used in AI and Data Science as a technique to evaluate the performance of the model, whereas, from a regulatory perspective, this is to evaluate whether the user needs have been met [11].

Additionally, the use of terminologies from one discipline in another has been identified as another challenge. Researchers described how a study was conducted to identify the number of devices approved in the US and Europe and reported issues when exploring the documentation of the device. It was claimed that there were discrepancies in the use of the terms associated with AI and ML. This issue, and the lack of transparency in terms of the documentation, made the identification of the AI-MD cumbersome [13]. Additionally, the possible misuse of terminologies may increase and create barriers to the development process of AI-MD [21].

There should be a commitment from standards organizations and stakeholders to overcome terminology challenges. From a standards body perspective, their intention and purpose are to harmonize terminologies and taxonomies [21]. Whereas stakeholders should adhere to the standards developed by the standardization bodies when researching and developing MDs in line with regulations to ensure proper and consistent implementation of such terminologies across the industry [21].

4 State-of-the-Art from Regulators

In Europe, the Medical Device Directives (MDD) was replaced with a new version named *Medical Device Regulations* (MDR). This new regulation was enacted in May 2021, and it was a response to the technological advances in the medical device industry [38]. A study [38] revisited the MDR to verify whether the new changes would improve performance and safety in AI-MD. Despite the fact that AI is not mentioned in the

document, the MDR would likely improve the performance and safety of most of the AI-MDs due to the new risk classification rules for software [38]. Based on this, it seems that AI-MDs would probably be classified in a higher risk classification, and therefore, such devices must be developed in a manner that is deemed safe before entering the European market. Although, it was also claimed that there is a lack in the evaluation process and external validation, which may affect the performance of AI-MD [38]. Besides the MDR, in April 2021, the European Commission released a draft of the AI Act to regulate and harmonize AI technologies across the Union [39]. The AI Act is based on a risk approach and describes a set of rules to classify AI systems as *minimal to little*, *limited*, *high*, and *unacceptable* risk. The AI Act proposed a list of requirements for high-risk AI systems. These requirements are related to risk and data management, technical documentation, record-keeping and traceability, transparency, human oversight, and adequate level of accuracy, robustness, and cybersecurity. In terms of adaptability, the AI Act proposes that providers must establish how the AI system and its performance would change over time. Moreover, post-market monitoring was established as a key requirement for adaptive AI systems in order to perform corrective actions more efficiently.

In April 2019, the FDA released a discussion paper in which it proposed a new *Regulatory Framework for Modifications in AI/ML-enabled SaMD* [7]. This framework includes a *predetermined change control plan* (PCCP) in order to assist manufacturers in the development of unlocked AI-MDs. The PCCP contains two sections: *pre-specifications* (PS) and *Algorithm Change Protocol* (ACP). The PS contains a list of future modifications related to the structure of the AI model, as it is expected that most of these will occur after the retraining process [7, 10]. The agency identified three changes in AI-MD after retraining: (1) performance; (2) inputs used in the model; and (3) the intended use of the device [7]. Whereas the ACP is associated with the *step-by-step* implementation of *methodologies* for future changes, i.e., procedures on *how* the algorithm will be retrained and change in post-market data conditions [7]. With the implementation of the PCCP, it is expected that AI-MDs will remain safe after retraining in post-market conditions [10]. Subsequently, the FDA held an open discussion⁶ with stakeholders in relation to this new proposed framework. The feedback was analysed, and in 2021 the FDA released an *Action Plan* based on the comments and suggestions from the open discussion [10]. In relation to the PCCP, the FDA reported that stakeholders claimed that the list of future modifications was “relevant and appropriate” but limited [10]. As a response to the feedback from stakeholders, the FDA is currently working on expanding the list of modifications, which will be included in a new draft guidance of ACP [10]. Another point from this list was related to *transparency to users*, in which the FDA plans to promote transparency via public workshops and labelling training for manufacturers [10].

The IMDRF published in May 2022 the final document *Machine Learning-enabled Medical Devices: Key Terms and Definitions*. This document is a result of the efforts of the group of regulators to harmonize relevant terms around ML technologies in the MDS industry. The baseline of this document is the standards ISO/IEC DIS 22989 and ISO/IEC TR 24027, related to IT and AI terminologies and bias, respectively. In a nutshell, the

⁶ Access to the archive discussion: <https://www.regulations.gov/document/FDA-2019-N-1185-0001>.

document covers definitions of Bias, Continuous Learning, types of learning approaches, and terms related to testing and training processes [11]. The IMDRF also included two types of changes in unlocked AI-MD: *to AI-MD* and *to AI-MD environment for data* [11]. Future *changes to AI-MD* refer to modifications to the *model*. Some changes to AI-MD include the retraining process with new data, additional tuning of hyper-parameters, and training of the model with different AI methods and algorithms [11]. On the other hand, *changes to the AI-MD environment for data* are related to external factors that affect the learning process and the AI model. Examples of this type of change are the alteration of the quality of the inputs provided by third sources, changes in clinical practices, and the population upon which the AI model was initially trained and tested during the development process may have changed [11].

5 Conclusions

Medical device software standards and regulations have evolved over many years to provide manufacturers with helpful guidance in developing safe medical device software. However, the increasing usage of AI in MDS presents challenges in terms of the traceability and explainability of such algorithms, and there is a need for greater guidance to manufacturers in relation to the development of the safety of MDs containing AI. The adoption of AI in MD has challenged the traditional regulatory framework and set barriers for manufacturers. Moreover, sometimes in AI is not possible to adequately design documents as the lack of guidance, standards, best practices, and harmonization of terminologies. These also may impact the transparency of AI applications.

We identify several future contributions to MDS and AI. A potential contribution in AI-MD is the adjustment of existing guidance and standards already applied to MDS but for an AI context [21]. It is fundamental to start with the development and standardization of the structure of AI-MD projects by designing a regulatory-friendly framework, revisiting and comparing SDLC frameworks commonly implemented for traditional MDS and AI [17] and, subsequently, tailoring them to an AI-MD context. It is assumed that most AI life cycle frameworks have been mainly employed for non-safety-critical environments. Hence, these frameworks should be inspected to verify whether they would satisfy the regulatory requirements for MD purposes. In addition, best practices, standards, and guidelines will be considered for the development of the framework in order to improve the explainability and traceability of AI-MDs. Additionally, human oversight and post-market monitoring will be considered in the design of this framework for risk mitigation purposes. Systems engineering, and socio-technical systems will be also considered. This work will provide a baseline for unlocked AI algorithms for future work.

We realize there are difficult challenges that need to be overcome in order to establish *universal rules and procedures* for AI, particularly, in healthcare, due to the diverse context, different pathologies, user cases, and the constant evolution of the technology [40]. *It will be challenging but not impossible.*

Acknowledgements. This work was financially supported by the HEA's Technological University Transformation Fund (TUTF), Biodesign Europe, and Dundalk Institute of Technology (DkIT).

References

1. Kok, J.N., Boers, E.J.W., Kusters, W.A., Putten, P., van der Poel, M.: Artificial intelligence: definition, trends, techniques, and cases. *Artif. Intell.* **1**, 270–299 (2009)
2. European Parliamentary Research Service: EU legislation in progress artificial intelligence act (2022)
3. Kotu, V., Deshpande, B.: Introduction. In: *Data Science: Concepts and Practice*, pp. 1–18. (2019)
4. Davenport, T., Kalakota, R.: The potential for artificial intelligence in healthcare. *Future Healthc. J.* **6**, 94–98 (2019). <https://doi.org/10.2139/ssrn.3525037>
5. Marketline: Global Artificial Intelligence (2021)
6. Datta, S., Barua, R., Jonali, D.: Application of artificial intelligence in modern healthcare system. In: Pereira, L. (ed.) *Alginates*. IntechOpen. (2019). <https://doi.org/10.5772/intechopen.90454>
7. Food and Drug Administration (FDA): Proposed regulatory framework for modifications to Artificial Intelligence/Machine Learning (AI/ML)-based software as a medical device (SaMD) - discussion paper and request for feedback (2019)
8. Reinsel, D., Gantz, J., Rydning, J.: *The Digitization of the World from Edge to Core* (2018)
9. Food and Drug Administration (FDA): Artificial Intelligence (AI) and Machine Learning (ML) in medical devices - executive summary for the patient engagement advisory committee meeting (2020)
10. Food and Drug Administration (FDA): Artificial Intelligence/Machine Learning (AI/ML)-Based Software as a Medical Device (SaMD) Action Plan (2021)
11. IMDRF Artificial Intelligence Medical Devices (AIMD) Working Group: Machine Learning-enabled Medical Devices: Key Terms and Definitions (2022)
12. IMDRF Software as a Medical Device (SaMD) Working Group: Software as a Medical Device (SaMD): Key definitions (2013)
13. Muehlematter, U.J., Daniore, P., Vokinger, K.N.: Health policy approval of artificial intelligence and machine learning-based medical devices in the USA and Europe (2015–20): a comparative analysis. *Lancet Digit. Health.* **3**, e195–e203 (2021). [https://doi.org/10.1016/S2589-7500\(20\)30292-2](https://doi.org/10.1016/S2589-7500(20)30292-2)
14. Knight, J.C.: Safety critical systems. In: *Proceedings of the 24th International Conference on Software Engineering - ICSE 2002*, p. 547. ACM Press, New York (2002). <https://doi.org/10.1145/581339.581406>
15. Papademetris, X., Quraishi, A.N., Licholai, G.P.: The FDA and software. In: *Introduction to Medical Software: Foundations for Digital Health, Devices, and Diagnostics*. Cambridge University Press, Cambridge (2022). <https://doi.org/10.1017/9781009091725>
16. Leveson, N.G., Turner, C.S.: An investigation of the Therac-25 accidents. *Computer (Long Beach Calif.)* **26**, 18–41 (1993). <https://doi.org/10.1109/MC.1993.274940>
17. Haakman, M., Cruz, L., Huijgens, H., van Deursen, A.: AI lifecycle models need to be revised. *Empir. Softw. Eng.* **26**(5), 1–29 (2021). <https://doi.org/10.1007/s10664-021-09993-1>
18. Amershi, S., et al.: Software engineering for machine learning: a case study. In: *IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 291–300. (2019). <https://doi.org/10.1109/ICSE-SEIP.2019.00042>
19. Ruparelia, N.B.: Software development lifecycle models. *ACM SIGSOFT Softw. Eng. Notes* **35**, 8–13 (2010). <https://doi.org/10.1145/1764810.1764814>

20. Nakatsugawa, M., et al.: The needs and benefits of continuous model updates on the accuracy of RT-induced toxicity prediction models within a learning health system. *Int. J. Radiat. Oncol. Biol. Phys.* **103**, 460–467 (2019). <https://doi.org/10.1016/j.ijrobp.2018.09.038>
21. Turpin, R., Hofer, E., Lewelling, J., Baird, P.: Machine learning AI adapting regulatory frameworks and standards. BSI and AAMI (2020)
22. Herron, M., Gallagher, J.: White Paper on Artificial Intelligence & Medical Devices: A New Regulatory Frontier. Mason Hayes & Curran (2021)
23. COCIR: Artificial intelligence in EU medical device legislation (2020)
24. Lekadir, K., Quaglio Gianluca, T., Garmendia, A., Gallin, C.: Artificial intelligence in healthcare (2020). <https://doi.org/10.1016/B978-0-12-818438-7.00013-7>
25. Turilli, M., Floridi, L.: The ethics of information transparency. *Ethics Inf. Technol.* **11**, 105–112 (2009). <https://doi.org/10.1007/s10676-009-9187-9>
26. Oettgen, P.: Transparency in healthcare - achieving clarity in healthcare through transparent reporting of clinical data. EBSCO Health (2017)
27. OECD: Recommendation of the council on artificial intelligence. OECD/LEGAL/0449 (2022)
28. High-level expert group on artificial intelligence: ethics guidelines for trustworthy AI. European Commission (2019)
29. Franca Salis, M.: A guide to Artificial Intelligence at the workplace. European Economic and Social Committee (2022)
30. Petch, J., Di, S., Nelson, W.: Opening the black box: the promise and limitations of explainable machine learning in cardiology. *Can. J. Cardiol.* **38**, 204–213 (2022). <https://doi.org/10.1016/j.cjca.2021.09.004>
31. Nuffield Council on Bioethics: Artificial intelligence (AI) in healthcare and research (2018)
32. Piltaver, R., Luštrek, M., Gams, M., Martinčič-Ipšić, S.: What makes classification trees comprehensible? *Expert Syst. Appl.* **62**, 333–346 (2016). <https://doi.org/10.1016/j.eswa.2016.06.009>
33. Regan, G., McCaffery, F., Mc Daid, K., Flood, D.: Medical device standards' requirements for traceability during the software development lifecycle and implementation of a traceability assessment model. *Comput. Stand. Interfaces* **36**, 3–9 (2013). <https://doi.org/10.1016/j.csi.2013.07.012>
34. Gotel, O.C.Z., Finkelstein, A.C.W.: Analysis of the requirements traceability problem. In: *Proceedings of the International Conference on Requirements Engineering*, pp. 94–101 (1994). <https://doi.org/10.1109/icre.1994.292398>
35. Food and Drug Administration (FDA): 21 CFR 820.30 design control guidance for medical device manufacturers (1997)
36. Char, D.S., Abràmoff, M.D., Feudtner, C.: Identifying ethical considerations for machine learning healthcare applications. *Am. J. Bioeth.* **20**, 7–17 (2021). <https://doi.org/10.1080/15265161.2020.1819469>
37. Mason Hayes & Curran: Insights healthcare AI in medical devices: key challenges and global responses. <https://www.mhc.ie/latest/insights/ai-in-medical-devices-key-challenges-and-global-responses>. Accessed 29 Aug 2022
38. Niemiec, E.: Will the EU medical device regulation help to improve the safety and performance of medical AI devices? *Digit. Health* **8**, 205520762210890 (2022). <https://doi.org/10.1177/20552076221089079>

39. European Commission: Proposal for a regulation of the European parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts (2021). <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206>
40. Sargent, S.L.: AI bias in healthcare: using ImpactPro as a case study for healthcare practitioners' duties to engage in anti-bias measures. *Can. J. Bioeth.* **4**, 112–116 (2021). <https://doi.org/10.7202/1077639AR>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





An Intelligent Empowering Agent (IEA) to Provide Easily Understood and Trusted Health Information Appropriate to the User Needs

Marco Alfano^{1,4} , John Kellett² , Biagio Lenzitti³ , and Markus Helfert^{1,4} 

¹ Innovation Value Institute, Maynooth University, Maynooth, Ireland
{marco.alfano, markus.helfert}@mu.ie

² Department of Emergency Medicine, Hospital of South West Jutland, Esbjerg, Denmark

³ Dipartimento di Matematica e Informatica, Università di Palermo, Palermo, Italy
biagio.lenzitti@unipa.it

⁴ Lero, Science Foundation Ireland Research Centre for Software, Limerick, Ireland

Abstract. Most members of the public, including patients, usually obtain health information from Web searches using generic search engines, which is often overwhelming, too generic, and of poor quality. Although patients may be better informed, they are often none the wiser and not empowered to communicate with medical professionals so that their care is compatible with their needs, values, and best interests. Intelligent Empowering Agents (IEA) use AI to filter medical information and assist the user in the understanding of health information about specific complaints or health in general. We have designed and developed a prototype of an IEA that dialogues with the user in simple language, collects health information from the Web, and provides tailored, easily understood, and trusted information. It empowers users to create their own comprehensive and objective opinion on health matters that concern them. This paper describes the IEA main characteristics and presents the results of subjective and objective tests carried out to assess the effectiveness of the IEA.

Keywords: Digital health · Patient empowerment · Intelligent agents · Tailored health communication · Artificial Intelligence · Big data · Machine learning

1 Introduction

Healthcare remains focused on disease management, and not on looking holistically at the health and wellbeing of the whole person [1]. The latter approach aims at empowering patients by helping them better manage their health [2–4]. Health literacy/education, information-seeking behavior, and shared decision making characterize an empowered person who understands his/her health/conditions and communicate with medical professionals to get care that is compatible with his/her needs, values, and best interests [5–7]. To be empowered in this way people/patients must:

1. have the necessary knowledge and self-awareness to **understand** their conditions and treatment options;
2. be able to make informed and conscious health choices (i.e., **decide**);
3. actively manage, with or without advice from medical professionals, their health and well-being (i.e., **act**).

Few applications for patient empowerment exist [1, 5]. Most members of the public, including patients, usually obtain health information from Web searches using generic search engines [4, 8], which is often overwhelming, too generic, outdated, and of poor quality [8, 9]. Although Artificial Intelligence (AI) could play an important role in health empowerment [10, 11], it presently mostly provides self-diagnosis apps, which act as substitute doctors and keep patients as passive recipients [12–14]. This paper provides further details and evaluation of an Intelligent Empowering Agent (IEA) [15] that exploits the whole Internet and uses AI to empower people/patients to obtain, through natural language, tailored, easily understood, and trusted health information from the Web.

2 Background and Motivation

2.1 Shortcoming of Current Available Conversational Agents for Patient Empowerment

Conversational agents are AI systems that simulate conversations with users, and inform them by generating easily understood dialogue. Only a limited number of studies have linked person/patient empowerment to AI and conversational agents. A literature review [15] found that:

- Conversational agents are mostly created for a specific condition.
- Empowerment is almost never addressed directly and, when it is, only some aspects are considered.
- Comprehension of health information/conditions is the least addressed step of empowerment.
- Information is seldom tailored to the needs of the user.
- The origin and veracity of the provided content is often not provided and all available information (e.g., on-line) is not used.
- User requirements, such as language complexity or information quality are not considered.

2.2 Requirements of Online Health Information Seekers

Previously published research and a literature review [4] have found that, when searching for health information on the Web, the main user requirements on the found information are:

- language complexity;
- information quality;
- information classification/customization (tailoring).

Language must be easy to understand and the information provided scientifically correct [16]. Moreover, health information should be tailored to the users unique needs and interests [17]. Since tailored health information is more personally relevant, it is more likely to be read, understood, and acted on [18–21]. As a consequence, empowering health information must be easy to understand (by a non-medical expert), of good quality (trustworthy), and tailored to the user specific profile and needs [4, 22].

2.3 Research Objective

By combining the potential of AI with the vast amount of health information available on the Web, the research aims to improve user empowerment by providing tailored health information that is trustworthy and easily understood, through an Intelligent Empowering Agent (IEA) that:

- interacts with the user to understand his/her profile and information need;
- retrieves information from the Web and uses an AI algorithm to customize it;
- presents this information as a tailored, intelligible, and trustworthy output that facilitates the users comprehension and decision making.

3 Principles of an Intelligent Empowering Agent

The components of an IEA are:

a User query

The user selects a complaint from a list or directly enters it as free text.

b User profile

The user profile is constructed from each user's current health status (e.g., symptoms and/or conditions), background health status (e.g., sex, age, gait, BMI, and comorbid illness), lifestyle information (e.g., sleep, drugs, and meal composition), dynamic health indicators (e.g., vital signs monitoring, physical activity monitoring and stress level), empowerment level (e.g., health literacy, motivation and gaining control), health and well-being needs (e.g., urgent health improvement and elective quality of life).

c Search engine

The search engine retrieves health information from the World Wide Web, health-data repositories, and internal information coming from previous searches (anonymized).

d AI algorithm

The AI algorithm takes the health information collected by the search engine and organizes it into categories (e.g., complaint description, alternate names, related complaints, and related diseases) by using a machine learning algorithm that analyses the section headings of the retrieved documents. The most appropriate information for the user is then chosen by using a decision tree algorithm and according to the following criteria [18, 19, 22]:

- **Language complexity**, to provide users with information that they can easily understand.
 - **Information quality**, to provide users with current, accurate, trustworthy, and unambiguous information.
 - **Custom information**, to provide users with tailored content (considering the user query and user profile).
- e **Output presentation.** Tailored relevant health information is provided on complaints (definitions of and related elements), diseases (definitions of and related elements), tests (descriptions of and related elements), and external information sources. A “traffic-light” color coding (i.e., red, amber, or green), that implies the need for an urgent consultation with a healthcare professional, is also provided. The overall process of the IEA is shown in Fig. 1.

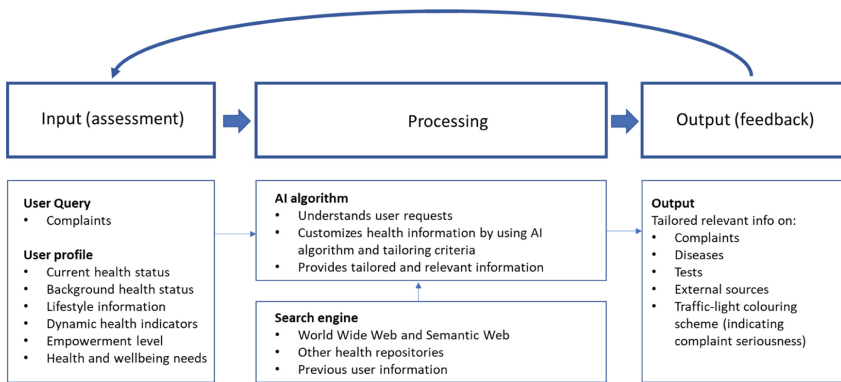


Fig. 1. IEA process model.

4 Implementation of the Intelligent Empowering Agent

An IEA prototype, the Conversational Health Agent for Person Empowerment (CHAPE) has been developed and can be accessed at <http://cohealth.ivi.ie/chape>. Initially, the user is asked to input age and sex and select a complaint from a list, which includes easily understood terms such as pain or discomfort, breathlessness, weakness or fatigue, etc. These complaints were derived from a classic textbook on symptoms [23], review of the literature, and expert opinion. Complaints are then further subdivided into more specific complaints, which include those most often associated with in-hospital death [24] and the commonest diagnoses encountered in primary care [25]. Alternatively, the user can directly type in any complaint in a free text area. Depending on the complaint selected and the user’s profile characteristics, such as age and sex, a further sub-list of possibly related complaints is presented, to help define the primary complaint more precisely (Fig. 2).

The interface is divided into three main sections. The top left section, 'Select age', has a dropdown menu set to '30-44 years'. The top right section, 'Select sex', has radio buttons for 'Male' (selected) and 'Female'. The middle left section, 'Indicate the problem', lists various symptoms: Pain or discomfort, Breathless, Weakness or Fatigue, Bleeding, Nausea/Vomiting, Diarrhea, Not feeling well, Impaired mobility, Impaired senses or thinking, Change in personal behaviour, and Change in body. Below this is a text input field 'Or type in a complaint'. The middle right section, 'Select a complaint', is a list of specific complaints: Pain or discomfort, Head/Neck, Chest, Abdomen, Arm and Shoulder, Back/Pelvis/Glute, Hip/Leg/Ankle/Foot, and Generalized pain or discomfort. The rightmost section, 'Select a complaint', lists further details: Burning abdominal pain, Pelvic pain, Sharp abdominal pain, and Suprapubic pain. Blue arrows indicate the flow from 'Indicate the problem' to 'Select a complaint' and then to the final 'Select a complaint' list.

Fig. 2. CHAPE interface allows users to specify their complaints in an easy and natural way.

An information window is then displayed (Fig. 3) and it contains:

- Complaint name with a background colour (red, amber, or green), which indicates the health risk.
- Complaint description.
- Alternate names of the complaint.
- Related complaints.
- Disease(s) associated with the complaint.
- Tests commonly used to further define the complaint.
- Web information related to the complaint, which is reliable and reputable.

The output window is titled 'Sharp abdominal pain (Complaint)' in a red header. It contains several sections: 'Description' with a paragraph about cramps; 'Alternate name' with a list including 'Stomach cramps', 'Gas pains', 'Intestinal colic', and 'Stomach pain'; 'Related complaints' with a list including 'Back pain', 'Blood in stool', 'Constipation', 'Diarrhea', 'Nausea', 'Rectal bleeding', 'Stomach bloating', 'Vomiting', and 'Vomiting blood'; 'Related diseases' with a list including 'Abdominal aortic aneurysm', 'Abdominal hernia', 'Acute kidney injury', 'Acute pancreatitis', 'Adrenal adenoma', 'Alcohol intoxication', 'Alcoholic liver disease', and 'Anal fissure'; 'Tests' with a list including 'Complete blood count', 'Electrolytes panel', 'Glucose measurement', 'Hematologic tests', 'Intravenous fluid replacement', and 'Kidney function tests'; and 'Web info' with a link to 'my.clevelandclinic.org' and a snippet of text about 'Abdominal (Stomach) Pain: Causes and Symptoms'.

Fig. 3. Output window containing information about the searched element and related complaints, diseases, tests, and Web sites.

When a related complaint, disease or test is selected a new information window is opened, which displays related complaints, diseases and tests ordered by taking into account the previous searches (Fig. 4).

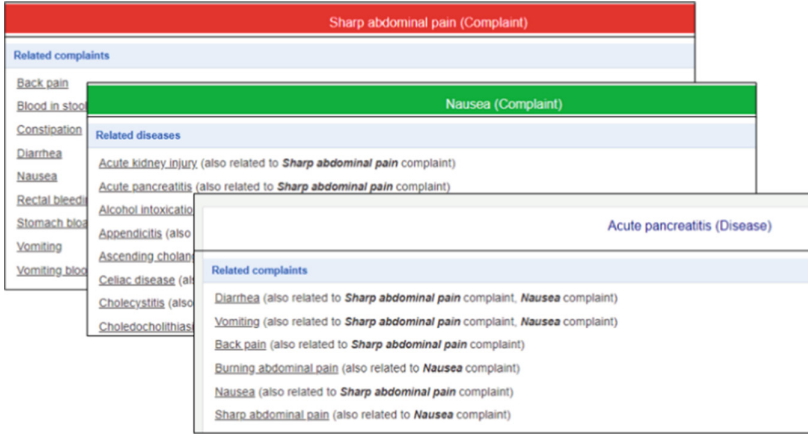


Fig. 4. Related complaints and diseases are ordered by considering their correlation with the previous searched elements.

The user's profile and chosen complaint are used to get a list of Web sites by means of Google's search engine. The web sites are then re-ranked by computing their **language complexity** and **information quality**.

Given a set of m Web pages, the **language complexity** of each Web page, is computed by considering the n words of the page (not considering the stop words) and computing the Word Familiarity (WF) as the number of Google results [22, 26, 27]. The *language complexity index* (LC) is then calculated according to the following formula (the higher the index the simpler the language):

$$LC = \left(\sum_{k=1}^n WF_k \div n \right) \div \max_m \left(\sum_{k=1}^n WF_k \div n \right) \quad (1)$$

The **information quality** is evaluated in terms of **reliability** and **timeliness**, as discussed in [22, 28, 29], by checking the metadata in the Web pages related to the schema.org vocabulary or similar ones (Dublin Core, Open Graph Protocol, etc.). A weighting, with an arbitrary maximum value of 10, is assigned to each element according to its relative importance. CHAPE assigns more weighting (2/3rd) to reliability than to timeliness (1/3rd) according to the following:

Reliability

- author: $w_1 = 1$.
- publisher: $w_2 = 1.5$.
- reviewedBy: $w_3 = 1.5$.
- recognizingAuthority $w_4 = 2.5$ (Tot. 6.5).

Timeliness

- dateCreated: $w_5 = 0.5$.
- dateModified: $w_6 = 1.5$.
- datePublished: $w_7 = 0.5$.
- lastReviewed: $w_8 = 1$ (Tot. 3.5).

Thus, given a set of m Web pages, for each Web page CHAPE checks the presence of one or more quality information elements and sums up the related weights. The *information quality index (IQ)* is computed as follows:

$$IQ = \sum_{k=1}^8 w_k \div \frac{\max}{m} \sum_{k=1}^8 w_k \quad (2)$$

Finally, the *ranking index* of the Web page is computed as follows:

$$R = \alpha * LC + (1 - \alpha) * IQ \quad (3)$$

where α allows to differently weigh the language complexity and information quality.

The Web pages with the highest-ranking indexes will appear first.

5 Evaluation of CHAPE

5.1 Subjective Tests

Subjective testing of CHAPE was performed by first-year social work students at the University of Palermo on 29th March 2022. After being invited to use CHAPE, students were asked to fill a short online survey (https://ec.europa.eu/eusurvey/runner/CHAPE_initial_2). The survey had four sections: 1. Non-sensitive user-profile information, 2. Questions on the use of CHAPE, 3. Desired additional features, and 4. Any other feedback. The survey presented statements about CHAPE whose agreement was expressed in a 1–5 Likert-type scale as follows:

1 = *strongly disagree*, 2 = *disagree*, 3 = *undecided*, 4 = *agree*, 5 = *strongly agree*.

Overall, fifteen responses were obtained. The user profiles of the respondents were the following:

- 14 females and 1 male; 14 respondents with an age range 19–40 and 1 respondent with an age range 41–60.
- Medical knowledge: 6 none, 5 basic, 3 average, and 1 good.
- Computer skills: 2 none, 8 basic, 2 average, 1 good, and 2 excellent.

Responses were assessed in terms of usability, user experience, perceived value, and potential users.

Usability: most respondents found CHAPE interface clear, fast, and easy to use without prior knowledge. It helps in identifying health information about complaints, but it is less useful in identifying diseases and medical tests related to a complaint (Table 1).

User Experience: most respondents could better understand their complaints and related diseases but not the medical tests (Table 2). The provided websites were considered trustworthy and the provided information was considered relevant.

Perceived Value: most respondents found that CHAPE would encourage users to take a more active interest in their health and wellbeing but did not think it would improve their health and communication with doctors (Table 3).

Potential Users: most respondents considered patients and/or the general public rather than medical professionals (including the social workers) were the most likely potential users (Table 4).

No specific comments were provided in terms of additional features and other feedbacks.

Table 1. Responses to “How easy to use do you find CHAPE?” question.

Statements	No of respondents (and %) who agree or strongly agree
I find the CHAPE interface clear and easy to understand	13 (86.7%)
CHAPE helps me to identify my problem/complaint	10 (66.7%)
CHAPE helps me to identify diseases related to my complaint	6 (40.0%)
CHAPE helps me to identify tests related to my complaint	4 (26.7%)
CHAPE helps me to identify web information related to my complaint	8 (53.3%)
I think that CHAPE is simple and can be used without prior knowledge	11 (73.3%)
I find that CHAPE is fast in responding to my input	10 (66.7%)

Table 2. Responses to “How helpful do you find CHAPE?” question.

Statements	No of respondents (and %) who agree or strongly agree
After using CHAPE I understand my complaints and diseases better	9 (60.0%)
CHAPE helps me to understand tests related to my complaint	7 (46.7%)
CHAPE helps me to improve my overall comprehension of complaints and diseases	11 (73.3%)
The websites that CHAPE provides me to explore further health information are useful for me	10 (66.7%)
I consider the information provided by CHAPE trustworthy	10 (66.7%)
I consider the information provided by CHAPE relevant to me	8 (53.3%)

Table 3. Responses to “Which of the following statements apply to you?” question.

Statements	No of respondents (and %)
CHAPE encourages me to take a more active interest in my health and wellbeing	12 (80.0%)
CHAPE helps me to improve communication with my doctor	2 (13.3%)
CHAPE helps me to improve my health	3 (20.0%)

Table 4. Responses to “Who should use CHAPE according to you?” question.

Statements	No of respondents (and %)
Patients	7 (46.7%)
Relatives	2 (13.3%)
Visiting health and social workers	2 (13.3%)
General public	11 (73.3%)
Nurses	1 (6.7%)
Doctors	1 (6.7%)

5.2 Objective Tests Based on Language Complexity and Quality Information

To evaluate the ability of CHAPE to provide better-quality information that is easy to understand, we re-ranked Google results according to *language-complexity index* (*LC*), *information-quality index* (*IQ*), and *ranking index* (*R*), by choosing $\alpha = 0.5$ to equally weigh *LC* and *IQ*. Four of the most searched health terms in Google are two complaints: *anxiety* and *depression* and two diseases: *diabetes* and *pneumonia*. For the first ten Google results of these four terms, we computed *LC*, *IQ*, and *R* as shown in Table 5. Table 6 shows the re-ranking of the Google results of CHAPE using the *R* index. CHAPE alters considerably the ranking of all four keywords. To better show this, Table 7 presents the Hamming and Manhattan distances between the original Google ranking and the CHAPE ones. The Hamming distance indicates how many positions in the new ranking differ from the original ones, a value of 10 indicating that all the positions have changed. The Manhattan distance provides a quantitative value of the distances of the new positions with respect to the original, a value of 0 indicates complete agreement while the maximum of 50 indicates the ranking order is completely reverse. Although CHAPE is still under development, it is already able to re-rank health-related Web pages, provided by a generic search engine such as Google, so that information extracted from higher-quality and easier to understand Web pages is shown first.

Table 5. Computation of *LC*, *IQ* and *R* for anxiety, depression, diabetes, and pneumonia keywords.

Original Google ranking	Anxiety			Depression			Diabetes			Pneumonia		
	<i>LC</i>	<i>IQ</i>	<i>R</i>	<i>LC</i>	<i>IQ</i>	<i>R</i>	<i>LC</i>	<i>IQ</i>	<i>R</i>	<i>LC</i>	<i>IQ</i>	<i>R</i>
1	0.86	0.50	0.68	0.88	0.00	0.44	0.71	0.67	0.69	0.75	0.33	0.54
2	0.76	0.33	0.55	0.93	0.00	0.46	0.80	0.33	0.57	0.79	0.33	0.56
3	0.96	0.50	0.73	0.78	1.00	0.89	0.69	1.00	0.84	1.00	0.33	0.67
4	0.79	0.00	0.40	1.00	0.00	0.50	1.00	0.33	0.67	0.99	0.50	0.75
5	1.00	1.00	1.00	0.99	0.67	0.83	0.63	0.33	0.48	0.99	0.50	0.75
6	0.75	0.50	0.63	0.74	1.00	0.87	0.99	1.00	0.99	0.99	1.00	1.00
7	0.38	0.50	0.44	0.98	0.67	0.82	0.82	0.00	0.41	0.99	0.50	0.75
8	0.79	0.33	0.56	0.98	0.00	0.49	0.73	0.67	0.70	0.80	0.50	0.65
9	0.86	0.50	0.68	0.98	0.67	0.82	0.94	1.00	0.97	0.96	0.50	0.73
10	0.91	0.50	0.71	0.97	0.67	0.82	0.75	0.67	0.71	0.96	0.50	0.73

Table 6. Re-ranking of Google results.

Original Google ranking	CHAPE re-ranking			
	<i>Anxiety</i>	<i>Depression</i>	<i>Diabetes</i>	<i>Pneumonia</i>
1	5	10	6	10
2	7	9	8	9
3	2	1	3	7
4	10	7	7	2
5	1	3	9	3
6	6	2	1	1
7	9	4	10	4
8	8	8	5	8
9	4	5	2	5
10	3	6	4	6

Table 7. Hamming and Manhattan distances.

Distance	Anxiety	Depression	Diabetes	Pneumonia
Hamming	8	9	9	10
Manhattan	34	38	42	40

6 Conclusions

Intelligent Empowering Agents (IEA) should enable anyone anywhere, regardless of educational level or health literacy, to have instant access to health information they understand, which empowers them to decide the wisest interventions, if any, for their immediate and future well-being. We have designed and developed an IEA that behaves “intelligently” by allowing the user to input his/her profile and requirements in an easy way. The system provides a customized list of complaints, to choose from, and tailored health information that is of good-quality and easy understood. To our best knowledge, this is the first attempt to create an intelligent empowering agent that exploits the potential of AI and the vast amount of health information available on the Web to facilitate comprehension and action on general complaints/diseases.

Future work aims to redesign the user interface to be more conversational (chatbot like) and include more user profile information, such as gait, body type, nutritional status, comorbidities etc. Complaints and diseases will be associated with Concept Unique Identifiers (CUI) of the Unified Medical Language System™ to map them to standard terms taken from medical-term classifications such as ICD-9, ICD-10, or SNOMED. AI will be used not only to filter information gathered from the Web but also to process previously collected user information. Although user input is anonymous, users will be

provided with an option to grant or withdraw informed consent to use their data. Once these improvements are made the agent is going to be tested on a wider demographic.

Acknowledgements. This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094_P2 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Science Foundation Ireland Research Centre for Software (www.lero.ie).

References

1. Snowdon, A.: Digital Health: A Framework for Healthcare Transformation. HIMSS, Chicago (2020). <https://www.himss.org/resources/digital-health-framework-healthcare-transformation-white-paper>
2. European Health Parliament: Patient empowerment and centredness (2017). <https://www.healthparliament.eu/patient-empowerment-centredness/#:~:text=There%20is%20a%20wide%20spread%20consensus,expanding%20burden%20of%20chronic%20diseases>
3. World Health Organization: Framework on integrated, people-centred health services: report by the Secretariat. World Health Assembly (A69/39), April 2016, pp. 1–12 (2016). https://apps.who.int/gb/ebwha/pdf_files/WHA69/A69_39-en.pdf
4. Alfano, M., Lenzitti, B., Taibi, D., Helfert, M.: Provision of tailored health information for patient empowerment: an initial study. In Proceedings of the 20th International Conference on Computer Systems and Technologies (CompSysTech 2019). Association for Computing Machinery, New York, pp. 213–220 (2019). <https://doi.org/10.1145/3345252.3345301>
5. Bodolica, V., Spraggon, M.: Toward patient-centered care and inclusive health-care governance: a review of patient empowerment in the UAE. *Public Health* **169**(971), 114–124 (2019)
6. Cerezo, P.G., Juvé-Udina, M.E., Delgado-Hito, P.: Concepts and measures of patient empowerment: a comprehensive review. *Revista Da Escola de Enfermagem* (2016)
7. Fumagalli, L.P., Radaelli, G., Lettieri, E., Bertele, P., Masella, C.: Patient empowerment and its neighbours: clarifying the boundaries and their mutual relationships. *Health Policy* **119**(3), 384–394 (2015)
8. Finney Rutten, L.J., Blake, K.D., Greenberg-Worisek, A.J., Allen, S.V., Moser, R.P., Hesse, B.W.: Online health information seeking among US adults: measuring progress toward a healthy people 2020 objective. *Public Health Rep.* **134**(6), 617–625 (2019)
9. Alfano, M., Lenzitti, B., Taibi, D., Helfert, M.: Language complexity in on-line health information retrieval. In: Zieffle, M., Maciaszek, L.A. (eds.) *ICT4AWE 2019*. CCIS, vol. 1219, pp. 79–100. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52677-1_5
10. Kondylakis, H., et al.: Smart recommendation services in support of patient empowerment and personalized medicine. In: Tsihrintzis, G., Virvou, M., Jain, L. (eds.) *Multimedia Services in Intelligent Environments*. Smart Innovation, Systems and Technologies, vol. 25, pp. 39–61. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-319-00375-7_4
11. Iatraki, G., et al.: Personal health information recommender: implementing a tool for the empowerment of cancer patients. *Ecancermedalscience* **12**, 1–11 (2018)
12. Davenport, T., Kalakota, R.: The potential for artificial intelligence in healthcare. *Future Healthc. J.* **6**(2), 94–98 (2019). <https://doi.org/10.7861/futurehosp.6-2-94>

13. Fast, E., Horvitz, E.: Long-term trends in the public perception of artificial intelligence. In: Proceedings of 31st AAAI Conference on Artificial Intelligence (AAAI 2017), pp. 963–969. (2017)
14. Jiang, F., Jiang, Y., Zhi, H., et al.: Artificial intelligence in healthcare: past, present and future. *Stroke Vasc. Neurol.* **2**, e000101 (2017). <https://doi.org/10.1136/svn-2017-000101>
15. Alfano, M., Kellett, J., Lenzitti, B., Helfert, M.: Proposed use of a conversational agent for patient empowerment. In: Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies, vol. 4, pp. 817–824 (2021). Scale-IT-up, ISBN: 978-989-758-490-9. <https://doi.org/10.5220/0010414408170824>
16. Banna, S., Hasan, H., Dawson, P.: Understanding the diversity of user requirements for interactive online health services. *Int. J. Healthc. Technol. Manag.* **15**(3), 253–271 (2016)
17. Kamel Ghalibaf, A., Nazari, E., Gholian-Aval, M., et al.: Comprehensive overview of computer-based health information tailoring: a systematic scoping review. *BMJ Open* **9**, e021022 (2019)
18. Bol, N., Smit, E.S., Lustria, M.L.A.: Tailored health communication: opportunities and challenges in the digital era. *Digit. Health* **6**, 1–3 (2020)
19. Cheung, K.L., Durusu, D., Sui, X., de Vries, H.: How recommender systems could support and enhance computer-tailored digital health programs: a scoping review. *Digit. Health* **5**, 1–19 (2019)
20. Kreuter, M., Farrell, D., Olevitch, L., Brennan, L.: Tailoring health messages: customizing communication with computer technology. Lawrence Erlbaum Associates, Mahwah, New Jersey (2000). <https://www.popline.org/node/174671>
21. Petty, R.T., Cacioppo, J.T.: Attitudes and Persuasion: Classic and Contemporary Approaches. Wm C. Brown, Dubuque, IA (1981)
22. Alfano, M., Lenzitti, B., Taibi, D., Helfert, M.: On-line retrieval of health information based on language complexity, information customization and information quality. In: Ziefle, M., Guldmond, N., Maciaszek, L.A. (eds.) ICT4AWE 2020. CCIS, vol. 1387, pp. 1–20. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-70807-8_1
23. MacBryde, C.M., Blacklow, R.S. (eds.): Signs and Symptoms: Applied Pathologic Physiology and Clinical Interpretation, 5th edn. Lippincott, Philadelphia (1970)
24. Kellett, J., Deane, B.: The diagnoses and co-morbidity encountered in the hospital practice of acute internal medicine. *Eur. J. Intern. Med.* **18**(6), 467–473 (2007)
25. Finley, C.R., et al.: What are the most common conditions in primary care? Systematic review. *Can. Fam. Physician* **64**, 832–840 (2018)
26. Kloehn, N., et al.: Improving consumer understanding of medical text: development and validation of a new subsimplify algorithm to automatically generate term explanations in English and Spanish. *J. Med. Internet Res.* **20**(8), e10779 (2018)
27. Leroy, G., et al.: Improving perceived and actual text difficulty for health information consumers using semi-automated methods. In: AMIA Annual Symposium Proceedings, pp. 522–531. (2012)
28. World Health Organization (WHO): Improving data quality: a guide for developing countries, pp. 1–74 (2003)
29. Eysenbach, G., Powell, J., Kuss, O., Sa, E.R.: Empirical studies assessing the quality of health information for consumers on the world wide web: a systematic review. *JAMA J. Am. Med. Assoc.* **287**(20), 2691–2700 (2002)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Comparison and Analysis of 3 Key AI Documents: EU's Proposed AI Act, Assessment List for Trustworthy AI (ALTAI), and ISO/IEC 42001 AI Management System

Delaram Golpayegani^(✉), Harshvardhan J. Pandit, and Dave Lewis

ADAPT Centre, School of Computer Science and Statistics, Trinity College Dublin,
Dublin, Ireland
{sgolpays,pandith,delewis}@tcd.ie

Abstract. Conforming to multiple and sometimes conflicting guidelines, standards, and legislations regarding development, deployment, and governance of AI is a serious challenge for organisations. While the AI standards and regulations are both in early stages of development, it is prudent to avoid a highly-fragmented landscape and market confusion by finding out the gaps and resolving the potential conflicts. This paper provides an initial comparison of ISO/IEC 42001 AI management system standard with the EU trustworthy AI assessment list (ALTAI) and the proposed AI Act using an upper-level ontology for semantic interoperability between trustworthy AI documents with a focus on activities. The comparison is provided as an RDF resource graph to enable further enhancement and reuse in an extensible and interoperable manner.

Keywords: Trustworthy AI · AI management system · ALTAI · AI Act · ISO/IEC 42001 · Ontology · Activity · Comparison

1 Introduction

The wide application of AI systems urges governments, legislators, standardisation bodies, and think tanks to encourage and sometimes obligate organisations to develop and use AI in a trustworthy manner. AI regulations, standards, and

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813497, as part of the ADAPT SFI Centre for Digital Media Technology is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant#13/RC/2106.P2. Harshvardhan J. Pandit has received funding under the Irish Research Council Government of Ireland Postdoctoral Fellowship Grant#GOIPD/2020/790.

© The Author(s) 2023

L. Longo and R. O'Reilly (Eds.): AICS 2022, CCIS 1662, pp. 189–200, 2023.

https://doi.org/10.1007/978-3-031-26438-2_15

guidelines developed separately and in isolation risk a highly fragmented landscape that can lead to regulatory and market confusion. Consequently, organisations are compelled to navigate a large number of competing and changing requirements from multiple sources regarding AI development and use. The lack of alignment between different sources of requirements, such as laws and standards, creates difficulties in identifying and fulfilling obligations.

In this paper, we identify the commonality, inconsistencies, and gaps across the following three dominant AI documents within the scope of EU’s regulatory regime: the proposed AI Act [1], Assessment List for Trustworthy AI (ALTAI) [2], and the draft ISO/IEC 42001 standard for AI management systems¹.

Amongst these three, we utilise ISO/IEC 42001 as the primary source of requirements given its distinct role as a certifiable standard, and compare the others with it to indicate adherence towards guidelines (ALTAI) and regulations (AI Act). More specifically, we investigate the following questions:

- (i) To what extent can ALTAI’s trustworthy AI requirements be integrated into ISO/IEC 42001’s AI management system activities?
- (ii) To what extent can AI Act’s high-risk AI obligations be integrated into ISO/IEC 42001’s AI management system activities?

We address the aforementioned questions by proposing a methodology to compare AI documents using an upper-level trustworthy AI ontology [3], which enables modelling and linking concepts within AI documents (see Sect. 2). We then demonstrate the comparison of ISO/IEC 42001 with ALTAI’s trustworthy AI (Sect. 3) and the AI Act (Sect. 4). The comparison is made available online as an RDF resource to enable further enhancement and reuse². We discuss semantic modelling of activities extracted from the documents in Sect. 5. In Sect. 6, related work on ontology-based comparison of policies, regulations, and standards is mentioned and we conclude the paper and identify avenues for future work in Sect. 7.

2 Methodology for Comparison and Analysis

AI documents can be compared on the basis of different semantic building blocks: key terms defined within them, activities mentioned, and normative requirements or obligations required to be met for compliance. Considering the central focus of management system standards on organisational activities and processes, we limit the scope of our comparison to activities.

Given that different standards, regulations, and policies are being created for evaluating trustworthiness of AI, there is bound to be some overlap between them. To assist in the task of comparing them, a conceptual model and framework is essential to identify and link together the relevant concepts within different documents. An ontological representation permits formalisation of the conceptual model and its application in use-cases. With this view, Fig. 1 presents

¹ <https://www.iso.org/standard/81230.html>.

² <https://github.com/delaramglp/aidocs>.

the core ontology for supporting mapping of concepts between different emerging AI standards. It is based on activities carried out within ISO/IEC (more specifically sub-committee 42) regarding AI standardisation and incorporates existing ISO/IEC standards and outputs for ‘characteristics’ expressed by trustworthy AI systems.

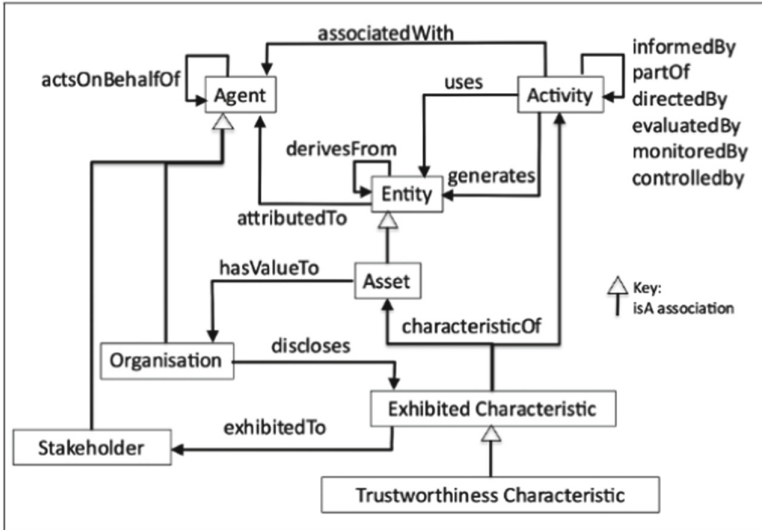


Fig. 1. Upper-level ontology for mapping trustworthy AI documents [3]

The premise of the ontology rests on the fact that several of trustworthy characteristics are yet to be clarified and defined in relation to AI and AI development activities. Therefore, it focuses on specifying the relationships between activities, entities, assets, and characteristics (exhibited for trustworthiness), agents, stakeholders, and organisations. The ontology is based on Basic Formal Ontology (BFO) - a generic upper-level ontology used in formalisations across domains, and the PROV-O ontology which is a W3C standard for expressing provenance.

The ontology provides a way to express activities of organisations that relate to AI where the trustworthiness is manifested through characteristics of Entities that make up a product or service employing AI. It also provides a way to depict the influence of entities, activities, and agents in these processes, and captures the role of stakeholders in disclosing and exhibiting trustworthiness of AI through its characteristics. The ontology thus enables representing use of AI from both within and outside the perspective of an organisation or service, and is useful for comparing different AI guidelines by using its conceptual model as a framework for identifying and aligning concepts.

We utilise the trustworthy AI ontology to compare AI documents in order to assess the degree of alignment between them by modelling and linking trustworthy AI activities mentioned within them. The following describes the steps taken for analysis and comparison of documents:

1. The documents are analysed to extract relevant activities to trustworthy AI, which then modelled as **Activity**.
2. **partOf** relationship is used to bridge the isolated sets of **Activities** identified from the documents.
3. An analysis is carried out to identify the overlaps and potential conflicts through investigation of activities that are mapped or could not be mapped using the **partOf** relation.

3 Comparison of ALTAI with ISO/IEC 42001

3.1 ALTAI Activities

ALTAI suggests a set of questions, grouped by the ethical principle under assessment, for assessing whether an AI system adheres to trustworthy AI requirements specified in [4] (see the structure of ALTAI in Fig. 2). Designed for trustworthy AI self-assessment, ALTAI provides useful hints regarding development and use of AI systems. One of the aspects of trustworthiness assessment is execution of particular activities; for example, ‘Are end-users or other subjects adequately made aware that a decision, content, advice or outcome is the result of an algorithmic decision?’, which is a question listed under Human Agency and Oversight requirements, implies execution of an activity to *inform end-users or other subjects that a decision, content, advice or outcome is the result of an algorithmic decision*. For the purpose of comparison, we made the management activities implied by ALTAI questions explicit.

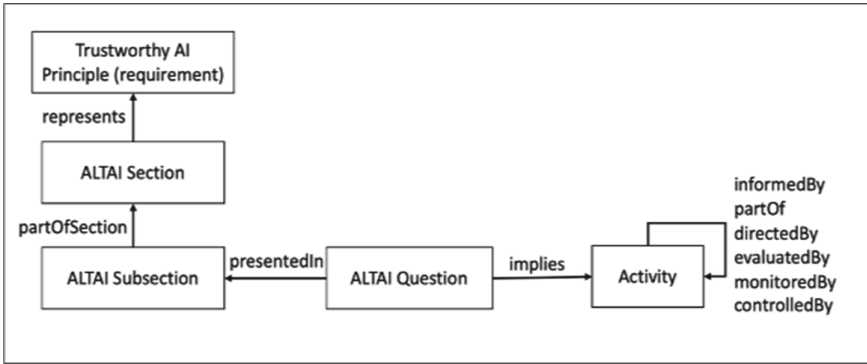


Fig. 2. ALTAI structure

3.2 AI Management System Activities

The ISO/IEC 42001 standard for AI management systems, being developed by JTC 1/SC 42, is currently (Nov’22) in DIS or draft stage, implying relative maturity awaiting final comments before publication. It follows the ‘harmonised structure’ of all management system standards developed by ISO, which is defined in

the openly available ISO/IEC Directives on procedures for ISO technical work³. Based on the harmonised structure, Lewis et al. [3] identified AI management system activities, where each is given an identifier, a label, and a ‘see also’ attribute which is a link to the relevant harmonised structure clause. The entities generated and used by each activity are represented in a similar manner. The updated list of AI management system activities, which reflects the latest version of the Directive published in 2022, is presented in Table 1.

Table 1. AI management system (AIMS) activities

o No	ID	AIMS activity (label)	HS clause (see also)
1	UOC	Understanding organisation and its context	4.1
2	USE	Understanding stakeholder needs and expectation	4.2
3	DS	Determine AIMS scope	4.3
4	EIMI	Establish, implement, maintain and continually improve management system and its processes	4.4
5	DLC	Demonstrate leadership and commitment to the management system	5.1
6	EP	Establish AIMS policy	5.2
7	ARRA	Assign roles, responsibilities and authorities	5.3
8	ARO	Address risks and opportunities	6.1
9	EPAO	Establish and plan to achieve AI objectives	6.2
10	ARRA	Assign roles, responsibilities and authorities	6.3
11	DAR	Determine and allocate resources for AIMS	7.1
12	DEC	Determine and ensure competence of people affecting AI performance	7.2
13	PA	Promote awareness	7.3
14	DC	Determine AIMS communication	7.4
15	CUCD	Create, update, and control documented information	7.5
16	PCP	Plan and control AI processes	8.1
17	MMAE	Monitor, measure, analyse and evaluate AI	9.1
18	IA	Internal (AIMS) audit	9.2
19	UMR	Undertake management review	9.3
20	DNCA	Detect non-conformance and take corrective action	10.1
21	CI	AIMS Continual improvement	10.2

3.3 ALTAI - ISO/IEC 42001 Activity Comparison

By comparing ALTAI with ISO/IEC 42001, we aim to investigate the following:

- Is there any organisational activity required for trustworthy AI that cannot be integrated into an AI management system?

³ <https://www.iso.org/sites/directives/current/consolidated/index.xhtml>.

- Which AI management systems activities do not play a role in achieving trustworthiness?
- What management systems activities are involved in achieving a particular trustworthy AI requirement, e.g. privacy and data governance?

Alignment Groups. In the comparison process, a number of commonly occurring structures are identified. For instance, multiple ALTAI activities that refer to achieving AI objectives such as *Accuracy*, *Explainability*, *Privacy*, and *Fairness* are **partOf** ‘establish and plan to achieve AI objectives’ activity. We categorise these structures into the 17 alignment groups listed in Table 2.

Table 2. ALTAI - AI management system activities alignment groups

ID	ALTAI activity structure	partOf (AIMS activity)
AG1	Assess the impact of the AI system	ARO
AG2	Assess the system vulnerabilities or threats	ARO
AG3	Assess whether the AI system respects a specific right	ARO
AG4	Establish processes to test or monitor AI impacts or risks	PCP & ARO & MMAE
AG5	Establish processes to measure and assess AI risks	PCP & ARO
AG6	Establish processes to mitigate, rectify, or avoid AI risks	PCP & ARO
AG7	Establish processes to achieve an AI objective	PCP & EPAO
AG8	Assess whether an AI objective is achieved	EPAO & MMAE
AG9	Establish processes to test and monitor AI objectives	PCP & EPAO & MMAE
AG10	Establish processes to measure and assess AI objectives	PCP & EPAO & MMAE
AG11	Provide information about a design decision	UOC
AG12	Determine compliance/Align the systems with a specific standard or guideline	PCP & UOC
AG13	Designate a role	ARRA
AG14	Establish a broad (e.g. ethics review board)	ARRA
AG15	Provide employee training/Ensure workers competence	DEC
AG16	Communicate with or inform users or third parties	DC
AG17	Inform staff and employees about the AI policy	PA

Insights. The comparison revealed that ALTAI is centred around trustworthy AI issues and principles rather than how to manage trustworthy AI processes and policies within an organisation. In comparison, the draft AI management system standard does not specifically refer to any trustworthy principle, however,

it provides a foundation for implementing these principles in an organisation. The two are therefore complementary regarding effective implementation and assessment of trustworthy AI, with the comparison providing a way to achieve trustworthiness through management system activities.

Table 3 presents the number of ALTAI activities that are mapped into each AI management system activity. It should be noted that the total number indicates the number of times an AI management system activity is individually mapped to ALTAI activities as the mapping between the two is many-to-many. Activities within AI management system that do not have a corresponding ALTAI activity are omitted from the table (8 in total).

As shown in the table, approximately 50% (73 of 144) of ALTAI activities refer to risk management which makes the fact that ALTAI adopts a risk-oriented approach towards trustworthy AI clear. The missing management system activities in the table, which are nearly half of total, demonstrates that processes and tasks at a high level of organisational governance and management are not covered in ALTAI.

Table 3. Number of ALTAI activities mapped into each AIMS activity

AIMS activity	AIMS activity (label)	Nos. ALTAI activities
ARO	Address risks and opportunities	73
PCP	Plan and control AI processes	54
EPAO	Establish and plan to achieve AI objectives	44
DC	Determine AIMS communication	22
MMAE	Monitor, measure, analyse and evaluate AI	20
UOC	Understanding organisation and its context	12
DEC	Determine and ensure competence of people affecting AI performance	7
ARRA	Assign roles, responsibilities and authorities	2
PA	Promote awareness	2

4 Comparison of AI Act with ISO/IEC 42001

4.1 The AI Act Activities

In April 2021, the European Commission published the proposal for EU AI regulation, called AI Act, to create a legal framework for trustworthy AI by laying down obligations which are proportionate to the level of risk imposed by AI systems. Under the AI Act, providers of high-risk AI systems, i.e. systems that are likely to cause harm to health, safety, and rights of individuals, are required to implement a quality management system (Art. 17), among other requirements.

The AI Act relies on creation of harmonised AI standards to facilitate conformity to its requirements by providing technical solutions (Art. 40).

Conformity with the AI Act’s high-risk AI obligations requires performing organisational as well as technical activities. By analysis of the requirements for high-risk AI systems and the obligations of providers of those systems, described in title III, Chaps.2 and 3, we identified 52 high-level organisational activities that are **associatedWith** high-risk AI providers, which are modelled as **Agents**. It is important to note that our list of activities is not exhaustive, and therefore performing the identified activities is essential for conformity to the AI Act but not necessarily sufficient.

4.2 AI Act - ISO/IEC 42001 Activity Comparison

Using the methodology described earlier, we mapped the activities identified from the AI Act to the ones extracted from ISO/IEC 42001. Table 4 shows mapping of AI Act’s risk management activities into AI management system.

Table 4. Comparison of AI Act’s risk management activities with AIMS

AI Act risk management activity	partOf (AIMS)
Establish risk management system	DC & EIMI & ARO
Implement risk management system	EIMI & ARO
Document risk management system	EIMI & ARO & CUCD
Maintain risk management system	EIMI & ARO
Identify/Analyse/Evaluate/Mitigate Risks	ARO
Communicate Residual Risk to Users	PA & AIRO
Identify Impact On Stakeholders (e.g. children)	USNE & ARO

Insights. Our analysis indicates activities to establish management systems, address risks, create documentation, and communicate with external entities are among the most mapped management system activities. This shows that in conformity to the AI Act’s legal requirements, documentation and sharing information with external stakeholders are as important as conducting risk management.

Identification of the degree to which compliance to ISO/IEC 42001 assists in conformity to AI Act’s high-risk AI obligations needs further investigation as our focus was primarily on the organisational activities explicitly referenced therein.

5 Semantic Modelling of Activities

Documents that specify guidelines generally refer to activities and processes across three distinct phases: ex-ante where a plan of activity must exist; ongoing

or during where an activity is currently in the process of being executed; and ex-post where an activity has finished execution or has produced artefacts. For AI guidelines, it is important to model the corresponding semantic representation of activities in a similar manner so as to distinguish when an organisation or system must have a plan in place representing some *future activity* versus having carried out that activity i.e. *in the past*. This notion is also applicable and demonstrated in the area of legal and regulatory compliance where an obligation can entail provenance of both a plan as well as executed activities, and therefore requires documentation at both ex-ante and ex-post phases [5].

Intended for self-assessment purposes, ALTAI predominately refers to the ex-post phase. This means that to provide answers to ALTAI questions we have to look into the results and artefacts of executed activities. Furthermore, separation between ex-ante and ex-post phases of ALTAI activities enables ex-ante planning for trustworthiness and ex-post trustworthy AI (self-) assessment as outlined by AI management system activities. However, for semantic representation of the activities extracted from ALTAI both planning and execution phases should be taken into account. For example, from ‘establish processes to assess AI risks’ two activities are inferred: plan for AI risk assessment (ex-ante) and AI risk assessment (ex-post). A semantic model of the former should be able to represent plans for risk assessment, intended steps and actions, responsible parties, and entities generated and used during the planning. This can be done by extending the Ontology for Provenance and Plans (P-Plan)⁴. Naja et al. [6] have adopted the same approach for recording accountability plans. Representing ex-post activities is possible by extending the PROV-O ontology.

To model previously introduced alignment groups we consider the ex-post phase. Each alignment group can be represented as an ontology design pattern (ODP) [7]. An example of one such pattern for AG17 (providing training for employees to ensure competence) that uses the PROV-O ontology to represent agents⁵ and activities is shown in Fig. 3. By modelling training activities using this pattern all processes and activities which are part of DEC (Determine and ensure competence of people affecting AI performance) can be uniformly represented, and retrieved e.g. using SPARQL queries.

Using the pattern as a generic template for different activities and roles regarding training enables a uniform mechanism to answer questions such as:

- Did the organisation provide training to staff on risk management?
- Who provided the training? When? To whom? On what topic?
- What activities are relevant to training?
- What are the subjects that the organisation provides training on?
- Who is trained on a specific topic, e.g. risk management?

⁴ <https://www.opmw.org/model/p-plan/>.

⁵ The PROV concepts of agents and entities are different from ALTAI and AIMS. In PROV, an entity is an artefact such as an input to an activity, and an agent is what is referred to as an entity within ALTAI, AIMS, and the general use of the words.

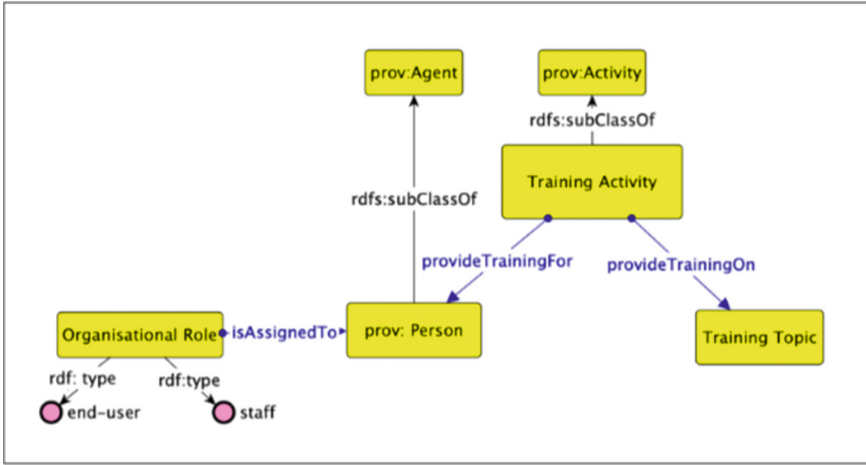


Fig. 3. Training activity pattern.

6 Related Work

Boer et al. [8] used an ontology-based approach to facilitate comparison of similar regulations, i.e. in a specific area such as tax, within different jurisdictions. Despres and Szulman [9] proposed an approach for integrating ontologies created from the European community directives. Fiorentini et al. [10] proposed an approach for harmonisation which compares documents using informal analysis, typology of standards, use-cases, and ontologies. Pardo et al. [11] created H2mO - an ontology for harmonisation of reference models and standards utilised in software process improvement. Koelle et al. [12] proposed a tool for ATM security which harmonises relevant standards and regulations. Lewis et al. [3] presented an analysis of the normative content of trustworthy AI guidelines presented by IEEE, EU HLEG, and OECD and mapped these guidelines into ISO 26000 social responsibility issues.

7 Conclusion

This paper presented a comparison and analysis between the EU AI Act, ALTAI, and ISO/IEC AI management system standard to identify the potential alignment between these 3 key documents. The assessment compared management-level activities mentioned in the documents and is represented formally using the trustworthy AI upper-level ontology proposed by [3].

Implications of Comparison and Analysis of AI Documents. Identification of the gaps existed in the AI documents being developed assists standardisation bodies in determining the areas that need creation or modification

of standards. Legislators can use the comparison to determine the degree to which compliance with existing AI standards contributes to conformity to legal obligations and identify the aspects of trustworthy AI that are not subject to regulation. Furthermore, comparison of activities provides a baseline for the communications between authorities and standardisation bodies for development of harmonised regulations and standards.

The comparison assists AI providers and developers in adoption of standards and guidelines required for satisfying legal requirements by helping them identify inconsistencies and areas of overlaps. It can also be used to ensure organisational AI policies are effective in satisfying normative and legal requirements.

Given the potential of AI research to cause harm, recently some AI conferences, such as NeurIPS⁶, provide ethical guidelines and ask researchers to assess the impact of their work on key areas of concern, e.g. safety, fairness, and privacy. The comparison methodology can be applied in assessing the alignment of ethical guidelines provided by different conferences, universities' policies on ethics and data protection as well as ethical assessment approaches.

Further Work. The comparison presented in this paper will be expanded to provide a more comprehensive analysis and alignment of key terms, technical activities, and requirements detailed within AI documents. Starting with the analysis provided in this paper, we aim to identify a common set of AI risk and impact assessment activities from the AI Act, ALTAI, and ISO risk management and management system standards and extend AIRO - an ontology for describing AI risks [13], to represent provenance of activities. Future work also includes updating this work based on changes made in the subsequent drafts and finalisations of the AI Act and ISO/IEC 42001 standard.

References

1. Artificial intelligence act: Proposal for a regulation of the European parliament and the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts (2021). <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELLAR:e0649735-a372-11eb-9585-01aa75ed71a1>
2. European Commission, Content Directorate-General for Communications Networks, and Technology. The Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self assessment. Publications Office (2020). <https://doi.org/10.2759/002360>. <https://data.europa.eu/doi/10.2759/002360>
3. Lewis, D., Filip, D., Pandit, H.J.: An ontology for standardising trustworthy AI. In: Factoring Ethics in Technology, Policy Making, Regulation and AI, Chap. 5 (2021). <https://doi.org/10.5772/intechopen.97478>
4. European Commission and Directorate-General for Communications Networks, Content and Technology. Ethics guidelines for trustworthy AI. Publications Office (2019). <https://doi.org/10.2759/346720>. <https://data.europa.eu/doi/10.2759/346720>

⁶ NeurIPS 2022 ethics guidelines <https://neurips.cc/public/EthicsGuidelines>.

5. Pandit, H.J., O'Sullivan, D., Lewis, D.: Test-driven approach towards GDPR compliance. In: Acosta, M., Cudré-Mauroux, P., Maleshkova, M., Pellegrini, T., Sack, H., Sure-Vetter, Y. (eds.) SEMANTiCS 2019. LNCS, vol. 11702, pp. 19–33. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33220-4_2
6. Naja, I., Markovic, M., Edwards, P., Cottrill, C.: A semantic framework to support AI system accountability and audit. In: Verborgh, R., et al. (eds.) ESWC 2021. LNCS, vol. 12731, pp. 160–176. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77385-4_10
7. Gangemi, A.: Ontology design patterns for semantic web content. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 262–276. Springer, Heidelberg (2005). https://doi.org/10.1007/11574620_21
8. Boer, A., van Engers, T., Winkels, R.: Using ontologies for comparing and harmonizing legislation. In: Proceedings of the 9th International Conference on Artificial Intelligence and Law, pp. 60–69 (2003)
9. Despres, S., Szulman, S.: Merging of legal micro-ontologies from European directives. *Artif. Intell. Law* **15**(2), 187–200 (2007)
10. Fiorentini, X., et al.: Towards a method for harmonizing information standards. In: 2009 IEEE International Conference on Automation Science and Engineering, pp. 466–471. IEEE (2009)
11. Pardo, C., et al.: An ontology for the harmonization of multiple standards and models. *Comput. Stand. Interfaces* **34**(1), 48–59 (2012)
12. Koelle, R., Strijland, W., Roels, S.: Towards harmonising the legislative, regulatory, and standards-based framework for ATM security: developing a software support tool. In: 2013 International Conference on Availability, Reliability and Security, pp. 787–793. IEEE (2013)
13. Golpayegani, D., Pandit, H.J., Lewis, D.: AIRO: an ontology for representing AI risks based on the proposed EU AI act and ISO risk management standards, pp. 51–65. IOS Press (2022)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





AI and ML in School Level Computing Education: Who, What and Where?

Joyce Mahon^(✉), Brett A. Becker, and Brian Mac Namee

University College Dublin, Dublin, Ireland

joyce.mahon1@ucdconnect.ie, {brett.becker,brian.macnamee}@ucd.ie

Abstract. This paper presents the results of a systematic review of the literature relating to artificial intelligence (AI) and machine learning (ML) education at school level. We conducted a search of the ACM Full-text Collection and 33 papers from the 197 search results were selected for analysis. In this context, we considered the research questions: 1) Who has been the focus of the research?, 2) What course content appears in the research?, and 3) Where has the research taken place? We find that there has been a recent marked increase in research on AI/ML for school level education, although most of this has been based in the United States. The majority of this research focuses on students, with very little specifically addressing teachers, experts, parents, or the wider school community. There is also a lack of attention paid to research focused on women or those from historically underrepresented groups and equity of access to AI/ML courses for school-level students. Finally, the content covered in the courses described in this research varies widely, possibly because there is so little alignment to computer science (CS) frameworks or curricula.

Keywords: AI · Artificial intelligence · Computer science · Education · Informatics · K-12 · Machine learning · Primary · Secondary · School

1 Introduction

Computer science (CS) education in primary and secondary schools (commonly referred to as K-12¹ in the education literature) has become increasingly prevalent in recent years, with more countries adopting, developing, and implementing standards of practice. Artificial intelligence (AI) and Machine learning (ML) are essential additions to K-12 computing education, as students are increasingly interacting with these technologies in their daily lives. For instance, the Irish

¹ The term K-12 comes from “kindergarten through 12th grade” in North American school systems.

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183 and Huawei Ireland.

© The Author(s) 2023

L. Longo and R. O'Reilly (Eds.): AICS 2022, CCIS 1662, pp. 201–213, 2023.

https://doi.org/10.1007/978-3-031-26438-2_16

Leaving Certificate Computer Science mentions both artificial intelligence and machine learning, albeit in a single learning outcome: *explain when and what machine learning and AI algorithms might be used in certain contexts*².

While there is considerable literature on general computing education [14] such as programming at the K-12 level, research on AI/ML in K-12 education is much less common. Although “AI education has begun to progressively trickle down to the K-12 range” [43], research on AI/ML in K-12 education is still in its infancy [35].

This paper describes a systematic review of the existing literature on AI/ML school-level education. The aim of this review is to broadly analyse the research that has been conducted in K-12 computing education that relates to AI/ML. Specifically, we are interested in the following questions:

RQ1: Who has been the focus of the research on AI and ML education at school level?

RQ2: What course content appears in this research?

RQ3: Where has this research taken place?

In answering these questions we also make several important observations with implications for teaching AI/ML to school students.

2 Method

Selecting search terms for a broad review of this literature proved challenging. Terms that are too general result in an unwieldy set of papers (a very large percentage of which are irrelevant), while terms that are too specific are likely to miss relevant papers. After some trial and error with a range of databases, we selected a combined search string that captures the area of interest: (*artificial intelligence* OR *machine learning*) AND (*school* OR *k-12*). To check this search string for appropriateness, we inspected the results for relevance to our research questions. This was in addition to the overall content matter, as well as adequate coverage of a small test-set of papers that together served as a model of the area we intended to survey. The search terms were then applied to the title and abstract of the ACM Full-text Collection. The search was conducted on the 10th August 2022, and identified 197 papers.

We selected the ACM Digital Library as our initial database because it contains the most computer science (or computing) education papers from any one publisher by a significant margin. A search of DBLP³ for *computer science education* returns 1,249 matches. 614 matches (40%) are from the top 10 venues, of which 6 are ACM venues accounting for 74% of the matches from the top 10 venues. Searching for *computing education* returns 1,328 matches. 431 (32%) are from the top 10 venues, of which 9 are ACM venues accounting for 93% of the matches from the top 10 venues.

² www.curriculumonline.ie/Senior-Cycle/Senior-Cycle-Subjects/Computer-Science/.

³ www.dblp.org.

Figure 1 shows the number of papers included in the originally retrieved set for each year since 1973. A marked increase in the appearance of the terms *artificial intelligence* and *machine learning* alongside the terms *K-12* and *school* in the ACM Full-text Collection in recent years is clearly shown. Only 10.7% of the 197 search results are from the 35 year period from 1973 to 2007, with the rest from the almost 15 year period from 2008 to August 2022. Over one quarter of the search results are from the year 2021.

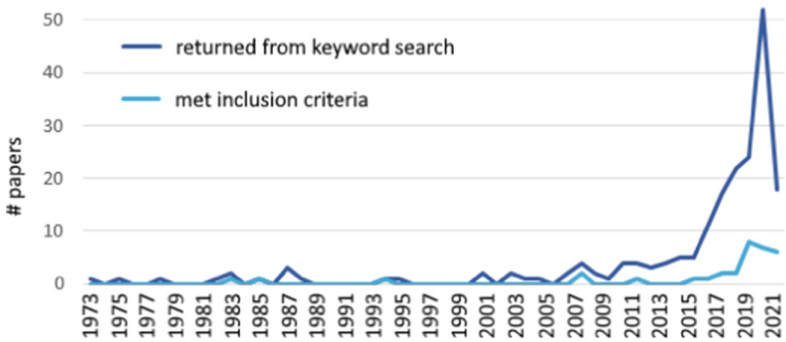


Fig. 1. Search results from the ACM Full-text Collection, Aug 10, 2022. 197 results were returned from a keyword search and 33 papers met the inclusion criteria.

All 197 results were written in English, and only 33 papers (16.75%) met the inclusion criteria for this review - they expressly related to the teaching of AI and/or ML in K-12 education (see Table 1). 62 papers were discounted (31.47%), as they were less than 3 pages in length. There was some interesting preliminary work in some of these shorter papers [17,28,29,38] but they were all without findings. The remaining 102 results (51.78%) were discussed in the context of unrelated fields, or included older students.

Table 1. The 33 papers of the 197 search results that met the inclusion criteria.

Year	N		Year	N		Year	N	
1983	1	[16]	2012	1	[31]	2019	3	[20,30,45]
1985	1	[23]	2016	1	[40]	2020	7	[6,10,19,21,27,41,42]
1994	1	[33]	2017	1	[32]	2021	7	[7,12,15,18,26,34,44]
2008	2	[4,9]	2018	2	[2,13]	2022	6	[5,11,22,24,25,36]

3 Results

3.1 RQ1: Who Has Been the Focus of the Research?

Focus on Students: The majority of the papers focus upon students’ teaching and learning (30 of 33 papers - approx. 91%), although 7 of these papers also address the role of the teacher - see Table 2. The age ranges of students vary widely - for example, one paper describes teaching basic ML concepts (image recognition, supervised learning, training data, model, feature, classifying, accuracy) to children aged from 5 years and older [36]. Another describes how ML fundamentals can be taught to children from 10–16 years old through hands-on activities on an educational web platform [26].

Table 2. The research focus (students only, teachers only, or students & teachers) of the studies that are included in the review.

Focus of research	N	
Students only	23	[2, 4, 6, 7, 9–12, 15, 16, 20, 23, 30–34, 36, 40–42, 44, 45]
Students & teachers	7	[5, 13, 19, 21, 24, 26, 27]
Teachers only	3	[18, 22, 25]

Number of Student Participants: Participant numbers also varied widely (see Table 3). For example, in Weng et al. [42] two elementary school math teachers, a parent and a primary school student, were involved in the development of robot maths quiz games.

In contrast, a 2012 paper [31] sets out a software engineering curriculum for students that includes subjects in AI and ML. Students were given a choice between a 3-unit or a 5-unit programme in their school, which held over 1,000 students aged 16–18 years.

Focus on Teachers: Three papers had a specific focus on teachers—see Table 2. Polak et al. [25] detailed an initial qualitative study with 14 teachers, school psychologists, and education managers from schools in four European countries. A survey targeting a larger group of European teachers was also designed to collect teachers needs and expectations, to create a supportive online educational platform that aids teachers in AI education. The second study [22] introduced a new instrument to measure teachers’ trust in AI-based educational technology, and used it to portray secondary-level school teachers’ attitudes towards AI. The final teacher-focused paper, by Lin & Brummelen [18], described workshops co-designed with 15 K-12 teachers, where teachers and researchers co-created lesson plans using AI tools and embedding AI concepts into various core subjects.

Targeted Groups: A number of the studies included in the review described work focused on specific, typically underrepresented groups. For example, a 2008

Table 3. The number of student participants that are in each study included in the review. * Exact number not provided.

Student participants	N	
1–10 students	4	[33,41,42,45]
11–30 students	6	[2,4,9,11,15,40]
31–100 students	9	[6,7,12,16,20,21,27,30,32]*
100 + students	5	[19,24,26,31,36]
None/Unknown	9	[5,10,13,18,22,23,25,34,44]

study using video games [9] specifically focused on girls, as did the Stanford Artificial Intelligence Laboratory’s Outreach Summer (SAILORS) program [40]. In another study [7] each teacher chosen for the workshops was asked to recruit 5 or 6 of their students, with at least 50% of the students identifying as non-male. The researchers in this study also targeted particular schools to provide opportunities for enrichment to low income families. A 2021 paper by Druga & Ko [12] recounts how they specifically chose many different locations for their workshops to include a diverse population of students. A 2008 study [4] describes how fifteen blind high school students created and personalized instant messaging chatbots using C# and were guided by both blind and sighted mentors. A 2021 study [18] that explores teachers’ perspectives in the adoption of AI curricula and learning tools, prioritised teachers who taught non-STEM classrooms. In 2019, a storybook was produced as part of a workshop, and it was written in a colloquial style, to make the workshop content accessible to multiple people in a students’ household [30]. Finally, The Curated Pathways to Innovation (CPI) web tool [19] was launched in 2020. It gathered existing online resources for CS and specifically focused on K-12 girls and historically underrepresented groups, so as to help students navigate their career journeys in STEM, particularly in computers.

3.2 RQ2: What Course Content Appears in the Research?

A broad range of course topics were covered, or referenced in the search results—see Table 4. The most important are listed below.

Ethics/Social Good: Sinha & George [30] describe a course that introduced students to the basic idea of intelligence, examining analytical, emotional, moral and social intelligence. In a 2020 course [27] students designed interactions to make an interface more human-friendly and ethical. Bilstrup et al. [5] highlighted criticisms from the literature with regard to ethical issues of ML.

Data Collection/Analysis: Srikand & Aggarwal [32] described a half-day data science tutorial that was designed to expose students to the full cycle of a typical supervised learning approach, while Zimmermann-Niefield et al. [45] used wearable sensors to allow students to leverage their domain knowledge to collect

Table 4. The topics referenced or covered in the courses that are described in the papers included in the review.

Content	N	
Ethics/Social good	9	[5–7, 13, 18, 25, 27, 30, 34]
Data collection/Analysis	14	[5, 6, 10, 12, 15, 20, 21, 24, 32, 34, 36, 41, 44, 45]
Classifiers	10	[10, 12, 15, 20, 21, 30, 34, 36, 40, 45]
NLP	3	[15, 18, 40]
Computational biology	3	[31, 34, 40]
Problem solving techniques	3	[2, 23, 42]
XAI/Black-box solutions	5	[15, 20, 22, 34, 41]
Various	13	[5, 9, 10, 12, 16, 18, 21, 24, 27, 30, 31, 34, 40]

data, build models, test and evaluate them. Mike et al. [21] reported on the pilot implementation of their data science curriculum, while Perach and Alexandron [24] discussed an ML and Deep Learning blended learning programme that used MOOCs. Bilstrup et al. [6] facilitated students to explore different data types and sources in their card-based workshop.

Classifiers: A 2022 a study explored children’s interactions with a simple image classification tool, using two features to classify images, using their own image data [36]. In Vachovsky et al. [40] students programmed a linear classifier. They also built their own Naive Bayes classifier and some students also implemented K-nearest neighbors classification. Sinha & George [30] describe how students wrote R code to create a simple model to classify flowers.

Natural Language Processing (NLP): A 2021 pilot study by Hjorth [15] had students use the Natural Language Processing 4 All (NLP4All) tool to learn about the policy views and communication styles of political parties by classifying tweets. In Vachovsky et al. [40] students learned how to use NLP to determine which area of the world needed disaster relief e.g. water or medical care. During a co-design workshop [18] teachers made connections by starting with a core subject concept and relating it to AI. NLP was identified as having a potential connection to English.

Computational Biology: In Vachovsky et al. [40] students were taught some of technical methods that are used in computational biology, focusing on gene expression from different types of cancer. Computational Biology also appears in the ‘Introduction to Artificial Intelligence’ topic in Sperling and Lickerman’s 2012 curriculum [31]. Tedre et al. [34] highlighted biological computing as an emerging technology.

Problem Solving Techniques: In a 1985 paper [23] Ourusoff discusses the nature of intelligence, and describes techniques that can be used in the classroom to model problem solving behaviour In 2018, students were given a basic course of Python programming using an interactive tool with mathematical reasoning

problems [2]. Researchers analysed how students improved their approach to problems.

Explainable AI (XAI)/Black-Box Solutions: The NLP tool NLP4All was designed to support students through an XAI interface. It scaffolds students without coding skill, helping “students find relationships between categories and features without explicit a priori knowledge” [15]. Meanwhile, Wen et al. [41] contend that their study on face glyph data visualization demystifies ML by looking inside the black-box.

Various Topics: There are a number of additional topics appeared in our results. For example, Korf [16] examined circuits in their syllabus, Carmichael [9] looked at game design, Sperling & Lickerman [31] included algorithms and graph theory, and Vachovsky et al. [40] explored the topic of self driving cars. Sinha & George [30] covered the historical development of machines, and in Lin & Van Brummelen [18] teacher groups identified AI project ideas such as “endangered plant identification” or a “mobile app for home automation”. Chittora and Baynes [10] demonstrated a regression problem and Mike et al. [21] explored algorithms such as the K-nearest neighbors algorithm. Sabuncuoglu [27] developed a 36-week open-source AI curriculum, that included the history of human and computer interaction, prototyping, and soundwaves. They also provided students with the opportunity to complete a project to address a United Nations Sustainable Development Goal (SDG) [1] problem. Bilstrup et al. [5] mention the environmental cost of training ML models.

Coding: A coding element appeared in 14 of the 33 papers identified (42.4%)—see Table 5. Python is the most commonly used language in our results. A CS syllabus that was designed in 1983 included programming segments in LISP [16]. In a 1994 paper [33] three students used C to develop software to play the game Connect 1 and to control a robotic arm making the moves on a vertical grid of six rows and seven columns. Bigham et al. [4] describe how students created their own chatbots. A small subset of C# was taught to students in order to prevent them from being overwhelmed by the syntax. Sinha & George [30] outline how students were introduced to very basic programming using the R programming language. They were taught how to write code and to use the iris dataset to create a simple model to classify flowers. In another study [20] an ML method for object recognition was developed, and students were asked to use a language familiar to them (typically C, C++, Python or JavaScript). The teacher, with the help of the students, built a web app (HTML and JavaScript) that learned to recognize objects when shown to a camera. A 2020 study used Zenbo as the development tool to write Zenbo Scratch for a robotic quiz game system for primary school students [42]. A software engineering curriculum for high school students used the DrRacket functional programming language [31]; while a 2022 course that taught reinforcement learning (RL) through virtual robotics, used Swift [11]. A blended learning program in 2022 used a series of Massive Open Online Courses (MOOCs) by Professor Andrew Ng. While not identified in the literature, these courses typically use Octave (MATLAB) [24].

Table 5. The programming languages used in the courses that are described in the papers included in the review.

Language	N		Language	N		Language	N	
MATLAB/Octave	1	[24]	Swift	1	[11]	Various	1	[20]
DrRacket	1	[31]	Zenbo Scratch	1	[42]	C#	1	[4]
Python	5	[2, 10, 19, 21, 40]	LISP	1	[16]	C	1	[33]
			R	1	[30]			

3.3 RQ3: Where Has the Research Taken Place?

The United States (US) was the most represented country in the research, with 14 of the 33 search results (42.4%) being based there, or being the principal location of the work/authors—see Table 6.

Table 6. Location of Research in 33 papers selected from 197 search results.

Country	N		Country	N		Country	N	
Peru	1	[2]	Finland	2	[34, 36]	Taiwan	1	[42]
Israel	4	[21, 22, 24, 31]	Turkey	1	[27]	Romania	1	[20]
Denmark	3	[5, 6, 15]	India	2	[30, 32]	Canada	1	[9]
United States	14	[4, 7, 10–12, 16, 18, 19, 23, 33, 40, 41, 44, 45]	Various	2	[13, 25]	Spain	1	[26]

4 Discussion

In addition to illustrating the landscape of AI/ML education at the K-12 level, this review gives rise to a number of important questions which we discuss here.

Which groups do not appear in the research? Aside from Eaton et al. [13] here is little evidence of qualitative research that involved experts, the school community or parents in the search results (there are some minor references to these groups [12, 18, 25, 27, 30, 36, 42]). It may be helpful in future work to gain these broader perspectives.

What do we know about the students who have been the focus of the AI/ML K-12 educational research? Overall, in terms of targeted approaches towards disadvantaged groups, we can glean very little from these search results. It is difficult to determine who is progressing and who is getting left behind. This is reflected in a 2020 study by Upadhyaya et al. [39] in the US, who analyzed seven years of K-12 computing education research data. They identified that “while it is clear that computing has entered the K-12 space, what

is still not clear is how equitable the access is to the computing due to data that is either not being collected or analyzed or is being under-reported". Bryant et al. [8] highlight how the stereotypes about "who does computer science" can preclude interest in the field with many perceiving computing as "irrelevant" and "asocial". They state that the "underrepresentation in computer science of women, domestic students of color, and students of lower socioeconomic status" is a national issue. A number of results in the search employed the use of robots, smart speakers or specially designed web based environments. Dietz et al. [11] highlighted issues such as "costly specialized equipment and ample physical space" as "barriers that limit access".

What content has not been addressed in the search results? A wide range of content appeared in the search results, but as Tedre et al. [34] have highlighted, there are other topics that are "working their way toward us". These include quantum computing, and neuromorphic computing. Auccahuasi et al. [2] maintain that Python "is being widely applied" in AI and quantum computing, as there are "multiple libraries such as numpy and scipy for scientific data processing, scikit-learn and TensorFlow for artificial intelligence and QISKit for quantum computing, which are constantly used, reviewed and improved by a large community of programmers". There is little reference to frameworks or curricula in the literature. Both Mike et al. [21] and Sperling & Lickerman [31] set out curricula that are linked to the Israeli high school CS curriculum. Hjorth [15] presents a learning unit that has been aligned with the Danish national standards for Social Studies. Sabuncuoglu has set out a proposed long term curriculum [27], based upon Touretzky's [37] 'Five Big Ideas in AI Education'. Polak et al. [25] used the 'Will, Skill, Tool' model as a theoretical lens, to guide the design of educational content and online platforms, so as to enable teachers to integrate AI education into their classroom.

5 Conclusion

Based on these results we have found a recent, marked increase in AI/ML K-12 computing education research, which has mainly taken place in the United States. There are wide variations in the age ranges of students involved and the number of student participants in each study. There is very little research that specifically focuses on teachers *teaching* AI/ML (although there is no shortage of use-cases for AI and ML aiding teachers [3] this was not a focus of this research). Reference to experts, parents, or the wider school community is also minimal. Further, a very small proportion of the research is focused on girls or those from historically underrepresented groups. We found a lack of clarity around equity of access to AI/ML K-12 courses and, overall, we are unsure as to how successful AI/ML K-12 courses have been at recruiting girls to and/or ultimately helping retain women in CS. We have identified evidence of the emergence of Python coding in K-12 courses as the dominant language used. Finally, there are wide variations in course content, and little alignment to CS frameworks or curricula. We have identified a number of open questions in the research work on K-12 AI/ML education and these will be addressed in future work.

References

1. Arora, N.K., Mishra, I.: United nations sustainable development goals 2030 and environmental sustainability: race against time. *Environ. Sustain.* **2**(4), 339–342 (2019)
2. Auccahuasi, W., Santiago, G.B., Núñez, E.O., Sernaque, F.: Interactive online tool as an instrument for learning mathematics through programming techniques, aimed at high school students. In: *Proceedings of the 6th International Conference on Information Technology: IoT and Smart City*, pp. 70–76. ACM (2018)
3. Becker, B.A.: Artificial intelligence in education: what is it, where is it now, where is it going? *Ireland's Yearbook Education 2017–2018*, 30, vol. 1, pp. 42–48. Education Matters, Dublin, Ireland (2017). ISBN 978-0-9956987-1-0
4. Bigham, J.P., Aller, M.B., Brudvik, J.T., Leung, J.O., Yazzolino, L.A., Ladner, R.E.: Inspiring blind high school students to pursue computer science with instant messaging chatbots. In: *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education, SIGCSE*, pp. 449–453. ACM (2008)
5. Bilstrup, K.K., Kaspersen, M.H., Assent, I., Enni, S., Petersen, M.G.: From demo to design in teaching machine learning. In: *FAccT 2022: 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 2168–2178. ACM (2022)
6. Bilstrup, K.K., Kaspersen, M.H., Petersen, M.G.: Staging reflections on ethical dilemmas in machine learning: a card-based design workshop for high school students. In: *DIS 2020: Designing Interactive Systems Conference 2020*, pp. 1211–1222. ACM (2020)
7. Brummelen, J.V., Tabunshchik, V., Heng, T.: “Alexa, can I program you?”: student perceptions of conversational artificial intelligence before and after programming Alexa. In: *IDC 2021: Interaction Design and Children*, pp. 305–313. ACM (2021)
8. Bryant, C., et al.: A middle-school camp emphasizing data science and computing for social good. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education, SIGCSE*, pp. 358–364. ACM (2019)
9. Carmichael, G.: Girls, computer science, and games. *ACM SIGCSE Bull.* **40**, 107–110 (2008)
10. Chittora, S., Baynes, A.: Interactive visualizations to introduce data science for high school students. In: *SIGITE 2020: The 21st Annual Conference on Information Technology Education*, pp. 236–241. ACM (2020)
11. Dietz, G., Chen, J.K., Beason, J., Tarrow, M., Hilliard, A., Shapiro, R.B.: Artonomous: introducing middle school students to reinforcement learning through virtual robotics. In: *IDC 2022: Interaction Design and Children*, pp. 430–441. ACM (2022)
12. Druga, S., Ko, A.J.: How do children’s perceptions of machine intelligence change when training and coding smart programs? In: *IDC 2021: Interaction Design and Children*, pp. 49–61. ACM (2021)
13. Eaton, E., et al.: Blue sky ideas in artificial intelligence education from the EAAI 2017 new and future AI educator program. *AI Matters* **3**, 23–31 (2018)
14. Gardner, T., Leonard, H.C., Waite, J., Sentance, S.: What do we know about computing education for K-12 in non-formal settings? A systematic literature review of recent research. In: *ICER 2022: ACM Conference on International Computing Education Research*, pp. 264–281. ACM (2022)
15. Hjorth, A.: Naturallanguageprocesing4all: - a constructionist NLP tool for scaffolding students’ exploration of text. In: *ICER 2021: ACM Conference on International Computing Education Research*, pp. 347–354. ACM (2021)

16. Korf, R.E.: A computer science syllabus for gifted pre-college students. In: Proceedings of the 14th SIGCSE Technical Symposium on Computer Science Education, pp. 237–240. ACM (1983)
17. Lee, S.Y., et al.: Designing a collaborative game-based learning environment for AI-infused inquiry learning in elementary school classrooms. In: Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE, p. 566. ACM (2020)
18. Lin, P., Brummelen, J.V.: Engaging teachers to co-design integrated AI curriculum for K-12 classrooms. In: CHI 2021: CHI Conference on Human Factors in Computing Systems, pp. 239:1–239:12. ACM (2021)
19. Linnel, N., et al.: Curated pathways to innovation: personalized CS education to promote diversity. *J. Comput. Sci. Coll.* **35**(10), 39–45 (2020)
20. Mariescu-Istodor, R., Jormanainen, I.: Machine learning for high school students. In: Koli Calling 2019: 19th Koli Calling International Conference on Computing Education Research, pp. 10:1–10:9. ACM (2019)
21. Mike, K., Hazan, T., Hazzan, O.: Equalizing data science curriculum for computer science pupils. In: Koli Calling '20: 20th Koli Calling International Conference on Computing Education Research, pp. 20:1–20:5. ACM (2020)
22. Nazaretsky, T., Cukurova, M., Alexandron, G.: An instrument for measuring teachers' trust in AI-based educational technology. In: LAK 2022: 12th International Learning Analytics and Knowledge Conference, pp. 56–66. ACM (2022)
23. Ourusoff, N.: The physical symbol system hypothesis of Newell and Simon: a classroom demonstration of artificial intelligence. *ACM SIGCSE Bull.* **17**, 19–23 (1985)
24. Perach, S., Alexandron, G.: A blended-learning program for implementing a rigorous machine-learning curriculum in high-schools. In: L@S 2022: Ninth ACM Conference on Learning @ Scale, pp. 267–270. ACM (2022)
25. Polak, S., Schiavo, G., Zancanaro, M.: Teachers' perspective on artificial intelligence education: an initial investigation. In: CHI 2022: CHI Conference on Human Factors in Computing Systems, pp. 431:1–431:7. ACM (2022)
26. Rodríguez-García, J.D., Moreno-León, J., Román-González, M., Robles, G.: Evaluation of an online intervention to teach artificial intelligence with learningml to 10–16-year-old students. In: SIGCSE 2021: The 52nd ACM Technical Symposium on Computer Science Education, pp. 177–183. ACM (2021)
27. Sabuncuoglu, A.: Designing one year curriculum to teach artificial intelligence for middle school. In: Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE, pp. 96–102. ACM (2020)
28. Sanusi, I.T.: Intercontinental evidence on learners' differentials in sense-making of machine learning in schools. In: Koli Calling 2021: 21st Koli Calling International Conference on Computing Education Research, pp. 46:1–46:2. ACM (2021)
29. Sanusi, I.T.: Teaching machine learning in K-12 education. In: ICER 2021: ACM Conference on International Computing Education Research, pp. 395–397. ACM (2021)
30. Sinha, S., George, C.P.: Artificial intelligence for all using R programming language. *AI Matters* **5**, 10–13 (2019)
31. Sperling, A., Lickerman, D.: Integrating AI and machine learning in software engineering course for high school students. In: Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education, pp. 244–249 (2012)
32. Srikant, S., Aggarwal, V.: Introducing data science to school kids. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, SIGCSE, pp. 561–566. ACM (2017)

33. Stuart, B.L.: Connect 4 as a problem in artificial intelligence and robotics. *ACM SIGCSE Bull.* **26**, 41–46 (1994)
34. Tedre, M., Denning, P.J., Toivonen, T.: CT 2.0. In: Koli Calling '21: 21st Koli Calling International Conference on Computing Education Research, pp. 3:1–3:8. ACM (2021)
35. Tedre, M., et al.: Teaching machine learning in K-12 classroom: pedagogical and technological trajectories for artificial intelligence education. *IEEE Access* **9**, 110558–110572 (2021)
36. Toivonen, T., et al.: Interacting by drawing: introducing machine learning ideas to children at a K-9 science fair. In: CHI 2022: CHI Conference on Human Factors in Computing Systems, pp. 16:1–16:5. ACM (2022)
37. Touretzky, D.S., Gardner-McCune, C., Martin, F., Seehorn, D.W.: Envisioning AI for K-12: what should every child know about AI? In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI, pp. 9795–9799. AAAI Press (2019)
38. Touretzky, D.S., Martin, F., Seehorn, D.W., Breazeal, C., Posner, T.: Special session: AI for K-12 guidelines initiative. In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education, SIGCSE, pp. 492–493. ACM (2019)
39. Upadhyaya, B., McGill, M.M., Decker, A.: A longitudinal analysis of K-12 computing education research in the united states: Implications and recommendations for change. In: Proceedings of the 51st ACM Technical Symposium on Computer Science Education, SIGCSE, pp. 605–611. ACM (2020)
40. Vachovsky, M.E., Wu, G., Chaturapruek, S., Russakovsky, O., Sommer, R., Fei-Fei, L.: Toward more gender diversity in CS through an artificial intelligence summer program for high school girls. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education, SIGCSE, pp. 303–308. ACM (2016)
41. Wan, X., Zhou, X., Ye, Z., Mortensen, C.K., Bai, Z.: Smileycluster: supporting accessible machine learning in K-12 scientific discovery. In: Proceedings of the 19th ACM International Conference on Interaction Design and Children, IDC 2020, London, United Kingdom, 17–24 June 2020, pp. 23–35. ACM (2020)
42. Weng, T., Li, C., Hsu, M.: Development of robotic quiz games for self-regulated learning of primary school children. In: AICCC 2020: 2020 3rd Artificial Intelligence and Cloud Computing Conference, pp. 58–62. ACM (2020)
43. Wong, G.K.W., Ma, X., Dillenbourg, P., Huan, J.: Broadening artificial intelligence education in K-12: where to start? *ACM Inroads* **11**, 20–29 (2020)
44. Zhou, X., Li, K., Munawar, A.M., Bai, Z.: Scaffolding design to bridge the gaps between machine learning and scientific discovery for K-12 STEM education. In: IDC 2021: Interaction Design and Children, pp. 604–609. ACM (2021)
45. Zimmermann-Niefield, A., Turner, M., Murphy, B., Kane, S.K., Shapiro, R.B.: Youth learning machine learning through building models of athletic moves. In: Proceedings of the 18th ACM International Conference on Interaction Design and Children, IDC, pp. 121–132. ACM (2019)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Author Gender Identification Considering Gender Bias

Manuela Nayantara Jeyaraj^(✉)  and Sarah Jane Delany 

Technological University Dublin, Dublin, Ireland

manuela.n.jeyaraj@mytudublin.ie, sarahjane.delany@tudublin.ie

Abstract. Writing style and choice of words used in textual content can vary between men and women both in terms of who the text is talking about and who is writing the text. The focus of this paper is on author gender prediction, identifying the gender of who is writing the text. We compare closed and open vocabulary approaches on different types of textual content including more traditional writing styles such as in books, and more recent writing styles used in user generated content on digital platforms such as blogs and social media messaging. As supervised machine learning approaches can reflect human biases in the data they are trained on, we also consider the gender bias of the different approaches across the different types of dataset. We show that open vocabulary approaches perform better both in terms of prediction performance and with less gender bias.

Keywords: Author gender identification · Gender bias · Open-vocabulary approach

1 Introduction

During the 2017 Labor leadership election in Britain, an analysis of the language used in news articles about the candidates showed discrepancies related to their gender in how they were described¹. The single male candidate was more likely to be discussed in terms of professional employment, politics and law and order and the two female candidates were much more likely to be discussed in terms of their families, in particular their fathers.

The language style, choice of words, etc. in text differs between men and women [3]. This can be viewed from 2 perspectives; one is towards the subject of the text (inferring whether the person discussed in the text is male or female), and the other is towards the author of the text (inferring whether the author of that text is male or female based on their style of writing). Our focus in this paper is on author gender identification which is the latter case.

Previous research in supervised learning for author gender prediction has generally used a closed vocabulary approach [9, 36]. The vocabulary used to represent the text is typically a list of characteristics of the text structure and content such as character frequencies and word or sentence count, vocabulary richness measures and the frequencies

¹ Gender bias in Political description of candidates: <https://www.theguardian.com/technology/2017/apr/13/ai-programs-exhibit-racist-and-sexist-biases-research-reveals>.

of an extensive list of predefined set of words and phrases identified through psychological or linguistic studies. In contrast, we show that an open vocabulary approach using feature selection, a data-driven approach that dynamically identifies the words that are more predictive of the author gender, performs significantly better than the closed vocabulary approach.

We evaluate the closed and open approaches on different types of textual content including (i) user-generated content which reflects the more modern, digital writing style such as tweets and blogs and (ii) text content that follow a more conventional writing style using eBooks from the Gutenberg digital repository.

Prediction models are often trained on datasets that reflect human bias and learn the same biases provided as examples to them [8]. This can lead to models making biased decisions that reflect human biases, including gender bias [37]. We show that the open-vocabulary approach displays significantly less gender bias than the closed approaches across all the datasets.

We also explore a hybrid closed and open approach, using a significantly smaller set of features which we call the POS (Parts-of-Speech) feature set. Though these POS features reflect a closed-vocabulary approach as they measure the proportions of word usage in text, they can be considered as moving towards a more open-vocabulary approach as they capture how different parts of speech are used. We found that combining the proposed POS feature set with a features obtained using an open vocabulary approach increases the capacity to identify author gender without having a significant impact on gender bias.

The rest of the paper is structured as follows. The following section outlines related work in author gender prediction. Section 3 outlines our methodology, Sect. 4 presents our evaluation and results while we conclude and outline our future work in Sect. 5.

2 Related Work

Initial work in the area of attributing text content to author gender used closed vocabularies and statistical methods [5, 7]. The closed vocabularies used extensive lists of stylometric textual characteristics, e.g., word frequencies, word length, and sentence count [4]. Since such count-based features were characterized by the length of the text, lists of vocabulary richness measures such as the hapax' legomena, Yule's K, etc., that described the lexical structure of a document independent of the length of text, were introduced [20, 40]. These vocabulary richness measures were originally defined for the author attribution tasks [20], but over time were adapted for author gender prediction [11, 21, 25, 40].

In addition to using stylometric features, researchers started exploring if the use of particular words in text can be attributed to a particular gender [26, 28]. This gave rise to the use of function words which include article words, pronouns, conjunctions, etc. as closed-vocabulary features [16]. Building on the idea of using a predefined dictionary of words as features, Tausczik et al. [38] used a set of words and phrases introduced by Pennebaker et al. [31] in their study on the psychometric properties of words from psychological or linguistic studies. These features were known as the LIWC features (Linguistic Inquiry and Word Count) [30].

Gradually, researchers started exploring the application of supervised ML techniques on these closed vocabulary features [2, 3, 6, 12]. A variety of classification techniques have been used, including Winnow [12], Decision trees [2], SVM [9], Random forest [34]. The limitation with the closed-vocabulary approach is that it requires an extensive list curated by humans, based on the counts or number of occurrences of words. As an example, the popular LIWC2015 dictionary is an extensive list of approximately 6,400 identified words [30]. Cheng et al. [9] chose to use 545 closed vocabulary features where they included function words as features on top of stylometric features. Feature selection techniques were then applied to reduce this vocabulary. Koppel et al. [24] attempted to identify the optimal number of features that can effectively predict an author's gender by performing feature reduction using multiplicative update rules where a weight vector is learned by iteratively going through each training instance. After the weights for all features are learned, the less prominent features displayed a weight that tend to zero. Using such a feature selection method, they were able to observe that the top 64 to 128 features were sufficient to effectively predict an author's gender.

Researchers started exploring open vocabulary methods to automatically identify content-based features that are indicative of an author's gender. Open vocabulary methods typically use a Bag of Words approach to identify the vocabulary across all training data. This resulted in very high dimensionality and a sparse representation. Hence, topic modelling approaches were used to identify a reduced set of features [23] which were shown to perform better than the closed vocabularies on the task [41]. One study found a subset of 83 closed vocabulary features outperformed content based features [41]. However, the comparison is against the top 1000 to 3000 content words with the highest tf-idf frequency values which does not necessarily select content features that are useful for distinguishing male and female authors.

The classification techniques used ranged from logistic regression [9], Adaboost [34], Random forest [29], through to SVM with linear kernel [9, 18]. The datasets used varied from proprietary non-open datasets from Facebook [15], blogs [27], news corpora [9], short-messaging-service (SMS) texts [14], to publicly available data such as the original Enron dataset² which originally had gender information but this has now been removed from the dataset [12].

The PAN CLEF (Conference and Labs of the Evaluation Forum) 2017 challenge has involved differentiating human authored from bot-generated text in twitter data and included the task of author gender identification. Some of the approaches to this challenge used word embeddings to represent the text [1, 10] however the best performing approach used tf-idf representation with topic modelling in the multi-class classification task of identifying bot-generated from male and female authored tweets.

The closest work to ours is the work done by Fatima et al. [15] which concluded that content based approaches with feature selection can be used for multilingual text. They evaluated a range of classification and feature selection approaches on a single proprietary Facebook posts and comments dataset. Our focus is on different styles and lengths of English language content and we consider gender bias.

² Enron dataset: <https://www.cs.cmu.edu/~enron/>.

3 Approach

We used 4 different datasets, each being representative of different lengths of text and different writing styles (traditional and more modern user-generated content). The characteristics of the datasets used are included in Table 1.

Table 1. Dataset description.

Dataset	Size (number of samples)	Class distribution		Min #chars	Max #chars	Avg #chars	Writing style	Length
		Male (%)	Female (%)					
eBooks dataset	18,398	50	50	2,174	19,228,992	491,602	Traditional-writing	Long-text
Race-gender Blogs dataset	1,230	63	37	800	16,918	1,100	User-generated	Medium-length
Blogger Blogs dataset	72,720	57	43	208	83,146	751	User-generated	Medium-length
Twitter dataset	205,367	50	50	1	140	96	User-generated	Short-text

The **Twitter dataset** is adapted from an original dataset provided by Rangel et al. [33] which was used to differentiate bot-generated tweets from human-authored tweets. We removed the bot-generated tweets and used only those generated by either a male or female human author. The dataset includes 100 tweets for each author and is a balanced dataset with 50% female-authored and 50% male-authored tweets. With the maximum number of characters in a tweet being 140 characters, this dataset is considered as short text content.

The **Race-gender Blogs dataset** was taken from the recent work published by Kambhatla et al. [22] where it was used to identify racial stereotypes using identity portrayal. The dataset was compiled from crowd-sourced workers on prolific.com where they were asked to provide blogs they've written with self-identified gender and racial information. This dataset is labelled as the author gender for each blog text is known.

The **Blogger Blogs dataset** was adapted from a dataset published by Schler et al. [35] which was scraped from blogs over 200 words published on blogger.com that included author-provided indication of gender. We removed blogs that contained words from languages other than English ending up with 72,789 blogs from 19,230 unique authors, with 57% male- and 43% female-authored instances.

The **eBooks dataset** is a set of English language long-text eBooks freely indexed by epub and kindle eBooks under the Gutenberg eBooks project [17]. Since the author gender is not available with the meta-data for each eBook, we used [gender.api](https://gender-api.com/)³ and [genderize](https://genderize.io/)⁴ APIs to infer the gender of the author based on their first name/s. The books where the gender inferred from both APIs matched were retained. There are significantly more male authored books available in Gutenberg than female authored books.

³ Gender-api: <https://gender-api.com/>.

⁴ Genderize API: <https://genderize.io/>.

We took all female-authored books available to us and randomly selected an equal number of male-authored books for our dataset. The resulting dataset included 18,398 books equally balanced between male and female authors.

For our evaluation, all 4 datasets above are split on a train-test split of 70:30. Parameter tuning was performed on the training data using cross validation to obtain the optimal set of hyper parameters for the SVM classifier.

We considered different feature sets to observe the effect that these features have in predicting the gender of the author from text. Our aim was to explore the differences between using the existing closed vocabulary feature sets and more open vocabulary feature sets that are derived from the textual content.

Closed-vocabulary features were derived from work by Koppel et al. [24] and Cheng et al. [9]. We implemented 66 stylometric character, word and structural features that were commonly identified as the significant discriminators of gender from the above research works (see Fig. 2).

In addition, all 373 function word features presented in Cheng et al. [9] were included in our closed-vocabulary features as well. This rendered a closed-vocabulary feature set of 439 features.

Content features are the dynamic, open-vocabulary words obtained directly from the text. We used a tf-idf term weighting representation to represent our open-vocabulary content features similar to [10]. This results in a very high dimensional, sparse vector representation for each document. We used a Chi-squared filter feature selection technique on each dataset and selected the top ranking 10,000 features as our open-vocabulary representation which we call the *content features*. In our evaluation, we explore the impact on performance of different numbers of content features from the open vocabulary set.

POS Proportion Features. The function words used in the closed vocabulary approach try to capture differences in gender writing style identified by linguistic and psychological studies [13]. Inspired by this, we used a feature set of 16 features which we call the POS features. They capture the frequency of use of different types of words which are identified by part-of-speech tagging the text content. Table 3 lists these features. While these may appear more like closed vocabulary features, the fact that they focus on different types of speech based on the word's syntactic function rather than a lexicon of words moves this set towards the open vocabulary approach.

We used an SVM classifier with a linear kernel as the classifier in our experiments. Preliminary results on the performance of a variety of classifiers across both open- and closed-vocabulary features showed that the SVM with a linear kernel performed consistently well. In addition, SVMs are commonly used for text classification tasks [39, 42].

To measure task performance on the task of gender author classification we used average class recall or accuracy across the male and female authored classes. To measure the gender bias of a model that predicts author gender we used the TPR_{gap} measure [32], as defined in Eq. 1 which measures the differences in the gender specific true positive rates.

$$TPR_{gap} = |TPR_{male} - TPR_{female}| \quad (1)$$

Table 2. 66 Stylometric closed-vocabulary features.

Feature ID	Character-based Features	Feature ID	Word-based Features
F1	Total number of characters (C)	F35	Total number of words (N)
F2	Proportion of total number of letters(a-z)	F36	Average length per word (in characters)
F3	Proportion of total number of upper characters	F37	Vocabulary richness - Type-token ratio (Lexical diversity)
F4	Proportion of total number of digital characters	F38	Herdan's C Measure
F5	Proportion of total number of white-space characters	F39	Guidan's Root TTR
F6	Proportion of total number of tab space characters	F40	Carroll's corrected TTR
F7-F34 Proportion of special characters (% , , etc.)		F41	Dugast's Uber Index
F7	Proportion of apostrophe	F42	Summer's Index
F8	Proportion of brackets	F43	Hapax legomena
F9	Proportion of colons	F44	Hapax dislegomena
F10	Proportion of comma	F45	Yules I measure
F11	Proportion of dashes	F46	Yules K measure
F12	Proportion of ellipsis	F47	Simpsons D measure
F13	Proportion of exclamation marks	F48	Herdan's Vm measure
F14	Proportion of full stops	F49	Sichels S measure
F15	Proportion of question marks	F50	Honores R measure
F16	Proportion of semi colons	F51	Entropy measure
F17	Proportion of slashes	F52-F55 Proportion of word length frequency distribution	
F18	Proportion of ampersands	F52	Proportion of words with length 3
F19	Proportion of asterisks	F53	Proportion of words with length 4
F20	Proportion of at signs	F54	Proportion of words longer than 6 characters
F21	Proportion of dollar signs	F55	Proportion of short words (1-3 characters)
F22	Proportion of equal signs		
F23	Proportion of greater than signs	Feature ID	Structural Features
F24	Proportion of less than signs	F56	Total number of lines
F25	Proportion of percentage signs	F57	Total number of sentences (S)
F26	Proportion of plus signs	F58	Total number of paragraphs (P)
F27	Proportion of left curly braces	F59	Proportion of sentences per paragraph
F28	Proportion of left square brackets	F60	Proportion of words per paragraph
F29	Proportion of left parentheses	F61	Proportion of characters per paragraph
F30	Proportion of right curly braces	F62	Proportion of words per sentence
F31	Proportion of right square brackets	F63	Proportion of sentences beginning with upper case
F32	Proportion of right parentheses	F64	Proportion of sentences beginning with lower case
F33	Proportion of underscores	F65	Proportion of sentences beginning with quotation
F34	Proportion of vertical lines	F66	Proportion of blank lines

This measure is an equality of opportunity measure where predictions are independent of gender but conditional on the ground truth or actual outcomes in the training data [19]. This uses a democratic parity measure which insists on equal outcomes for both genders regardless of prevalence or ground truth.

4 Evaluation

Figure 1a shows the average class accuracy on different feature sets across all the datasets.

The content feature set which is the open vocabulary approach significantly outperforms the closed vocabulary features across all three datasets. The newly proposed 16

Table 3. POS Features.

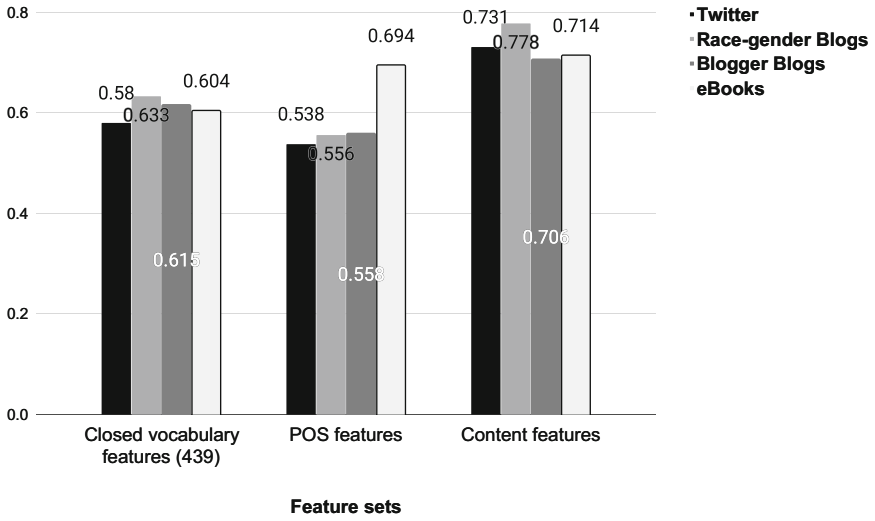
Feature ID	POS features
F67	Proportion of adjectives
F68	Proportion of adverbs
F69	Proportion of adposition words
F70	Proportion of auxiliary words
F71	Proportion of coordinating conjunctions
F72	Proportion of interjection
F73	Proportion of nouns
F74	Proportion of numerals
F75	Proportion of particle words
F76	Proportion of pronouns
F77	Proportion of proper nouns
F78	Proportion of subordinating conjunctions
F79	Proportion of symbols
F80	Proportion of verbs
F81	Proportion of determiner words
F82	Proportion of other

POS features perform better than the closed vocabulary features on the more structured, long-text eBooks dataset but does not work as well as the closed-vocabulary features on the user-generated content in the twitter and blogs datasets. This may be due to the nature of user generated digital content such as tweets and blogs which can have irregular and incomplete sentences and depend more on the use of slang, acronyms and emoticons. As the POS feature set uses different types of speech based on the word's syntactic function this requires the text to have a certain level of structure to it. However with only 16 features in the POS feature set, it performs very well compared with the significantly larger numbers of features required by the other two feature sets.

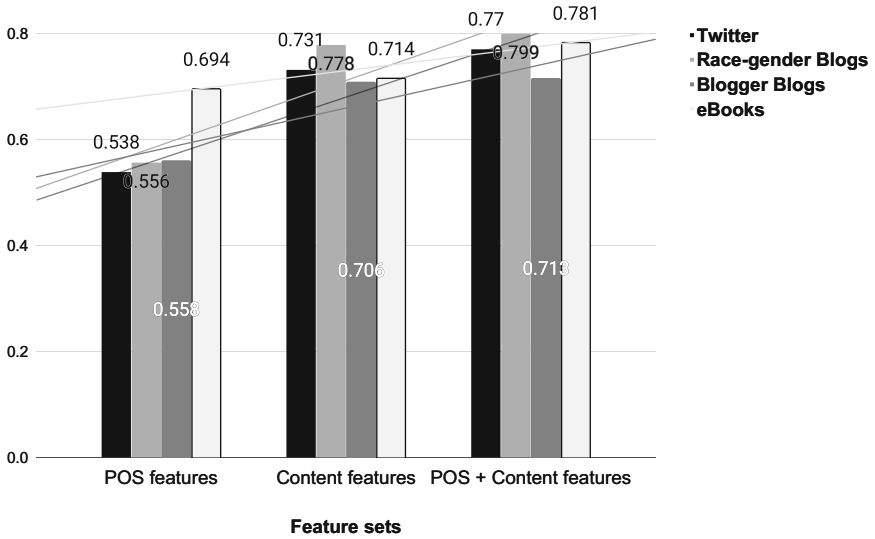
Figure 1b shows the performance of the classifier as the POS features are combined with the open-vocabulary content features. Here, adding the 16 POS proportions to the content features increased the performance across all 4 datasets.

We also evaluated the feature sets for bias using the TPR_{gap} gender bias measure shown in Eq. 1. Figure 2 shows the gender bias of the classifier for each of the feature sets. The higher the value the more gender bias displayed. Bias displayed on the right side of the figure indicates that more male-authored documents are classified correctly than female-authored documents, meaning more female-authored documents are predicted as male than vice versa. We consider this as male gender bias. Bias displayed on the left side of the figure indicates female gender bias.

Overall the content features from the open vocabulary approach displays less gender bias than the closed vocabulary approach. Both approaches display mostly male gender bias across all four datasets with the level of gender bias for the eBooks dataset on the closed vocabulary features exceedingly high at 66%.



(a) Average class accuracy on individual feature sets.



(b) Addition of 16 POS Proportion features to open vocabulary content features.

Fig. 1. Classification performance on different feature sets across all four datasets.

The POS features display significantly less gender bias across all datasets except the blogs from the blogger dataset. Also, the POS feature set interestingly shifts the bias more towards female bias than male bias, particularly for the user-generated content. Though the addition of the POS features to the content features increased the prediction performance for all datasets, it has only shown a positive influence in reducing the gender bias for the more traditional eBooks dataset with the bias for the user-generated content datasets remaining more or less the same.

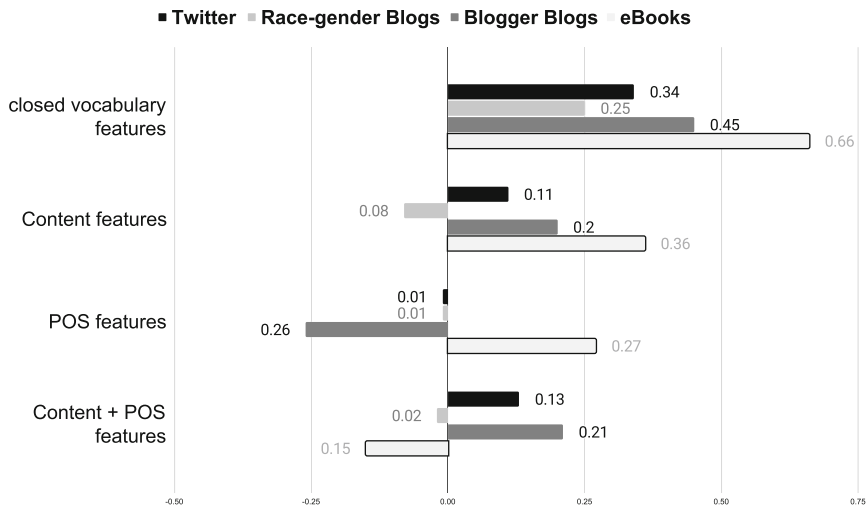


Fig. 2. Gender bias for all feature sets across all 4 datasets.

Given the good performance of the content features, we explored the impact of the number of content features used.

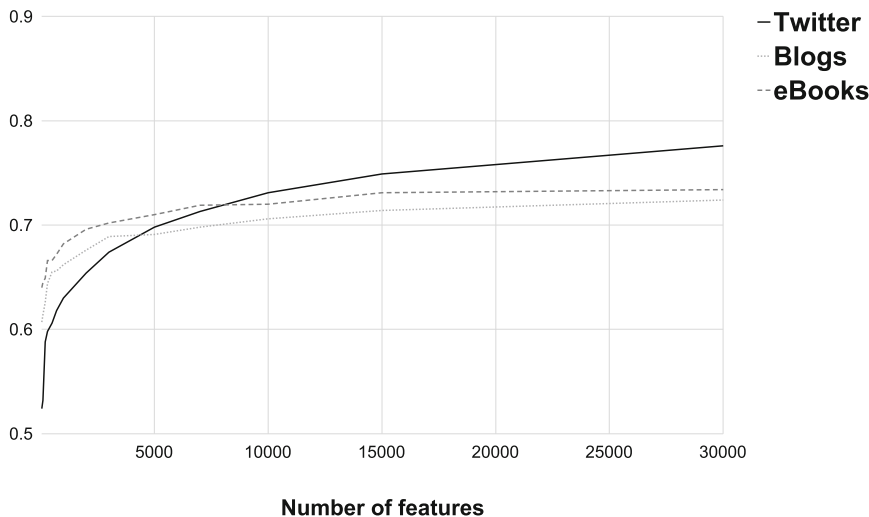


Fig. 3. Performance as the number of content features increases.

Figure 3 shows the average class accuracy as the number of features used increases for the eBooks, Blogger Blogs and Twitter datasets.

The graph shows that the performance for the Blogger Blogs and eBooks datasets level out at around 10,000 features but the performance steadily increases for the Twitter dataset. In fact, the performance continues to increase steadily even after 30,000

features with a classification performance of 0.8 at 100,000 features. This is not surprising as the Twitter dataset is considered short-text and the lack of text content would result in a very sparse representation reducing the signal in the text.

5 Conclusion

This research presents the impact of closed-vocabulary features and open-vocabulary features on author gender identification in terms of accuracy and gender bias. We were able to observe that open vocabulary features perform better than closed-vocabulary features in accurately identifying an author’s gender from text. In addition, we also propose a much smaller set of 16 POS features that reflect the frequency of usage of different parts-of-speech in the content. We suggest that these follow a more open-vocabulary approach. Though these POS features do not outperform the content features, they show much less gender bias as well as an interesting shift to female bias for the user-generated content. The addition of POS features to content features increased the prediction performance across all datasets while not significantly impacting the gender bias of the models.

As shown in Fig. 2, though the POS features display a generally lower gender bias than the content features, the addition of POS features to content features does not necessarily reduce the gender bias on user-generated content. Hence, further experimentation is required to explain this behaviour for the user-generated content.

By identifying the features that are highly predictive of the author’s gender, we hope to explore methods to effectively recommend linguistic modifications and provide positive reinforcement to authors about their language use to prompt a more gender-neutral writing style.

Acknowledgements. This work was funded by Science Foundation Ireland through the SFI Centre for Research Training in Machine Learning (18/CRT/6183). For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

1. Akhtyamova, L., Cardiff, J., Ignatov, A.: Twitter author profiling using word embeddings and logistic regression. In: CLEF (Working Notes) (2017)
2. Apte, C., Damerau, F., Weiss, S.M., Apte, C., Damerau, F., Weiss, S.: Text mining with decision trees and decision rules. In: In Proceedings of the Conference on Automated Learning and Discovery, Workshop 6: Learning from Text and the Web. Citeseer (1998)
3. Argamon, S., Koppel, M., Fine, J., Shimon, A.R.: Gender, genre, and writing style in formal written texts. *Text Talk* **23**(3), 321–346 (2003)
4. Aries, E.J., Johnson, F.L.: Close friendship in adulthood: conversational content between same-sex friends. *Sex Roles* **9**(12), 1183–1196 (1983)
5. Baayen, H., Van Halteren, H., Tweedie, F.: Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. *Lit. Linguist. Comput.* **11**(3), 121–132 (1996)
6. Burger, J.: Discriminating gender on Twitter. *EMNLP-Association for Computational Linguistics* (2011)

7. Burrows, J.F.: Not unless you ask nicely: the interpretative nexus between analysis and information. *Lit. Linguist. Comput.* **7**(2), 91–109 (1992)
8. Cadwalladr, C.: Google, democracy and the truth about internet search. *Guardian* **4**(12), 2016 (2016)
9. Cheng, N., Chandramouli, R., Subbalakshmi, K.: Author gender identification from text. *Digit. Investig.* **8**(1), 78–88 (2011)
10. Daneshvar, S., Inkpen, D.: Gender identification in twitter using N-grams and LSA. In: *proceedings of the Ninth International Conference of the CLEF Association (CLEF 2018)* (2018)
11. De Vel, O., Corney, M., Anderson, A., Mohay, G.: Language and gender author cohort analysis of e-mail for computer forensics. In: *Proceedings of Digital Forensics Research Workshop*, pp. 1–16 (2002)
12. Deitrick, W., Miller, Z., Valyou, B., Dickinson, B., Munson, T., Hu, W.: Author gender prediction in an email stream using neural networks (2012)
13. Eichstaedt, J.C., et al.: Closed-and open-vocabulary approaches to text analysis: a review, quantitative comparison, and recommendations. *Psychol. Methods* **26**(4), 398 (2021)
14. Fatima, M., et al.: Multilingual SMS-based author profiling: data and methods. *Nat. Lang. Eng.* **24**(5), 695–724 (2018)
15. Fatima, M., Hasan, K., Anwar, S., Nawab, R.M.A.: Multilingual author profiling on Facebook. *Inf. Process. Manag.* **53**(4), 886–904 (2017)
16. Garcia, A.M., Martin, J.C.: Function words in authorship attribution studies. *Lit. Linguist. Comput.* **22**(1), 49–66 (2006)
17. Gerlach, M., Font-Clos, F.: A standardized project Gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy* **22**(1), 126 (2020)
18. Grivas, A., Krithara, A., Giannakopoulos, G.: Author profiling using stylistic and structural feature groupings. In: *CLEF (Working Notes)* (2015)
19. Heidari, H., Loi, M., Gummadi, K.P., Krause, A.: A moral framework for understanding fair ML through economic models of equality of opportunity. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 181–190 (2019)
20. Holmes, D.I.: Authorship attribution. *Comput. Humanit.* **28**(2), 87–106 (1994)
21. Hoover, D.L.: Another perspective on vocabulary richness. *Comput. Humanit.* **37**(2), 151–178 (2003)
22. Kambhatla, G., Stewart, I., Mihalcea, R.: Surfacing racial stereotypes through identity portrayal. In: *2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1604–1615 (2022)
23. Kiatkawsin, K., Sutherland, I., Kim, J.Y.: A comparative automated text analysis of Airbnb reviews in Hong Kong and Singapore using latent Dirichlet allocation. *Sustainability* **12**(16), 6673 (2020)
24. Koppel, M., Argamon, S., Shimoni, A.R.: Automatically categorizing written texts by author gender. *Lit. Linguist. Comput.* **17**(4), 401–412 (2002)
25. Kucukyilmaz, T., Cambazoglu, B.B., Aykanat, C., Can, F.: Chat mining for gender prediction. In: Yakhno, T., Neuhold, E.J. (eds.) *ADVIS 2006. LNCS*, vol. 4243, pp. 274–283. Springer, Heidelberg (2006). https://doi.org/10.1007/11890393_29
26. Mehl, M.R., Pennebaker, J.W.: The sounds of social life: a psychometric analysis of students' daily social environments and natural conversations. *J. Pers. Soc. Psychol.* **84**(4), 857 (2003)
27. Mukherjee, A., Liu, B.: Improving gender classification of blog authors. In: *EMNLP*, pp. 207–217 (2010)
28. Mulac, A., Bradac, J.J., Gibbons, P.: Empirical support for the gender-as-culture hypothesis: an intercultural analysis of male/female language differences. *Hum. Commun. Res.* **27**(1), 121–152 (2001)

29. Palomino-Garibay, A., et al.: A random forest approach for authorship profiling. In: Proceedings of CLEF (2015)
30. Pennebaker, J.W., Boyd, R.L., Jordan, K., Blackburn, K.: The development and psychometric properties of LIWC2015. Technical report (2015)
31. Pennebaker, J.W., Francis, M.E., Booth, R.J.: Linguistic inquiry and word count: LIWC 2001. Mahway Lawrence Erlbaum Assoc. **71**(2001), 2001 (2001)
32. Prost, F., Thain, N., Bolukbasi, T.: Debiasing embeddings for reduced gender bias in text classification. *GeBNLP 2019* **9573**, 69 (2019)
33. Rangel, F., Rosso, P.: PAN19 author profiling: bots and gender profiling (2019). <https://doi.org/10.5281/zenodo.3692340>
34. Shboev, A., Litvinova, T., Gudovskikh, D., Rybka, R., Moloshnikov, I.: Machine learning models of text categorization by author gender using topic-independent features. *Proc. Comput. Sci.* **101**, 135–142 (2016)
35. Schler, J., Koppel, M., Argamon, S., Pennebaker, J.W.: Effects of age and gender on blogging. In: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, vol. 6, pp. 199–205 (2006)
36. Simaki, V., Aravantinou, C., Mporas, I., Kondyli, M., Megalooikonomou, V.: Sociolinguistic features for author gender identification: from qualitative evidence to quantitative analysis. *J. Quantit. Linguist.* **24**(1), 65–84 (2017)
37. Sun, T., et al.: Mitigating gender bias in natural language processing: literature review. *arXiv preprint arXiv:1906.08976* (2019)
38. Tausczik, Y.R., Pennebaker, J.W.: The psychological meaning of words: LIWC and computerized text analysis methods. *J. Lang. Soc. Psychol.* **29**(1), 24–54 (2010)
39. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.* **2**(Nov), 45–66 (2001)
40. Tweedie, F.J., Baayen, R.H.: How variable may a constant be? Measures of lexical richness in perspective. *Comput. Humanit.* **32**(5), 323–352 (1998)
41. Wanner, L., et al.: How to use less features and reach better performance in author gender identification. In: *LREC 2014*, pp. 1315–1319 (2014)
42. Zhang, W., Yoshida, T., Tang, X.: Text classification based on multi-word with support vector machine. *Knowl.-Based Syst.* **21**(8), 879–886 (2008)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Identity Term Sampling for Measuring Gender Bias in Training Data

Nasim Sobhani^(✉) and Sarah Jane Delany

Technological University Dublin, Dublin, Ireland

`nasim.x.sobhani@mytudublin.ie`, `sarahjane.delany@tudublin.ie`

Abstract. Predictions from machine learning models can reflect biases in the data on which they are trained. Gender bias has been identified in natural language processing systems such as those used for recruitment. The development of approaches to mitigate gender bias in training data typically need to be able to isolate the effect of gender on the output to see the impact of gender. While it is possible to isolate and identify gender for some types of training data, e.g. CVs in recruitment, for most textual corpora there is no obvious gender label. This paper proposes a general approach to measure bias in textual training data for NLP prediction systems by providing a gender label identified from the textual content of the training data. The approach is compared with the identity term template approach currently in use, also known as Gender Bias Evaluation Datasets (GBETs), which involves the design of synthetic test datasets which isolate gender and are used to probe for gender bias in a dataset. We show that our Identity Term Sampling (ITS) approach is capable of identifying gender bias at least as well as identity term templates and can be used on training data that has no obvious gender label.

Keywords: Machine learning · Gender bias · Evaluation

1 Introduction

Studies have shown gender bias in natural language processing tasks such as machine translation [18], co-reference resolution [17, 23, 25] and abusive and hate speech prediction [6, 14]. Gender bias has also been found in deployed NLP systems. In 2018 Amazon discontinued the use of an AI recruitment tool which showed significant bias against women¹ These downstream tasks that use machine learning models built on natural language content can reflect biases in the data on which they are trained.

The primary method to measure bias in a downstream task is to measure performance differences across gender as the system's performance should not be influenced by gender. This requires a way to isolate gender in the test instances

¹ <https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKC N1MK08G>.

which are used to measure performance. This is typically done by using synthetic test data that is appropriate for the task at hand. This test data is designed through the use of templates which can be filled in with content relevant for the task and duplicated for different gender identities. As an example, in an abusive content prediction task in work by [14], the template sentence “You are a *<adjective>* *<identity term>*” generated a number of test instances labelled for the classification task (abusive and non-abusive) and identified for gender. *<adjective>* was replaced with adjectives such as disgusting, filthy, nasty for abusive instances and adjectives such as lovely, excellent, incredible for non-abusive instances, while *<identity term>* was replaced with common gender identity pairs such as man/woman, boy/girl. This generated gender-swapped labelled test instances that were used to measure the difference in performance across genders.

There are some challenges with these template approaches. The artificial nature of the generated text does not reflect the true distribution and content of the task data. The templates have to be designed specifically for the downstream task and are not general across tasks. In addition the actual performance of these generated test datasets on the downstream task has been shown to be poor.

As an alternative to synthetic test data this paper proposes an approach to a more confident measure of gender bias by selecting appropriate test data from the original datasets and identifying their gender to allow the measurement of task performance across genders. Our approach, which we call Identity Term Sampling (ITS), is compared with the identity term template approach on the task of abusive content detection. We also apply it to a text classification task where the data is not typically expected to have gender bias and we show no significant gender bias evident.

The rest of this paper is structured as follows, Sect. 2 discusses related work in measuring gender bias in natural language tasks, Sect. 3 explains our ITS approach, Sect. 4 details the evaluation of our approach and the results and findings are discussed in Sect. 5.

2 Related Work

Natural Language Processing (NLP) models and systems are trained on human generated text content and they can reflect existing biases in the data when used in downstream applications [6, 14]. In addition to the training data itself, word embeddings which are distributed representations that are generated from large corpora of natural language and are used to represent words and sentences, can reflect and sometimes even amplify certain characteristics of the data including gender stereotypes [2, 3, 26].

As a first step towards reducing bias in an NLP system, we need to identify and measure any bias that might exist. Over the last few years a lot of research has been conducted to identify and measure bias in the training data [6, 11, 26] and in embeddings that might be used to represent the training data [2, 24]. An effective technique for evaluating bias in training data, which is known

as gender-swapping, involves replacing female/male definitional words by their equivalent male/female definitional words in the test set and comparing the overall performance of the system. The difference between the original test set and the gender-swapped results illustrates the system’s fairness [11].

Another technique to evaluate gender bias is generating a synthetic test set with test instances that isolate gender. This approach is called Gender Bias Evaluation Testsets (GBETs) by [21], and has been used to evaluate bias in a variety of different NLP tasks including sentiment analysis [10], abusive language detection [6, 14] and coreference resolution [17, 25].

GBETs can be generated in different ways depending on the NLP task to be tackled. For instance, a GBET for coreference resolution named GAP [23] is a human labeled ambiguous pronoun-name pairs corpus mined from Wikipedia. Similarly, to analyse gender bias in coreference resolution [7] constructed a dataset which is also scraped from Wikipedia, OpenSubtitle and Reddit comments. The template approach described above is also used to generate GBETs and involves creating sentence templates, that include gender identification words, appropriate for the downstream task. Pairs of sentences are generated from the template, one for each gender, and differences in the performance of the NLP system between the generated test sentences with a male and female gender identity facilitate the measurement of gender bias in the dataset. This gender identity template approach has been used in variety of different NLP tasks including sentiment analysis [10], abusive language detection [6, 14] and coreference resolution [17, 25].

More recently StereoSet [12] and CrowS-Pairs [13] GBETS have been proposed to evaluate bias in language models. These GBETs are crowd-sourced, template based which are created and annotated by crowdsourcing to measure bias in different domains. Each example consists of a pair of stereotype and anti stereotype sentences in case of CrowS-pairs. However, StereoSet contains of triplets of sentences with each instance corresponding to a stereotypical, anti-stereotypical or a meaningless association. An additional study presents a large GBET dataset called HOLISTICBIAS for measuring bias. This dataset is assembled by using a set of demographic descriptor terms in a set of bias measurement templates and can be used to test bias in language models [19].

There are a variety of measures used to detect gender bias in NLP methods [20]. Most of the recent work on evaluating gender bias in NLP systems use variations on Hardt et al.’s work on equalised odds and equal opportunity [9]. These measures are group measures and use the gender distributions in the training data rather than the democratic parity measure which insists on equal outcomes for both genders regardless of prevalence or ground truth.

There has also been a lot of work in identifying gender bias in word embeddings which have become a common form of representation of textual content in NLP systems. The existence of gender stereotypes in pre-trained word embeddings has been shown by [2, 24] and in contextualized word embeddings including ELMO by [1, 15]. The Word Embedding Association Test (WEAT) [3] has also

been proposed to measure model bias inside word embeddings through the difference in the strength of association concepts.

3 Approach

As the extent of gender bias in a natural language system is evident by the task performance differences across genders, the test data used to measure performance needs to include gender. The first step in the proposed approach, which we call Identity Term Sampling (ITS), is to identify the gender of instances in the training dataset in order to identify appropriate test instances which can be used to measure performance in the downstream task. Our approach then randomly selects the gendered test instances from the training data to be used to estimate the gender bias.

The gender identification step in Identity Term Sampling is based on the frequency of gender identity words in a data instance. ITS can assign gender to those instances that contain at least one gender identity term. The gender identity terms we use are those terms from a list of gender definitional pairs proposed in work by [2] and are given in Table 3b. These ten gender pairs were found by crowdsourcing to be the most frequent words used to define gender among a list of gender definitional and stereotype gender association words. For each instance in our datasets the frequency of male and female identity terms that occur in the text content is counted. The gender assigned to the data instance is the gender with the larger frequency of identity terms. Data instances with equal numbers of male and female gender identity terms are not identified with a gender as there was no obvious gender.

As an initial validation of the ITS approach we compared the gender identified by ITS against the actual gender on the BiasBios dataset [5], a dataset of 397,340 biographies across 28 different occupations. The ITS technique successfully identified 91.8% of the biographies correctly with only 4.1% misidentified and just over 4% were identified as no obvious gender.

To explore the gender identification approach we applied it to a number of datasets of user generated content which are used for text classification tasks. These datasets include two Twitter datasets used for the identification of abusive content and a review dataset used for sentiment analysis or opinion prediction. Twitter datasets used for abusive content detection are highly likely to exhibit bias and are used in other bias identification work [4, 14]. A hotel review dataset is less likely to exhibit gender bias in the training data.

The **Hate Speech** dataset [22] is a collection of almost 17K tweets consisting of 3,383 samples of sexist content, 1,972 samples of racist content and 11,559 neutral samples.

The dataset is transformed to a binary classification problem by labelling the sexist and racist samples as “abusive” class and neutral samples as “non abusive” class.

The **Abusive Tweets** dataset is a large scale crowd-sourced dataset, collected by [8]. The size of the dataset is just under 100k tweets and it is annotated

with four labels: *hateful*, *abusive*, *spam* and *none*. By combining the *none* and *spam* instances into a “non-abusive” class, and the hateful and abusive instances to an “abusive” class, we transform the dataset to a binary classification task, similar to the Hate Speech dataset.

The **Hotel Reviews** dataset has been scraped from booking.com and made available in Kaggle². The dataset contains almost 515,000 reviews and scores for 1493 luxury hotels across Europe. The classification task is to predict whether a textual review is a good or a bad review (i.e. a satisfied or unsatisfied customer). Each review in the dataset has a rating between 2.5 and 10 where higher is better in terms of satisfaction. The reviews were split into two classes: “unsatisfied” for reviews with a rating of less than 5, and “satisfied” for those with a rating of 5 or higher. The original dataset is highly imbalanced with 95% of the reviews in the “satisfied” class.

Table 1 shows the overall size and the per class and per gender distribution of data for the three datasets.

Table 1. Class distribution, gender identified data percentage and overall size for each dataset

Dataset	Class	Class%	Gender identified		Size
			F(%)	M(%)	
Hate speech	Abusive	31.4	3.6	1.6	16K
	Non abusive	68.6	1.9	3.3	
Abusive tweets	Abusive	32.1	2.0	2.9	100K
	Non abusive	67.9	2.2	4.5	
Hotel review	Unsatisfied	4.3	0.1	0.2	515K
	Satisfied	95.7	1.4	1.7	

To illustrate the effect of gender identification, each data instance is categorised into one of four groups. Data instances that do not have any of the gender identity words in them are categorised as No-Gender (NG). Data instances which contains equal numbers of male and female identity terms are categorised as Equal-Gender (EG). The other two categories are Positive Gender (PG) and Strongly Positive Gender (SPG) and use the proportion of male and female identity terms. The data instance is identified as the gender with the higher proportion of identity terms. If the proportion is between 50% and 75% the data instance is categorised as Positive Gender, and if it is 75% or higher, it is categorised as Strongly Positive Gender.

Table 2 describes the results of gender identification on the datasets, showing the size proportion of each dataset with gender identified and the proportion of the data of each category with gender identified. It is evident that most of the gendered data in all datasets is categorized as Strongly Positive indicating

² <https://www.kaggle.com/datasets/jiashenliu/515k-hotel-reviews-data-in-europe>.

that typically over 75% of the definitional words in the gendered data are for one specific gender. As a result of applying the proposed method, 11% of Hate Speech data and almost 12% of the Abusive Tweets data are gender identified. The Hotel Reviews dataset has significantly less gendered instances with only 3.6% with gender identified.

Table 2. The results of identifying gender in the datasets, showing the size and proportion of each dataset with gender and the proportion of the gendered data of each category: EG equal gender, PG positive gender, SPG strongly positive gender.

Dataset	NG (%)	Gender data	EG (%)	—Percentage of gender identified data—			
				Female		Male	
				PG (%)	SPG (%)	PG (%)	SPG (%)
Hate Speech	89.0	1758 (11.0%)	4.6	0.7	49.9	0.6	44.3
Abusive Tweets	88.1	11914 (11.9%)	3.1	0.8	34.5	1.0	60.6
Hotel Review	96.4	18771 (3.6%)	3.8	1.1	41.6	1.3	52.2

4 Evaluation

The aim of the evaluation is to measure gender bias using our ITS approach for creating test instances identified with gender which are necessary for measuring the difference in task performance across genders. We compare this with using the synthetic test instances generated using the identity term template approach.

The evaluation uses the text classification tasks of abusive content detection on the Hate Speech and Abusive Tweets datasets described in Sect. 3 above. We also include an evaluation on the Hotel Review dataset where the expectation of gender bias in the data is less. Due to the highly imbalanced class distribution in the Hotel Review dataset, a subset of the data was sampled. A equal distribution of both classes that did not have the gender identified was sampled from the dataset in the dataset and this was added to the test data to give a subset of just under 60,000 instances with a class distribution of 63%/37% for “satisfied”/“unsatisfied”.

For classification a simpler version of the deep neural network model used by [5] is used. Our model consists of an embedding layer as an input layer using Word2Vec embedding, followed by a bidirectional long short term memory (BiLSTM) layer to encode the input sentence, a dropout layer and a linear output layer with cross entropy to compute the loss which comes after a dense layer with Relu activation. All hyper parameters were tuned on a 20% split of the training data.

Gender bias is measured using the test instances that have gender identified. As can be seen from Table 2 the number of examples with gender identified in the datasets is limited. As we need gendered examples in both training and test splits, we split the datasets 70%/30% into stratified training and holdout testing splits. The full holdout test split is used to measure overall task performance.

Table 3. The templates, identity term pairs and the nouns & attributes

(a)

Templates
You are a <adjective> <identity term>
<verb> <identity term>
Being a <identity term> is <adjective>
I am a <identity term>
I hate <identity term (plural form)>

(b)

Identity Pairs
woman/man girl/boy
female/male she/he
daughter/son her/his
herself/himself gal/guy
mother/father Mary/John

(c)

Nouns & Adjectives & Verbs	Target
disgusting, filthy, nasty, rotten, horrible, terrible awful, stupid, moronic, dumb, ugly, repulsive, vile idiotic, shitty, fucked, kill, murder, hate, destroy	Abusive
great, fun, nice, neat, happy, good, best, fantastic wonderful, lovely, excellent, incredible, friendly gracious, kind, caring, hug, like, love, respect	Non-Abusive

For our proposed ITS approach the test split is used to measure gender bias. We repeat this process five times and report average performance measures across the five holdout splits.

For the identity term template approach the test instances which are generated from the templates are used to measure gender bias. Following work by [14] the templates we used are given in Table 3a. Table 3b lists the identity term pairs we used to give sets of paired gendered test data. These pairs are the same as those we used for the identification of gender in our ITS approach. Table 3c shows the nouns and adjectives used to fill the templates.

The identity term template approach generated 1480 synthetic test samples in total, 740 pairs with equal sets of male and female instances and equal distribution across the “abusive” and “non-abusive” classes. The distribution of the test instances for our ITS approach varied slightly for each holdout split. Table 4 shows the percentage of the dataset that was used as test data and the female and male distribution of the test data per class for both ITS and identity term template approach across the three datasets. This shows that the amount of gendered test data varies regardless of approach while the template approach generates a standard set of synthetic test data.

5 Results and Discussion

Task performance is measured using average class accuracy due to the imbalanced class distributions in all datasets as evident in Table 1. We measure gender bias using True Positive Rate Gap (TPR_{gap}) [16] which is an equality of oppor-

Table 4. Percentage of the dataset used as gendered (G) test data and the distribution of gendered test data for Identity Term Sampling (ITS) and Identity Term Template (ITT) across the five holdout splits

Dataset	Class	Identity term sampling			Identity term template		
		G (%)	F (%)	M (%)	G (%)	F (%)	M (%)
Hate speech	Abusive	$1.6 \pm 4 \times 10^{-4}$	$69 \pm 3 \times 10^{-2}$	$31 \pm 3 \times 10^{-2}$	4.6	50	50
	Non abusive	$1.5 \pm 3 \times 10^{-4}$	36 ± 10^{-2}	64 ± 10^{-2}	4.6	50	50
Abusive tweets	Abusive	$1.5 \pm 4 \times 10^{-4}$	41 ± 10^{-2}	59 ± 10^{-2}	0.7	50	50
	Non abusive	$2.0 \pm 3 \times 10^{-4}$	33 ± 10^{-4}	67 ± 10^{-4}	0.7	50	50
Hotel review	Unsatisfied	0.1 ± 10^{-4}	$40 \pm 5 \times 10^{-3}$	$60 \pm 7 \times 10^{-3}$	—	—	—
	Satisfied	0.9 ± 10^{-4}	$45 \pm 9 \times 10^{-3}$	$55 \pm 8 \times 10^{-3}$	—	—	—

tunity measure and measures the differences in the gender specific true positive rates and is defined in Eq. 1.

$$TPR_{gap} = |TPR_{male} - TPR_{female}| \quad (1)$$

The results of measuring gender bias using both the identity term template approach (labelled ITT) and our new Identity Term Sampling (ITS) approach for the Hate Speech and Abusive Tweets datasets are displayed in Figs. 1a to 1c. Each figure gives results for a single dataset and the left hand y-axis is classification performance and the right hand axis is the TPR_{gap} which reflects the gender bias. Each figure gives the performance on the test data for each class and for each gender. The True Positive Gap TPR_{gap} for each class is also displayed on the graph.

Across the Hate Speech and Abusive Tweets datasets (Figs. 1a & 1b) where some level of gender bias may be expected, the TPR_{gap} is higher for our proposed ITS method than for the template method. It is significantly higher in the Hate Speech dataset. This shows that our proposed method is identifying gender bias at least as well as the template approach which uses synthetic data. It also suggests that the use of test data that is aligned with the original data as it is extracted from it, may be a more confident way of identifying gender bias in the data.

Looking at the gender level classification results on both datasets to identify where this gap comes from, the pattern is the same across both datasets. The accuracy on the female data is lower than the male data for the “non abusive” class. This indicates that examples of non-abusive content that are identified as female (i.e. more likely to be about women) are classified incorrectly as “abusive” more often than examples of non-abusive content that are identified as male, i.e. about men. And the reverse happens in the “abusive” class, examples of abusive content that are identified as female are more often classified correctly as “abusive” than examples of abusive content that are identified as male. This suggests that the model built on this training data is demonstrating gender bias by treating gender differently. This pattern is extremely evident in the Hate Speech dataset.

Table 5. Accuracy per class, average class accuracy (ACA) on the gendered test data for identity term template (ITT) & ITS approaches and ACA for each dataset.

Dataset	Class	Class accuracy		Gender identified test ACA		Overall testset ACA (%)
		ITS (%)	ITT (%)	ITS (%)	ITT (%)	
Hate speech	Abusive	79.8 ± 0.06	40.0 ± 0.14	78.9 ± 0.08	64.6 ± 0.11	82.2 ± 0.02
	Non abusive	78.0 ± 0.09	89.3 ± 0.09			
Abusive tweets	Abusive	84.2 ± 0.01	39.0 ± 0.23	88.4 ± 0.01	65.3 ± 0.17	91.4 ± 0.007
	Non abusive	92.5 ± 0.02	97.6 ± 0.03			
Hotel review	Unsatisfied	64.8 ± 0.16	–	75.7 ± 0.05	–	84.4 ± 0.06
	Satisfied	86.6 ± 0.06	–			

Figure 1c shows the results of the Hotel Review dataset. As it is difficult to generate appropriate identity term templates that will be adequately representative for this domain, we do not include figures for the identity term template approach. As can be seen from the figure, the ITS gender gap for this dataset is very small. This is not surprising as we would not generally expect there to be significant gender bias in user generated hotel reviews. However, it is worth noting that the ITS TPR_{gap} for the “unsatisfied” class in the Hotel Reviews is higher than the TPR_{gap} for the template based approach for the “non abusive” class in the Abusive Tweets dataset. This suggests that there may be some element of gender bias in this dataset, specifically in the “unsatisfied” class - the pattern is similar to that identified in the other two datasets. The examples of “unsatisfied” content which are identified as female (i.e. about women) are more slightly more often classified correctly as “unsatisfied” than reviews that are identified as male (i.e. about men).

The classification results on the holdout test data and on the gendered test data for each dataset across the five holdout splits is shown in Table 5. The last column in the table shows the average class accuracy (ACA) on the full test data, averaged with the standard deviation across all five holdout splits. This shows how well the model can perform at the task of abusive content prediction with the ACA on the Abusive Tweets dataset higher at 91% than the Hate Speech at 82%. The gender-identified ACA columns show the performance of the model on just the test data with gender identified for both the ITS and identify term template (ITT) approaches. Across the two abusive content datasets the proposed ITS approach achieves significantly better performance on the gendered test data than the template approach. This is not surprising as the ITS test data is sampled directly from the original training data. However, this suggests that the templates used to measure gender bias are not reflective of the data as the model is unable to classify them well. The class accuracy columns in the table show the average class accuracy with standard deviation results for the test data with gender identified. In both abusive content datasets, the ITT approach has a very poor classification performance on the abusive class with less than 50% accuracy in both cases and a high standard deviation, suggesting that the template sentences generated for the abusive content do not reflect at all the actual abusive content in the datasets. The use of the original data which

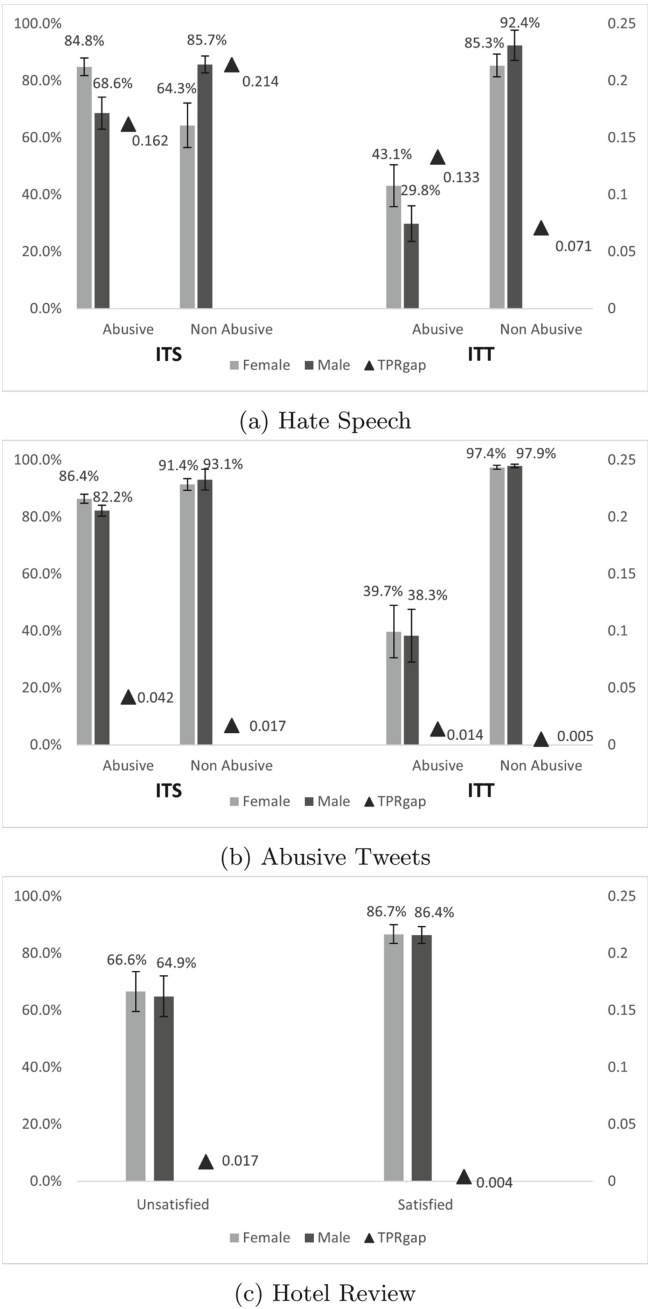


Fig. 1. Accuracy and TPR_{gap} for Identity Term Sampling (ITS) and Identity Term Template (ITT)

the proposed ITS approach achieves a significantly better performance on the abusive class suggesting better test data.

6 Conclusions and Future Work

In this work, we propose an Identity Term Sampling technique to overcome one of the challenges faced in evaluating bias in training data which is the absence of gender in existing datasets. The proposed method addresses the challenges and the limitations of using GBETs by automatically identifying gender for some instances in a dataset and using these to evaluate the gender bias. We evaluated the performance of ITS on an abusive content classification task using datasets which are likely to contain gender bias and a sentiment analysis task using a dataset which is less likely to contain gender bias.

Our experiment results show ITS can identify gender bias at least as well as existing template based approaches. Classification results on the gendered test data used to measure gender bias show that template based approaches do not generate test data that is appropriate for the task at hand while ITS uses test data that is better aligned to the task. While the gender identification performed in this work might be considered naive, we suggest that this approach has some promise as a more confident mechanism of measuring gender bias through automatic identification of gender. Future work will consider including more focused methods of identifying gender in text instances.

Although ITS has shown promising results in this work, it should be mentioned it might be challenging to use ITS on some types of natural language datasets. User generated content including movie and book reviews potentially can contain a wide range of gender identity words and it may be challenging to identify a single gender. More focus on refining the identification of gender in ITS may help in this respect.

Our evaluation of ITS focused on using the dataset itself for the evaluation of gender bias without applying any data augmentation techniques often used in this domain. In future work we will consider the impact of applying gender swapping as a data augmentation technique on the test instances that are generated by the ITS approach giving additional test data and equal distribution of test data. Finally, it has been observed that a wide range of research into gender bias predominantly focuses on distinguishing two genders, male and female, neglecting the fluidity and continuity of gender as a variable [20]. Future work will consider extending the ITS approach to non binary genders and also include gender-neutral linguistic norms such as ‘they’ in English.

Acknowledgements. This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

1. Basta, C., et al.: Evaluating the underlying gender bias in contextualized word embeddings. In: Proceedings of the 1st Workshop on Gender Bias in NLP. ACL (2019)
2. Bolukbasi, T., et al.: Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In: Advances in NeurIPS (2016)
3. Caliskan, A., Bryson, J.J., Narayanan, A.: Semantics derived automatically from language corpora contain human-like biases. *Science* **356**(6334), 183–186 (2017)
4. Davidson, T., et al.: Racial bias in hate speech and abusive language detection datasets. In: Proceedings of the 3rd Workshop on Abusive Language Online. ACL (2019)
5. De-Arteaga, M., et al.: Bias in bios: a case study of semantic representation bias in a high-stakes setting. In: Proceedings of FAT* (2019)
6. Dixon, L., et al.: Measuring and mitigating unintended bias in text classification. In: Proceedings of the 2018 AAAI/ACM Conference on AIES, AIES 2018. ACM (2018)
7. Emami, A., et al.: The KnowRef coreference corpus: removing gender and number cues for difficult pronominal anaphora resolution. In: Proceedings of ACL (2019)
8. Founta, A.M., et al.: Large scale crowdsourcing and characterization of twitter abusive behavior. In: Twelfth International AAAI Conference on Web and Social Media (2018)
9. Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. *Adv. Neural. Inf. Process. Syst.* **29**, 3315–3323 (2016)
10. Kiritchenko, S., Mohammad, S.: Examining gender and race bias in 200 sentiment analysis systems. In: Proceedings of Conference on Lexical & Computational Semantics (2018)
11. Lu, K., et al.: Logic, Language, and Security: Essays Dedicated to Andre Scedrov on the Occasion of His 65th Birthday. Springer, Heidelberg (2020)
12. Nadeem, M., Bethke, A., Reddy, S.: StereoSet: measuring stereotypical bias in pretrained language models. In: Proceedings of ACL and the 11th IJCNLP). ACL (2021)
13. Nangia, N., Vania, C., Bhalerao, R., Bowman, S.R.: CrowS-pairs: a challenge dataset for measuring social biases in masked language models. In: EMNLP (2020)
14. Park, J.H., Shin, J., Fung, P.: Reducing gender bias in abusive language detection. In: Proceedings of EMNLP. ACL (2018)
15. Peters, M.E., et al.: Deep contextualized word representations. In: Proceedings of the NAACL. ACL (2018)
16. Prost, F., Thain, N., Bolukbasi, T.: Debiasing embeddings for reduced gender bias in text classification. In: Proceedings of the 1st Workshop on Gender Bias in NLP (2019)
17. Rudinger, R., et al.: Social bias in elicited natural language inferences. In: Proceedings of the First ACL Workshop on Ethics in NLP. ACLs (2017)
18. Savoldi, B., Gaido, M., Bentivogli, L., Negri, M., Turchi, M.: Gender bias in machine translation. *Trans. ACL* **9**, 845–874 (2021)
19. Smith, E.M., et al.: “I’m sorry to hear that”: finding bias in language models with a holistic descriptor dataset. arXiv preprint [arXiv:2205.09209](https://arxiv.org/abs/2205.09209) (2022)
20. Stanczak, K., Augenstein, I.: A survey on gender bias in natural language processing. arXiv preprint [arXiv:2112.14168](https://arxiv.org/abs/2112.14168) (2021)

21. Sun, T., et al.: Mitigating gender bias in natural language processing: literature review. In: Proceedings of the ACL. ACL (2019)
22. Waseem, Z., Hovy, D.: Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In: Proceedings NAACL Student Workshop (2016)
23. Webster, K., Recasens, M., Axelrod, V., Baldridge, J.: Mind the GAP: a balanced corpus of gendered ambiguous pronouns. *Trans. ACL* **6**, 605–617 (2018)
24. Zhao, J., Wang, T., Yatskar, M., Cotterell, R., Ordonez, V., Chang, K.W.: Gender bias in contextualized word embeddings. In: Proceedings of the NAACL. ACL (2019)
25. Zhao, J., Wang, T., Yatskar, M., Ordonez, V., Chang, K.W.: Gender bias in coreference resolution: Evaluation and debiasing methods. In: Proceedings of the NAACL (2018)
26. Zhao, J., et al.: Men also like shopping: reducing gender bias amplification using corpus-level constraints. In: Proceedings of the EMNLP (2017)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





How Augmented Reality Beauty Filters Can Affect Self-perception

Clara Isakowitsch^(✉)

University College Dublin, Belfield, Dublin 4, Ireland
claraisakowitsch@ucdconnect.ie

Abstract. Augmented reality is used on visual social media platforms such as Snapchat and Instagram with filters that can be applied to the user's face. These filters detect and transform facial features by overlaying digital masks on moving faces. Augmented reality beauty filters (ARB filters) alter the appearance of the face by conforming it to current beauty ideals. Prior to the development of ARB filters selfies could only be enhanced by retroactive photo editing. However, ARB filters adapt to facial features in real time, resulting in a unique digital beautifying process. This qualitative study explores how the use of ARB filters impacts people's perceptions of themselves. It is based on online interviews that were conducted with eight individuals. The results are analysed within the frameworks of Extended Mind Theory and Enactivism and indicate that ARB filters may have a greater impact on people's self-perception than retroactive photo editing.

Keywords: Augmented reality · Beauty filters · Self-perception · Social media · Extended mind theory · AI technologies

1 Introduction

Selfies play a major role in self-presentation on social media sites. Visual social media platforms, such as Instagram and Snapchat, have grown in popularity over the past fifteen years especially among 15–25 year old members of generation Z. Since Instagram and Snapchat introduced augmented reality (AR) filters in 2015, these filters have become a popular widespread feature for taking selfies. 600 million people use them each month on Instagram or Facebook, 76% of Snapchat users apply them every day [9]. Some of these filters are silly, such as cat ears or fantasy characters, while others allow users to digitally alter their face to conform to specific beauty standards. These augmented reality beauty filters (ARB filters) do not only apply digital makeup, but they go beyond that by deforming the user's jaw and nose, expanding their eyes and lips, and smoothing their skin [2]. Prior to the development of ARB filter, selfies could only be enhanced by retroactive photo editing. Now, facial enhancement is achieved via AR filters that adapt to facial features in real time. This takes the interaction with the beautified self to a different level. While taking a selfie, the user moves and their

beautified self moves with them. There seems to be a gap in the literature when it comes to the specific effects of the process of taking a selfie with AR beauty filters in contrast to retroactively editing a static image. This study aims to shed light on these differences and the effects of AR-beautification on self-perception.

2 Background

2.1 Augmented Reality Beauty Filters

Augmented reality is a technology that combines reality and digital information by overlaying digital content on objects, humans or places in the real world. ARB filters are automated photo editing tools using artificial intelligence and computer vision to detect facial features and modify them [11]. Beauty filters are developed by individual creators as well as cosmetic brands and retailers. While some beauty filters are merely for applying make-up, the majority of beauty filters alter the contours and shapes of the face. Most beauty filters modify their users' facial features in a similar way. It's what the journalist Jia Tolentino points out as the *Instagram Face* characterized by poreless skin, high cheekbones with catlike eyes, a slim nose and full lips. It is a face that is "distinctly white but ambiguously ethnic" [14].

A study by Rosalind Gill, a professor of Social and Cultural Analysis, conducted in the United Kingdom in 2021, found that 90% of young women either apply beauty filters or edit their photos before posting them on social media. According to Gill, beauty filters and photo editing contribute to a society where young people constantly feel scrutinised and judged by their peers [7]. The way AR technology is used in social media raises a number of other ethical concerns, ranging from the promotion of plastic surgery to facial recognition to questions about the nature of AR. Behr et al. [2005] [3] formulated the following four risks of XR (virtual and augmented reality) technology: (i) motion sickness; (ii) information overload; (iii) intensification of experience and (iv) cognitive, emotional and behavioral disturbances after re-entering the real world after the XR experience. XR technology is highly persuasive, and that is where the risks of ARB filters lie. Since the "reality" of the face is experienced through the device, virtual aspects may become indistinguishable from the real. For vulnerable groups such as children and adolescents, discriminating between the real and the virtual can pose an even greater challenge. The embodied experience of virtually altering one's face can create confusion in people about their real bodies and lead to a kind of body dysmorphia [13]. Finally, the immersive nature of AR technology could change perceptions of the self in a different way than normal photo manipulation.

2.2 Effects on Self-perception

Motivations. The motivations for self-presentation on visual social media platforms such as Instagram, Snapchat or TikTok are multi-faceted. They are associated not only with the drive to present the ideal self, but also with presenting

the true self and transforming the self [9]. Those who use AR filters to depict an idealised version of the self often want to come across as more beautiful or *cool*. This fits with the user behaviour of many, where only content that shows the best moments or looks is uploaded [12]. For those wanting to present their true self, AR filters can be tools to experiment with different aspects of the self. Despite the digital modification, representation through AR filters can be congruent with the user's true self. Finally, filters can be used to transform and discover new aspects of the self. This allows individuals to engage in cognitive processes (thinking, hoping or fearing) about who they could be [9]. When users engage with filters for entertainment and fun it is often associated with escapism. Users may feel like they are mentally transporting themselves to imaginary worlds. According to Javornik et al. [2022] [9] the impact of AR filters on mental health and well-being depends on the underlying motivations for their use. Those who use AR filters as a form of entertainment, social interaction, or as means to transform their digital selves often benefit from the technology, and a positive effect on the mood can be observed. However, using AR beauty filters to idealize the self and fake one's image often leads to lower self-acceptance and has negative effects on users' self-perceptions.

Self-discrepancy. When we look at ourselves through the lens of our phone camera or in the mirror, we may see the same image. Yet AR technology allows us to overlay a filter on the image on our phone in real time. This creates an obvious discrepancy with what we see when we look in the mirror, where no such filters can be applied. The negative effects resulting from this experience can be explained with the self-discrepancy theory (SDT) by Higgins [1987] [8]. According to Higgins's theory there are three domains of the self: a) the *actual* self, a representation of the attributes a person thinks they possess; b) the *ideal* self, representing the attributes that someone would like to possess and c) the *ought* self which represents the attributes that someone believes they should possess. A perceived discrepancy between the actual and the ideal self can lead to feelings of disappointment and dissatisfaction as one feels that one cannot live up to one's own beauty aspirations. Discrepancies between the actual and the ought self can also result in negative emotions and in the fear of negative outcomes. According to SDT an individual is motivated to align their actual self with the ideal self and the ought self. In their work on the effects of beauty filters on the self-image of Saudi women Alsaggaf [2021] [1] stated that the use of beauty filters on Snapchat leads to a contradiction the user feels when seeing the actual image compared to the ideal image using Snapchat beauty filters. The negative emotions following this perceived contradiction align with SDT.

Extended Mind Theory. One risk associated with AR is its intensity, which can make it difficult for users to distinguish between the real and the virtual world. Using Clark and Chalmers's Extended Mind Theory the human and the AR application could be considered a coupled system. The distinction between the real and virtual world becomes difficult because the virtual extends the boundaries of the mind, rather than being external to it. In his book "Reality+" Chalmers [2022] [4] argues that AR affords us the opportunity to expand and

augment our minds. AR glasses that project virtual items into the actual world to aid in navigation augment both the physical world and the mind by enhancing mental capacities [4]. Similarly ARB filters may extend our mind by allowing us to visualise what a “better” version of us might look like.

3 Method

3.1 Participants

8 people participated of whom 4 identified as female, 2 as non-binary and 2 as male. Their age varied between 24 and 33. All participants were White and of either German or U.S. origin. Convenience sampling was used for recruiting the participants.

3.2 Materials

Each participant used a computer that had Zoom installed. Additionally the participants used their smartphone and the Instagram app that was installed on it. Six participants interacted with a filter called “Mary Phillips” created by Sophie Katirai that adapts the face to female standards of beauty. Two participants interacted with a filter for male beauty standards called “Men’s Beauty” created by Jason Emer. These two filters were chosen because of their popularity. Both creators are famous *influencers* whose filters are frequently used.

3.3 Procedure

At the beginning of the Zoom interview participants were asked to hide their own self view on Zoom, so that their own reflection on Zoom wouldn’t distract them. They were then asked to have their Instagram app with the filter ready and also have their front camera open, so that they could switch between the two applications on their phone. There were two parts in this experiment. In the first part participants were asked to apply the filter and take a frontal picture of themselves without making any movements. They were then asked to look at this static beautified image. In the second part participants were asked to apply the filter and move around and look at themselves from all kinds of angles. After and before each part of interacting with the static or moving beautified self they were asked to switch back to their front camera. In the second part when they moved with the filter participants were asked to take a selfie with the filter. Later they were asked to take a selfie without any filter. Participants were asked to indicate when they started and finished with taking a selfie. That way a time stamp could later be identified in the transcript.

3.4 Interview Questions

A semi-structured interview was conducted with most questions belonging to one of the three categories: a) the emotional experience, b) self-perception and c) selfie behaviour. Questions were asked during and after the interaction with the ARB filter. Category a) includes, for example, questions like: “How do you feel about yourself when you look at yourself with the filter?”, “What kind of emotions did you experience in the moment when you switched from the filtered version of yourself to the no-filter version?” Questions belonging to category b) comprise “How much do you identify with this filtered version of yourself?”, “Which facial features are modified by the filter?”, “Is there anything about your face you notice in a different way now?” Category c) includes: “What kind of movements do you do when taking a selfie?”, “Were the movements you did when you were taking a selfie the same with and without the filter?” Furthermore participants were asked whether they experienced a difference between interacting with the static or moving beautified self.

4 Analysis

(See Fig. 1).



Fig. 1. Left with the ARB filter, right without a filter.

4.1 Self-perception While Using the Filter

Feeling Disconnected. All participants described their experience of interacting with the filter, either static or moving as somehow fake. Four participants used the word “artificial” to describe their beautified self. Often the artificiality was the reason why participants did not like the filter or could not identify with the filtered version of themselves. One participant described her experience like that: “I definitely feel like it makes my face look so much more artificial. It makes me feel like I’ve gotten plastic surgery”. Most participants described a feeling of disconnection when looking at themselves with the filter. Two participants used the word “alienating”. One participant even experienced anxiety: “Seeing myself in that altered, and augmented way kind of gives me anxiety. It’s just weird because I never use these filters and now I see it as like: ‘I actually could even look better’ kind of. I don’t know if this is necessarily better. But just the possibility of it kind of triggers a weird feeling, a kind of anxiety maybe”. Five participants said that they feel like a different person with the filter. One participant said “I feel like I’m looking at a Disney movie version of myself”. Another participant said: “It just kind of reminds me of a generic Instagram model face. It’s just sort of a very trendy cat eye, blush, makeup, kind of person”. A third participant described her experience like that: “I think I less identify with myself and more with like Instagramers you know like influencers”. Two other participants attributed this different person that they saw even with a different kind of personality. One said “I don’t think anyone looks like that. Or if they really did I feel like I wouldn’t like them”. Another participant articulated “If I would look at a picture like this. I wouldn’t think it’s a sympathetic person. It’s very artificial”.

Positive/Negative Aspects of the Filter. All participants liked certain aspects of the filter. For some it was the makeup, for others it was the enhancement of the lips or the smoothing of the skin. Most participants liked that the dark circles and wrinkles under their eyes disappeared, making them look more fresh. Two female participants noted that the filter made them look more young, one said: “With the filter when I move I actually feel very cute. I think my lips are very big and the head is small. It looks really a bit like a child”. Nearly all participants reported that the filter might “objectively” make them more pretty. Overall however, all of them rejected the idea that the filter made their face look more beautiful than in real life. One participant said: “Its kind of like, maybe adapting my face to beauty standards or like a perception of how male beauty works that I’m not subscribing to”. One participant noted that she did not like the way the filter shapes the eye into a cat eye form: “I know that is actually from like a racist idea of beautifying Asian features, so I don’t like that”. Two participants complained that the filter erased their freckles, a feature they both like about their face. Most participants found the filter too strong and noted that they would like it more if it was more subtle.

Getting Used to the Filter. Most participants described that the longer they looked at themselves with the filter (either moving or static) the more real it felt

since they got used to it. One participant said: “Like the first second I was like okay that’s really not me, but the longer I look at it I would say it looks kind of similar to me”. Another participant said: “The longer I stare at it the less absurd it looks. And the more I could see like: ‘okay yeah I can look like that’ or it doesn’t look that unnatural”.

4.2 Emotional Experience After Using the Filter

After Interacting with the Static Beautified Selfie. When going back to the unfiltered front camera three people described their emotional response as more positive than negative. Looking at themselves again without a filter made them realise that they do actually like their normal face more than the filtered version. One participant said: “My first reaction was that I’m surprised that I don’t look as bad as I thought I did. I don’t see my flaws or they’re not glaring out”. Three participants experienced negative emotions when they switched back to their front camera. One participant said: “I sense some sort of downgrade.”, another said: “Going from a beauty filter that has a lot of makeup to a no makeup face, that’s a little jarring”. Most participants pointed out that certain facial features became more visible to them after looking at the beautified selfie. One participant said: “I do see things that I didn’t see before like I realise more now what the differences are. Like my wrinkles around my eyes, and maybe my nose and my lips. I don’t feel like before, I do feel like it’s changing that. Because before I felt pretty positive about myself”.

After Interacting with the Filter While Moving. Seven out of the eight participants described that they were experiencing a negative emotional reaction in the moment when they switched back to their front camera. Two participants described their emotional response with a feeling in their heart. One said: “I do felt like some heaviness in my heart somehow. Yeah, it’s this kind of unsettling feeling. Even though I’m not sad, but there’s this kind of anxious feeling for sure.”, the other participant said: “I definitely kind of felt like a heartdrop. Like: ‘this is what I look like’ and then having to really quickly smile and remind myself that I look great regardless. The more I looked at the filter the more I got used to seeing it and switching back I had to very quickly go through those emotions”. Again most participants seemed to notice certain facial features more than before applying the filter. One participant described it like that: “I can definitely notice how different my jaw line is between the filter and no filter. And that doesn’t make me feel great. And so looking at my face now, I just noticed my jaw, a lot more. I look so much more round without the filter”. Three participants stated that they were disappointed when they saw their face without a filter again. They used expressions like “downgrade” and feeling “underwhelmed”, “disappointed” and “less enthusiastic” about their physical appearance. One participant said: “I feel like a weight on my body. I feel more weak now”.

4.3 Selfie-behaviour

One objective of this study was to explore whether people behaved differently when taking selfies with and without the ARB filter. The transcript disclosed the time it has taken participants to take a selfie. The time it took them to take a selfie with the filter ranged between 3 and 10 s, whereas the time it took to take a selfie without the filter ranged between 5 and 20 s. Six of the eight participants took less time with the filter than without. Half of the participants indicated that taking a selfie with an ARB filter is easier than taking a normal selfie, because the filter allows them to take the selfie from a greater variety of different angles. One participant said: “I think the filter is a little bit more forgiving, in the sense that there are more angles where my face looks good. So, I can use maybe an angle that I normally wouldn’t go for on a selfie”. Three participants said that it didn’t make a difference for them in their selfie behaviour whether they applied a filter or not. One person said that she felt uncomfortable being watched while taking a selfie: “I feel like my selfie taking behavior is really influenced by the fact that I’m being watched. I was really like: ‘okay I really want to get out of the situation’. Otherwise maybe I would have taken more time”. One male participant reported that the filter did not always settle properly on his face. He said, that this is why it took him longer to take a selfie with the filter, since he needed to first find an angle where the filter is actually working.

4.4 Difference Between Static and Moving

Technology Failure. Most ARB filters on Instagram adapt the user’s face to female standards of beauty. In comparison there are only very few filters for male beauty. The filter “Men’s Beauty” used in this study worked a lot less well than the female beauty filter. One male participants who used the filter reported that the filter does not adapt quickly enough to his movements, resulting in a strange feeling and showing him that the filter is not real. He described his experience the following way: “I think it’s just like, funny somehow. And it’s ridiculous too. So, as I see this filter applying and then not applying or not being like set the way it has to be on my face I identify less with it. It’s like when I’m moving and then my eyebrows are kind of in my eyes. So that makes me realise it’s a filter and it’s not real”. This participant reported that looking at the static beautified image had a stronger effect on his self-perception than moving with the filter, due to this technology failure. The other male participant decreased the intensity of the filter during the experiment which improved the functionality and prevented the filter from not properly settling on his face. This aspect of technology failure did not occur to the same extend with the female and non-binary participants who used the female beauty filter. However, one female participant had a similar feeling of confusion when one time the filter did not set properly: “Just one thing, which is funny because I looked at myself and then I took a sip of coffee, and then it didn’t work anymore. So I was like really confused for one second, I was like: ‘oh I look nice’ and then suddenly my eyes looked like they normally do”.

Movements. Most participants indicated that the interaction with both digital beautifications (static and moving) affected them in some way. One participant said: “I think both are kind of jarring, and I think both change your perception of how you look”. Six of the eight participants found the experience of moving with the filter more intense than looking at the static beautified selfie. They noted that the beautification felt more real when they saw the filter adapt to their movements. One participant said: “When I move, the filter feels more natural. Maybe because movements are part of your identity and you can recognise yourself better”. Another participant regarded the filter more as a part of their face when they were moving: “When I was moving, I had more of a chance to think about the filter more. Before when it was static it was just like: ‘okay this is my face and this is the filter’ and it didn’t feel very meaningful that I was using the filter. It didn’t feel like it impacted me but then when I was moving around it felt more like it was a part of my face”. For another participant it was the possibility to see herself from different perspectives that made a difference: “I feel like being able to move from side to side made it look more like myself. I think it was especially visible from the front view that it doesn’t look like myself. The moving aspect made it look a bit more like me and I could identify more with it. Whereas with the frontal picture, I only had this one perspective and I was like okay this looks very different. But when I can move I do identify more with it for example when I’m seeing my nose from the side it looks more like my real nose”. Another participant also described the experience of moving with the filter as more intense yet she felt less like herself: “I think when I just used the filter while moving, I was kind of pretending to be someone”.

Time Spent Interacting with the Filter. All participants spent more time looking at themselves while moving than looking at the static beautified selfie. This time difference was not intended in the design of the experiment but more a result of how the participants answered the questions. All participants answered in more detail when they were moving around, it seemed like as they moved, there were more things they noticed about the filter. Moreover, they sometimes took longer to answer as they were moving and talking at the same time and sometimes seemed distracted by that. Participants spend on average 3.3 min looking at the beautified selfie and 6.2 min moving with the filter. Two participants noted that this time difference changed the impact the filter had on them. One of them said: “I think the only difference was just the length of the two things. I looked longer at myself while moving. The longer that I was interacting with the filter, the more I kind of got used to it and could accept it as myself. And I think that probably had a negative effect on me changing back to what I actually look like”. This time difference in the experiment must be considered as a potential factor that influenced the different experience people had when interacting with a static selfie vs. moving with the filter. However, one could argue that in real life people normally spent more time interacting with ARB filters on social media (since they spent some time moving with the filter before taking a selfie) than the time they spent looking at an already enhanced static selfie.

5 Discussion

5.1 Extended Mind and Self-perception

When switching back to their unfiltered front camera, many participants had negative emotions about their physical appearance. The interaction with the beauty filter made certain features of their face become more visible to them. Some participants were more aware of wrinkles, skin irritations, or the roundness of their faces - features that are usually not associated with beauty ideals. Their self-perception seemed to be affected in the sense that particular facial features stood out more to them after viewing their enhanced selves.

Six participants said that moving while using the ARB filter was a more intense experience than interacting with a static selfie. This aligns with theoretical considerations of extended mind theory. Applying Clark's and Chalmer's theory, the human and the ARB filter could be considered as a coupled system that extends the mind's boundaries. This may result in an inability to distinguish between the actual and beautified enhanced self. ARB filters could serve as visualisation aids for the mental image of the ideal or ought self. Herein lies the distinction: the self and a manipulated static image of the self would not be considered a coupled system as there is no interaction between an image and the self. The cognitive process of imagining how one would look if they resembled the manipulated selfie would still need to be carried out by the individual. The augmented self however, seems real because it is not static, but reflects people's movements. The AR beauty application and the self interact, facilitate a cognitive process and therefore become a coupled system. These considerations and the results of this study led me to hypothesise that AR facilitates a digital beautification process that could potentially have a greater impact on people's self-perception than retroactive photo editing.

5.2 Self-discrepancy

The negative emotions the participants experienced when they switched back to their normal front camera could be a result of an increased perceived discrepancy between the ought self and the actual self. All participants disliked most aspects of the filter although it might adapt their face to certain beauty ideals. They rejected the idea that this beautified self improves their natural beauty. However, most participants did experience negative emotions after seeing their enhanced self. I assert that internalised beauty ideals of how one *should* look like influenced the participants more than their own ideal self. Higgin's self-discrepancy theory discriminates between ideal-actual and ought-actual self-discrepancies. According to the participants statements I argue that participants' experiences can at best be described in terms of perceived ought-actual self-discrepancies. Five individuals reported seeing a different person when interacting with the filter. Surprisingly, two people assigned this different person unsympathetic character traits. According to Javornik's studies on why individuals use filters, the participants may perceive the filter as a mask or costume that mentally transfers them

to another world. Attributing their augmented self with different character traits could indicate that these participants identified less with the filtered version of themselves.

5.3 Sense-Making

Sense-making that happens through movements and interactions with the environment is often described in enactive and embodied cognitive science. I particularly found the statement of one participant interesting in that context: “When I move, the filter feels more natural. Maybe because movements are part of your identity and you can recognise yourself better”. The ability to move and interact with the filter seemed to assist her in recognising herself in the augmented version of herself. An analogy to the concept of *dynamic touch* first coined in ecological psychology by Gibson [1992] may be made here. Instead of mentally calculating the weight of an object by looking at it, dynamic touch describes an exploration with the hands. The work of the muscles that causes pressure and deformation to muscles and tendons offers information about the weight of the object that becomes available at the wrists [5, p. 105–135]. In enactivist terms the concept of dynamic touch aligns with sense-making activities [15]. Similarly vision is described by Alva Noë as a process that depends on interactions between the perceiver and the environment and involves movements [10]. Following this argument I claim that sense-making of the digital self can only happen when one moves and interacts with the ARB filter. This gives support to my hypothesis that AR as a technology could have a greater impact on self-perception than photo editing.

6 Conclusion

Overall this research adds to existing literature exploring the impact of beauty filters on users’ self-perception and body image. It calls into question current digital beautifying procedures by focusing on augmented reality that adapts to movements in real-time. The purpose of this study was to investigate how ARB filters influence people’s self-perception. Participants were first instructed to engage with a static enhanced selfie during the experiment. They were asked to look at themselves and move with the ARB filter in the second part. I was curious to see if the two components of this experiment had distinct effects on people. The majority of participants stated that moving with the filter affected them more than looking at the static enhanced selfie. It is not clear however if this effect could be observed due to the fact that people spend more time moving with the filter than looking at the enhanced selfie. Future studies could investigate the hypothesis that ARB filters have a greater impact on users than retroactive photo enhancement. Questions about how movements manifest feelings of identity could be additionally explored.

References

1. Alsaggaf, R.M.: The impact of snapchat beautifying filters on beauty standards and self-image: a self-discrepancy approach. In: *The European Conference on Arts And Humanities 2021* (2021)
2. Barker, J.: Making-up on mobile: the pretty filters and ugly implications of snapchat. *Fashion Style Popular Cult.* **7**(2–3), 207–221 (2020)
3. Behr, K.M., Nosper, A., Klimmt, C., Hartmann, T.: Some practical considerations of ethical issues in VR research. *Presence* **14**(6), 668–676 (2005)
4. Brady, J.C., Gerber, T., Chalmers, D.: The near and far out: virtuality, extended minds, and reality+. *Epoché Mag.* **48** (2022). <https://epochemagazine.org/48/the-near-and-far-out-virtuality-extended-minds-and-reality/>
5. Chemero, A.: *Radical Embodied Cognitive Science*. MIT Press, Cambridge (2011)
6. Clark, A., Chalmers, D.: The extended mind. *Analysis* **58**(1), 7–19 (1998)
7. Gill, R.: *Changing the perfect picture: smartphones, social media and appearance pressures*. City, University of London (2021)
8. Higgins, E.T.: Self-discrepancy: a theory relating self and affect. *Psychol. Rev.* **94**(3), 319 (1987)
9. Javornik, A., et al.: ‘What lies behind the filter?’ Uncovering the motivations for using augmented reality (AR) face filters on social media and their effect on well-being. *Comput. Hum. Behav.* **128** (2022). <https://doi.org/10.1016/j.chb.2021.107126>
10. O’reagan, J.K., Noë, A.: A sensorimotor account of vision and visual consciousness. *Behav. Brain Sci.* **24**(5), 939–973 (2001)
11. Ryan-Mosley, T.: Beauty filters are changing the way young girls see themselves (2021). <https://www.technologyreview.com/2021/04/02/1021635/beauty-filters-young-girls-augmented-reality-social-media/>. Accessed 06 May 2022
12. Shein, E.: Filtering for beauty. *Commun. ACM* **64**(11), 17–19 (2021)
13. Slater, M., et al.: The ethics of realism in virtual and augmented reality. *Front. Virtual Reality* **1**, 1 (2020). <https://doi.org/10.3389/frvir.2020.00001>
14. Tolentino, J.: The age of Instagram face (2019). <https://www.newyorker.com/culture/decade-in-review/the-age-of-instagram-face>. Accessed 09 May 2022
15. Travieso, D., Lobo, L., De Paz, C., Langelaar, T.E., Ibáñez-Gijón, J., Jacobs, D.M.: Dynamic touch as common ground for enactivism and ecological psychology. *Front. Psychol.* **11**, 1257 (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Privacy-Enhanced ZKP-Inspired Framework for Balanced Federated Learning

Stefano Marzo^{1(✉)}, Royston Pinto¹, Lucy McKenna², and Rob Brennan³

¹ Dublin City University, Dublin, Ireland

{stefano.marzo2,royston.pinto2}@mail.dcu.ie

² ADAPT Research Centre, Trinity College Dublin, Dublin, Ireland

lucy.mckenna@adaptcenter.ie

³ ADAPT Research Centre, University College Dublin, Dublin, Ireland

rob.brennan@adaptcenter.ie

Abstract. Federated learning (FL) is a distributed machine learning approach that enables remote devices i.e. workers to collaborate to compute the fitting of a neural network model without sharing their data. While this method is favorable to ensure data privacy, an imbalanced data distribution can introduce unfairness in the model training, causing discriminatory bias towards certain under-represented groups. In this paper, we show that imbalance federated data decreases indexes of equity i.e. differences in treatment for underrepresented classes. To address the problem, we propose a federated learning framework called Z-Fed that 1) balances the training without exchange of privacy protected data using a zero knowledge proof (ZKP) technique, and 2) allows for the collection of information on data distributions based on one or more categorical features to produce metadata about population proportions. The proposed framework infers the precise data distribution without exchanging knowledge of the data categories and uses it to coordinate a balanced training set. Z-Fed aims to mitigate the effect of imbalanced data in FL while respecting privacy and without using mediators or probabilistic approaches. Compared to a non-balanced framework, Z-Fed improves fairness and equality measured in equal opportunities (EPD) by 53.54%, equal odds (EOD) by 56.41%, and statistical parity (SPD) by 46.1% on imbalanced UTK datasets, reducing biased predictions among subgroups. EPD, EOD, and SPD measure the disparity of treatment between privileged e.g. over-represented and non-privileged groups. Given the results obtained, Z-Fed can reduce discriminatory behaviors and enhance trustworthiness of federated learning.

Keywords: Federated learning · Zero knowledge proof · Unbalanced data · Fairness · Privacy · Bias

1 Introduction

Federated learning (FL) is a machine learning (ML) technique that allows an artificial neural network (ANN) model to be trained through the use of decentralized edge devices i.e. workers that maintain data locally, without sharing the data with the server i.e. the service provider. This method requires a central server to broadcast the ANN model to multiple workers, coordinating transmission and responses. The workers will locally fit the ANN and send the updated weights back to the server. FL principles [10] require that the server responsible to orchestrate the learning process does not receive workers' data under any circumstances, allowing a neural model to be trained without compromising data privacy. Thus, it is possible to overcome problems related to the processing and storing of personal data, and to obtain trained predictive models. However, in a FL environment data can be unevenly distributed within the workers, leading to under-representation of one or more specific population subgroups. This can result in unfair prediction, statistical disparity, and inequity [11].

Considering a classical FL approach [10], the service provider has no means to ensure that data is evenly proportioned, or to estimate the impact of the data distribution across the whole set of workers on ML predictions. Zero knowledge proof (ZKP) [2] can be used to prove a statement having no knowledge of the statement itself. The proposed approach consist of implementing the Schnorr's ZKP authentication protocol [3, 8] that can be used to infer the data distribution of the remote workers without data exchange. Motivated by experimental results that show unfair treatment for imbalanced data (Sects. 2, 5), the following research question is investigated: to what extent can ZKP inferred data about the proportions of population groups in a federated learning environment mitigate federated learning bias while in compliance with GDPR and EU guidelines for data ethics and trustworthy AI?

By performing differentiated evaluations on an ANN model trained on imbalanced data, it is possible to observe an average increment of disparity that leads to prediction bias as described in Sect. 5. This outcome drove the design of a self-balancing ZKP FL environment called Z-Fed to support a fair, privacy-preserving learning process for datasets with multiple sensitive population categories. The technical process followed to achieve Z-Fed is as follows: 1) The federated server generates tokens to authenticate all the possible workers with the ZKP Schnorr's protocol [3]; 2) Workers encrypt their feature labels i.e. categorical labels, fit the learning model, and send the update to the server; 3) The server can zero-knowledge prove that workers belong to group identified by a certain feature label by retaining the encrypted version of the workers' labels and count individuals; 4) The server uses a self-balancing queue system to accept updates in a manner that ensures the clients will not compromise the balance.

In this paper, we use an ANN based on the statistical gradient descent (SGD) algorithm for weights update. This model is used for supervised training tasks on the UTK dataset [12] and is trained using images of faces to predict their age. We implemented an FL framework and trained the ANN with balanced

and imbalanced samples of the dataset in order to select appropriate metrics of comparison.

The main contributions of the present paper are the following: 1) Identification of a set of metrics i.e. EOD, EPD, and SPD to measure equality degradation in imbalanced FL training; 2) Design of a self-balancing ZKP FL framework, *Z-Fed*, implementing zero knowledge authentication to avoid malicious workers updating the model; 3) Implementation of ZKP inference of worker data distribution to allow data augmentation and rejection of imbalanced updates and counter effect bias; 4) Evaluation of the self-balancing framework based on an stochastic gradient descent (SGD) ANN. The experimental results can be summarized as follows: with respect to an imbalanced FL framework, the measured scores relative to absolute multi-class (Sect. 5) statistical parity difference (SPD), equal opportunity difference (EPD), and equal odds difference (EOD) are considerably improved in the experiments conducted using self-balancing *Z-Fed*. Detailed results are available in the evaluation Sect. 5.

2 Background and Motivations

Fairness and equity are general ideas not restricted to AI. An application that implies decision-making processes can show discriminatory bias towards some specific groups and thus must be evaluated in terms of fairness. The EU guidelines for trustworthy AI [6] define disparate treatment as a major concern in AI. In the fair credit reporting act (FCRA), fairness regards individual attributes such as gender, race, religion, age, sexual orientation and more. An unfair or disparate treatment occurs when the outcome of a decision is biased by such factors. While for explainable algorithms it can be easier to identify possible discrimination, this represents a major challenge in FL [11].

This section reviews the main research findings in the area of machine learning in FL environments involving the use of unbalanced data. It will be discussed how the presented approach contributes in relation to the existing effort. The three main areas of research were 1) fairness and advances of FL, 2) machine learning with unbalanced data, and 3) ZKP methods.

Federated Learning. A central server can collect fitted ANN model weights using synchronous or asynchronous protocols [10]. While recent advancements in FL led to the design of solutions to deal with the accuracy reduction due to uneven distribution of data using mediators [4] or probabilistic approaches [9], the proposed method focus on improving the fairness of the predictive model by performing ZKP self-balancing.

Imbalanced Data Machine Learning. Ensemble methods are proposed to reduce bias in imbalanced data learning [7], but they can suffer the presence of outliers typical of FL. The main proposed solutions for dealing with imbalance data are sampling and augmenting data [5]. Over-sampling, often implemented by artificially creating minority classes to counter the effect of disproportions [13], shows promising results, but requires access to data, and hence is not suitable for FL environments.

Zero Knowledge Proof. ZKP can be used to enhance data privacy in online communication [2] and can be implemented using iterative [1] or non-iterative methods [3]. Iterative ZKP is inconvenient in FL, since it considerably increases the communication overhead. Non-iterative implementations of ZKP are often used for authentication [8] without involving exchange of privacy protected data, which make this method suitable for FL. ZKP authentication allows a server to prove that a client knows certain information without revealing it. This is possible through the use of encrypted tokens, i.e. proofs and signatures, that ensure that only authorized clients can be authenticated.

3 Requirements

While performing FL, the federated server requires the users to ZKP authenticate to prove that 1) they are authorized to contribute training and 2) they are not holding data which does not belong to any subgroup. While distributing the model for ML, the server is able to count the number of samples of each category used for training.

ZKP Framework Initializer. ZKP authentication is enabled by public data structures, i.e. tokens that use elliptic curves and generator points to prove that a specific authentication proof come from the same token that was generated from the server during the registration process. Since there is no exchange of private data in the registration phase, a malicious remote client could force the server to register its features even if they do not belong to the *features* public dictionary. This is possible because the server would receive only an encrypted version of the label based on the private number n of the ZKP client. In this case, this malicious behavior leads to distortion in the count of groups and results in ineffective control of data balance. Moreover, the framework would suffer higher computational and storage payload by generating multiple authentication tokens for each worker.

In order to counter the effects of unreliable clients, keep the learning environment trustworthy, and to reduce the computational resource needed, it is possible to create a very limited number of authentication tokens during an early server setup phase.

Learning Model. Any machine learning model can be used as long as it presents the following APIs: 1) initialize the learning model, setting e.g. the learning rate η , 2) read the values of the trainable parameters e.g. weights together with the configuration settings, 3) fit an input list X given a label list y e.g. propagate the inputs through the neural layers, measure the loss and update the training parameters, 4) load external training parameters received e.g. from remote clients, and 5) produce a list of predictions y_{pred} given a list of inputs X .

4 Design of Z-Fed

A ZKP protocol is designed to enable the server to register every possible subgroup within the dataset. To do so, the server must be aware in advance of the

possible categorical features that data can possibly have, e.g. values of ethnic group and gender. For this purpose, it is possible to create a number of *client prototypes* equal to the number of subgroups present in the data. Client prototypes are not used for ANN training, but only for creating registration tokens. The server can use them to set up encrypted dictionaries that are used to count the number of samples belonging to specific subgroups. A service called *framework initializer* is required to generate a private number n that can be transmitted to the workers and produces the client prototypes. In the proposed architecture, the federated framework is initialized using the aforementioned service.

ZKP Server. The ZKP server must be initialized with a private *password* to prevent the tokens from being vulnerable by using encryption. The server can use the password to generate ZKP signatures. Having the server countersigning a client signature allows the client to prove that the server is legit. The server must store a copy of the *features* data structure. From now on, we will refer to the possible features in the dictionary of the UTK dataset as feature name, e.g. *Ethnicity* and *Gender*, and will refer to the possible values as feature labels, e.g. *Female*, *Male*, etc.

The server, during the registration phase, is able to create *tokens* for authenticating authorized workers. An authorized ZKP client can send its signature to the server, and later the server can use the client signature to create a *token*. The registration phase ends when all the possible subgroups combinations, (e.g., *Gender: Female, Ethnicity: Asian, . . . , Gender: Male, Ethnicity: White*) have a server-side token representation. The server can authenticate clients by checking if they have a proof that is compatible with any of the tokens, meaning that the client belongs to a specific subgroup.

The server retains an encrypted representation of the client subgroup categories in its encrypted dictionary. At any given moment, the server can assess whether the distribution of data is even or not. Before a worker is requested to contribute to the ML process, the server can check if the worker would result in an uneven data distribution, and in this case it will reject any update from it. When a worker is not able to train the distributed model because of potential imbalance, the server is able to register the workers' identifier to possibly connect to it later in case its update would not result in imbalance. To optimize the process, the server retains a priority queue data structure. Moreover, if the dataset is highly imbalanced, the server can augment the training mechanism by requesting multiple epochs of training for under-represented groups.

The ZKP Server requires a data structure to store the ZKP parameters needed for registration and authentication [2,3], such as the elliptic curve of choice *curve*, the public *Salt* value, the private number n , the hash function of choice *hash*, and the curve generator point g .

The ZKP Server needs a dictionary structure named *groups* to count encrypted versions of subgroups. Since every feature label has a token representation on the server, for each of the k feature names on the shared *features* dictionary, a client must show to have k feature labels compatible with the feature structure in order to authenticate. Once authenticated, the server will

receive k count updates indexed with k hexadecimal hash number. The k hashes will be summed and used as a dictionary key to manage the FL server queuing protocol.

ZKP Client. A ZKP client is responsible for representing a specific individual tuple *feature name*, *feature label* in the distributed dataset. Given the number k of feature names in the features dictionary, every worker will instantiate k ZKP clients. Every ZKP client store the *feature name*, the *feature label*, the ZK data structure analogue to the one of the ZKP server. ZKP clients can generate a signature encrypted using a password. In this case, the ZKP client password, i.e. *secret*, is the hashed value of the *feature label* joint with the private number n : $secret = hash(feature\ label|n)$, where $|$ is a string operator, e.g. concatenation. Using the private number n the ZKP server is not able to decode the client token to read the label. Moreover, the ZKP client can create an *encrypted label* using a different method to joint the *feature label* with the number n , e.g. $encrypted\ label = hash(n|feature\ label)$. The client is safe to publicly send the value of *encrypted label* without revealing the *secret* or the *feature label*. The *encrypted label* value is used to server side count the subgroups.

Worker Registration and Authentication. A server S , i.e. verifier, and a client, c i.e. prover, are such that c can prove to S that a given condition results true, avoiding sharing any information but the fact that the condition is true. The server chooses a password $S_{password}$ and the client chooses a secret c_{secret} e.g. the value of the ethnic group of belonging that does not want to share with S . Based on [3], S and c choose the following public parameters respectively: an elliptic curve S_{curve}, c_{curve} with elliptic curve generator points S_g, c_g , a hash function S_{hash}, c_{hash} , and a relatively big random number S_{salt}, c_{salt} . In addition, ZKP server and client produce random private variables, S_n, c_n respectively, used to compute a specific point on the elliptic curve. Using these settings it is possible to create a signature i.e. token of the form of $token = g \times hash(secret|salt) \bmod n$ which can be shared publicly revealing no information about the secrets. After c sends its signature to S , the latter can subsequently sign the received token, publish the newly signed token, and retain public client parameters, i.e. *registration*, this way the server can prove if a further token comes from the same client that used the same server signature in the past, i.e. *authentication*.

ZKP Framework Initializer. We propose a trusted external service in charge of generating one client prototype for each of the possible q combination of feature names and values. The proposed framework initializer has the duty of coordinating with the ZKP server to register the client prototypes and generate the q authentication tokens. This can be achieved only if all the clients share the same private number n , that allows them to sign the server tokens to generate proofs. For this reason, all the workers must connect to this service and get the value of n prior to authenticate to the ZKP server. This results in an additional step in the FL process, but the framework allows doing this asynchronously. The initialization process is described in Fig. 1a.

Federated Server. A self-balancing federated server must be able to discern updates based on the subgroup of belonging of the client. After ZKP authentication, the server estimates whether the count of subgroups would result in imbalance, and, under this circumstance, rejects the update. The server may identify workers with an identification number w_{ID} . This allows the server to organize rejected workers into queues and efficiently select workers for further balanced updates. To simplify, the presented model executes synchronous FL, meaning that the server elaborates updates one at a time. The server retains a dictionary of queues, used to store examples belonging to different subgroups. Since the workers have one or more hexadecimal hashed labels representing the subgroups, it is possible to sum the values to create the index of a hash-table used to access the specific subgroup queue. Having the count of subgroups, it is possible to check if an update will keep the model balanced.

Federated Worker. The federated worker is responsible for training the distributed model and provide weight updates to the server. Workers retain a number k of ZKP clients equals to the number of feature names present in *features*. Workers present a data structure to store the parameters required for model training, and a local copy of the learning model. A worker can retain a list of pairs of training features and ground truth, X and y respectively. Additionally, workers retain a dictionary of the secret feature names and labels for subgroup count. A generic Z-Fed worker must load the model received from the server, propagate the model using the X, y pairs, calculate loss and update the weights, send the updates of weights and subgroups count to the server. The components of Z-Fed are shown in Fig. 1b.

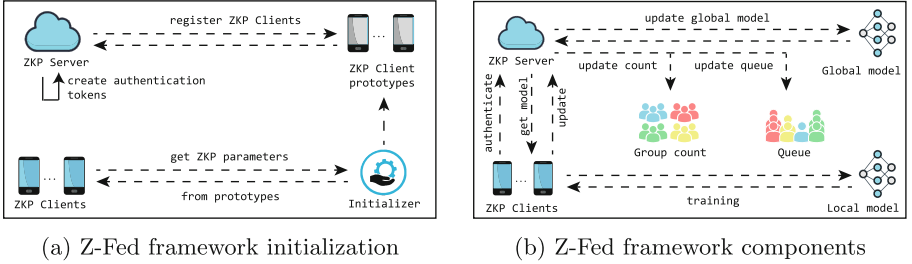


Fig. 1. Design of the Z-Fed framework

The workflow for the initialization, groups count, queue management, data augmentation, and model training of Z-Fed is described as follows: 1) The framework initializer generates a random private number n and uses *features* to create as many client prototypes as population subgroups; 2) Asynchronously, the server can instantiate the ML model and prepare weights; 3) Once the client prototypes are ready, the framework initializer can request the server to produce the required authentication tokens; 4) Workers are initialized and updated using

the client prototypes, from this moment they can retrieve authentication tokens from the server, authenticate and receive the ML model for FL; 5) Server authenticates workers and uses the updates to train the global ML model; 6) Rejected workers are organized into a structure of queues to reschedule the training efficiently. A diagram of the Z-Fed workflow is shown in Fig. 2.

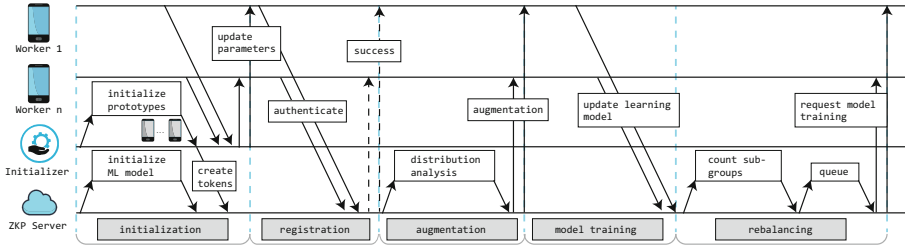


Fig. 2. Z-Fed workflow diagram: initialization, registration, data augmentation with population proportion analysis, federated model training, and re-balancing of workers by ZKP count of subgroups.

5 Evaluation

In this paper, we focus mainly on: 1) the difference in rate of favorable outcomes for unprivileged groups with respect to privileged groups i.e. statistical parity difference (SPD) across subgroups, 2) the difference in rate of true positive prediction outcomes between privileged and unprivileged groups i.e. the equal odd difference (EOD) across subgroups, and 3) the difference of probability to get true positive and false positives between privileged and unprivileged groups, i.e. the equal opportunity index (EPD).

Benchmark Measurements. To the best of our ability, scientific research presenting fairness measurements against imbalanced UTKFace (UTK) datasets [12] could not be found for benchmark. The UTK dataset presents the records of 23706 persons, providing their age, ethnicity, gender, and a black and white picture. An exploratory search was conducted based on the following hypothesis: in a federated environment, it is possible to measure bias using EPD, EOD, and SPD if the training dataset is class imbalanced.

To assess the influence of imbalanced training data in FL, we trained ANNs with an up-sampled UTK dataset and measured fairness metrics afterwards. We used face images to predict four age ranges i.e. *0 to 9*, *10 to 19*, *20 to 29*, and *30 to 39* considering four ethnic groups i.e. *Asian*, *Black*, *Indian*, *White* and two gender groups i.e. *Female*, *Male*. Ethnic and gender groups can be considered in all the 8 possible combinations to form subgroups.

Imbalanced Datasets. A simple way to create class inequity is to define a privileged (PR) class that is over-represented with respect to the other unprivileged classes. Four different dataset are built, choosing one ethnic group as privileged, with a class proportion distributed as follows: 85% of the samples belong to the privileged group and the remaining 15% of the sample are equally split among the rest of the unprivileged ethnic groups. All the datasets have the gender and the age range features balanced. The datasets described previously will be further identified as ASIAN-PR, BLACK-PR, INDIAN-PR, WHITE-PR. In addition, an ethnic-gender class balanced dataset (BAL) is set up for training, evaluation and comparison.

ML Model Architecture. The ANN has the following fully-connected layer (FCL) structure: $2304 \times 96 \times 4$ neurons plus one bias neuron per FCL, and uses a sigmoid activation function and a mean square error (MSE) loss function. The ANN model has a total of 884,736 training parameters i.e. weights \mathbf{w} and achieve an average accuracy of 50.8%, variance 1.07% after fitting 16,000 samples in one epoch with $\eta = 0.025$ on UTK.

FL Settings. Every worker holds a sample of size one, and the FL framework is set up to compute one epoch per training cycle. In these settings, the model performed an average SPD of 1.86%, EPD of 4.6%, and EPD of 1.84% on BAL, and we measure an SPD of 15.02%, EPD of 15.01%, and EPD of 5.17% on ASIAN-PR. In addition, the model shows a negligible difference in average absolute EOD on both BAL and WHITE-PR, while showing a flat slope on BAL and significant growth of inequity in ASIAN-PR during the model update rounds as shown in Fig. 3. Moreover, we tested the accuracy of the ANN against a specific ethnic group, measuring the variance among subgroups. Considering an accuracy variance of 0.09% on BAL, the ANN shows a subgroups accuracy variance of 3.22% on ASIAN-PR, meaning that it is more likely to have different treatment in case of imbalanced data. Figure 3 shows equality scores of ANN while training.

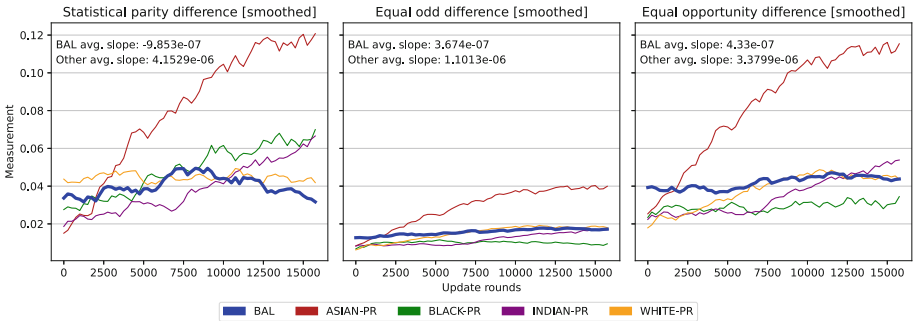


Fig. 3. Measure of equality in terms of SPD, EPD, and EOD on different balanced and imbalanced datasets.

The settings of the experiments performed involve having multiple unprivileged classes and one privileged class. This requires to calculate the SPD, EPD, and EOD metrics one time for each unprivileged class, with respect to the privileged class. Since the purpose of Z-Fed is to mitigate the effect of imbalanced data in a FL environment, we decided to treat both kind of discriminatory behaviors, i.e. favoring privileged groups and favoring unprivileged groups, with the same importance. For a privileged class PR , and l unprivileged classes UPR_i , with $i = 1, \dots, l$, we calculate the SPD, EPD, and EOD values l times with respect to PR to have a fine-grained measurement of equity. These evaluations can be expensive and difficult to interpret in presence of a high number of different subgroups, considering e.g. the possible combinations of ethnicity, gender, age, etc. For this reason, we consider a more convenient absolute value of equity $|m|$ such that $0 \leq |m| \leq 1$ and present the average of all the values obtained from the l unprivileged subgroups. To summarize, for each measurement m , across l unprivileged groups, the absolute multi-class equity score is: $\sum_{i=1}^l |m_i|/l$. In this paper, we refer to the absolute multi-class measurements of statistical parity difference, equal opportunity difference, and equal odds difference, as SPD, EPD, and EOD, respectively.

We used the datasets and the results of the experiment described in Sect. 2 as a baseline. In the Z-Fed framework, the support for self-balancing learning can be arbitrarily disabled for testing purposes. We use the imbalanced datasets described and multiple instances of the same learning model for each experiment, and run training sessions on Z-Fed in order to obtain: 1) the Z-Fed model trained with highly imbalanced classes and self-balancing mode disabled, denoted as *imbalanced*, and 2) the Z-Fed model trained with the same highly class imbalanced dataset and self-balancing mode enabled, denoted as *rebalanced*. The self-balancing Z-Fed is set up to perform multiple training epochs on under-represented groups to counter-effect the fact of having a relatively small number of examples in the dataset. Identically as it was done for the FL experiment in Sect. 2, we tested the multiple instances of the same learning model on Z-Fed, using the face images as training features and predicting the age ranges. It is important to point out that the features used for creating imbalanced data, i.e. ethnicity, are not training features, means that the influence that can have on predictions is indirect. The age range chosen as feature to predict is, in each dataset, balanced, meaning that for the four age ranges 0–9, 10–19, 20–29, 30–39 have a proportion of $25\% \pm 1\%$ each in every experiment. The test sets, used to measure the equity scores in the *imbalanced* Z-Fed experiments, were sampled maintaining the original class proportion of the privileged and unprivileged subgroups. To test the performance of the *rebalanced* experiments we used a class balanced test, this decision is taken to respect the proportions of the balanced training set. We measure the SPD, EOD, and EPD for each of the four experiments, the results are presented in Table 1. By analyzing the proportion of the population groups, Z-Fed is able to request more training epochs to worker belonging to under-represented classes. This results in a bigger number of training updates for the *rebalanced* experiments. In terms of SPD, Z-Fed successfully

Table 1. Z-Fed measurements of SPD, EPD, and EOD

Notation	Updates	SPD		EPD		EOD	
ASIAN-PR [imbalanced]	16,008	0.1046	(reference)	0.1127	(reference)	0.038	(reference)
ASIAN-PR [rebalanced]	22,737	0.0216	−79.30%	0.0216	−80.80%	0.007	−81.46%
BLACK-PR [imbalanced]	16,008	0.0461	(reference)	0.0587	(reference)	0.0211	(reference)
BLACK-PR [rebalanced]	22,737	0.0488	+5.63%	0.0488	−16.89%	0.0162	−23.02%
INDIAN-PR [imbalanced]	16,008	0.087	(reference)	0.0977	(reference)	0.0344	(reference)
INDIAN-PR [rebalanced]	22,737	0.0194	−77.79%	0.0194	−80.14%	0.0064	−81.20%
WHITE-PR [imbalanced]	16,008	0.0381	(reference)	0.040	(reference)	0.0141	(reference)
WHITE-PR [rebalanced]	22,737	0.0255	−32.95%	0.0255	−36.34%	0.008	−39.97%

reduce the class bias in ASIAN-PR, INDIAN-PR, and WHITE-PR by 79.3%, 77.79%, and −32.95% respectively. Z-Fed produces a small SPD increment of 5.63% in BLACK-PR, meaning that the overall accuracy of the rebalanced learning model has the tendency of favoring either privileged or unprivileged groups in this particular experiment. The measures of EPD show a considerable improvement in fairness in all the experiments ASIAN-PR, BLACK-PR, INDIAN-PR, and WHITE-PR, with a decrement of opportunity disparity of the 80.8%, 16.89%, 80.14%, and 36.34% respectively. The proportions about true positive results and false positive results in predictions improve considerably with the use of Z-Fed. The EOD measurements also show a notable improvement in fairness across all the experiments. In ASIAN-PR, BLACK-PR, INDIAN-PR, and WHITE-PR, the odd disparity was reduced by 81.46%, 23.02%, 81.2%, and 39.97% respectively. The true positive rate measurement within privileged and unprivileged groups is considerably improved by the use of Z-Fed.

6 Conclusions

FL is a promising ML method that assists data privacy. However, we show how imbalanced data leads to disparity (unfairness) in the UTK dataset. The Z-Fed framework proposed is able to mitigate FL bias by reducing disparities without compromising privacy. We show that ZKP enables to count the number of population samples keeping track of the proportion of subgroups, e.g. ethnicity, gender. Subgroups proportion can be used to rebalance the FL samples and augment ML data, achieving an increment of fairness in terms of three measures: statistical parity difference, equal odd difference, and equal opportunity

difference. On average, Z-Fed improves the EPD of 53.54%, the EOD of 56.41%, and the SPD of 46.1% on imbalanced UTK datasets.

References

1. Bazrafshan, M., Gatsis, N.: Convergence of the Z-bus method for three-phase distribution load-flow with zip loads. *IEEE Trans. Power Syst.* **33**(1), 153–165 (2018)
2. Buchanan, W.J.: 11 Zero-knowledge proof (ZKP) and privacy preserving, pp. 337–368 (2017)
3. Chatzigiannakis, I., Pyrgelis, A., Spirakis, P.G., Stamatiou, Y.C.: Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices. In: 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, pp. 715–720 (2011)
4. Duan, M., et al.: Astraea: self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In: 2019 IEEE 37th International Conference on Computer Design (ICCD), pp. 246–254. IEEE Computer Society, Los Alamitos (2019)
5. He, H., Garcia, E.A.: Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **21**(9), 1263–1284 (2009)
6. High-Level Expert Group on AI: Ethics guidelines for trustworthy AI. Report, European Commission, Brussels (2019). <https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai>
7. Kenfack, P.J., Khan, A.M., Kazmi, S.A., Hussain, R., Oracevic, A., Khattak, A.M.: Impact of model ensemble on the fairness of classifiers in machine learning. In: 2021 International Conference on Applied Artificial Intelligence (ICAPAI), pp. 1–6 (2021)
8. Pathak, A., Patil, T., Pawar, S., Raut, P., Khairnar, S.: Secure authentication using zero knowledge proof. In: 2021 Asian Conference on Innovation in Technology (ASIANCON), pp. 1–8 (2021)
9. Wang, H., Kaplan, Z., Niu, D., Li, B.: Optimizing federated learning on non-IID data with reinforcement learning. In: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, pp. 1698–1707 (2020)
10. Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., Yu, H.: Federated learning. *Synthesis Lect. Artif. Intell. Mach. Learn.* **13**(3), 1–207 (2019)
11. Zhang, D.Y., Kou, Z., Wang, D.: FairFL: a fair federated learning approach to reducing demographic bias in privacy-sensitive classification models. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 1051–1060 (2020)
12. Zhang, Z., Song, Y., Qi, H.: Age progression/regression by conditional adversarial autoencoder. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4352–4360 (2017)
13. Zhou, Y., Shu, J., Zhong, X., Huang, X., Luo, C., Ai, J.: Oversampling algorithm based on reinforcement learning in imbalanced problems. In: GLOBECOM 2020–2020 IEEE Global Communications Conference, pp. 01–06 (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Automatic Vehicle Ego Body Extraction for Reducing False Detections in Automated Driving Applications

Ciarán Hogan¹(✉) and Ganesh Sistu²

¹ Department of Electronic and Computer Engineering, University of Limerick,
Limerick, Ireland

ciararan.hogan9@gmail.com

² Valeo Vision Systems, Tuam, Galway, Ireland

ganesh.sistu@valeo.com

Abstract. Fisheye cameras are extensively employed in autonomous vehicles due to their wider field of view, which produces a complete 360-degree image of the vehicle with a minimum number of sensors. The drawback of having a broader field of view is that it may include undesirable portions of the vehicle's ego body in its perspective. Due to objects' reflections on the car body, this may produce false positives in perception systems. Processing ego vehicle pixels also uses up unnecessary computing power. Unexpectedly, there is no literature on this relevant practical problem. To our knowledge, this is the first attempt to discuss the significance of autonomous ego body extraction for automobile applications that are crucial for safety. We also proposed a simple deep learning model for identifying the vehicle's ego-body. This model would enable us to eliminate any pointless processing of the car's bodywork, eliminate the potential for pedestrians or other objects to be mistakenly detected in the car's ego-body reflection, and finally, check to see if the camera is mounted incorrectly. The proposed network is a U-Net model with a ResNet50 encoder pre-trained on ImageNet and trained for binary semantic segmentation on vehicle ego-body data. Our training data is an internal Valeo dataset with 10K samples collected by three separate car lines across Europe. This proposed network could then be integrated into the vehicles existing perception system by extracting the ego-body contour data and supplying this to the other algorithms which then ignore the area outside the contour coordinates. The proposed network can run at set intervals to save computing power and to check if the camera is misaligned by comparing the new contour data to the previous data.

Keywords: Autonomous vehicles · Computer vision · U-Net · ResNet · Semantic segmentation · Vehicle ego-body · Region of Interest (ROI) · Fisheye

1 Terminology

The following is a short explanation of technical terms that will be used throughout this paper:

Region of Interest (ROI): For the purpose of this paper ROI relates to the active scene of the cameras view, where the vehicles own body is not visible (i.e road, pedestrians, other vehicles etc). This is a common computer vision term used to describe the useful areas of an image to be fed to algorithms.

Semantic Segmentation: Semantic segmentation is the labelling or classifying of every pixel in a image.

Vehicle Ego-Body: This refers to a vehicle's own body which is visible in wide field of view cameras like fisheye. (Derived from the Latin meaning of ego which is "I").

Mask: Mask is a term used to describe a binary image that defines where a particular object or region is in the image. Each mask represents a class.

Contour Values: Contour values refer to the coordinates of the boundary line between the Region of interest and the vehicle ego-body.

Convolutional Neural Network (CNN): Convolution neural network is a type of feed-forward neural network used in tasks like image analysis, natural language processing, and other complex image classification problems

Fisheye: Fisheye refers to wide field of view cameras that usually covers angles of about 180° .

Intersection over Union (IoU): IoU is an evaluation metric typically used for segmentation tasks. IoU is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth.

2 Introduction

In recent years the improvement of deep learning techniques like convolutional neural networks and recurrent neural networks have resulted in rapid growth in the area of autonomous driving. Deep learning models play a vital role in the operation of autonomous vehicles but they are not without their faults and limitations [1].

First of all, false detection of objects, road markings, curbs, and pedestrians in the reflection of the bodywork of the vehicle can cause serious problems in autonomous vehicles. This occurs when the vehicle ego-body acts as mirror like surface and the network then falsely detects the reflection of an object or a pedestrian on this surface. Examples of these false detections can be seen in Fig. 1.

This could lead to emergency braking and could result in the car being rear ended and the passengers seriously injured or, could potentially be fatal if this

occurred at motorway speeds. We have already seen that too many reputable car manufacturers have had issues exactly like this recently [14,15]. In 2019 one large reputable car manufacturer announced a National Highway Traffic Safety Administration or ‘NHTSA’ investigation and recall of one of its vehicles due to the automatic emergency braking engaging when there was no obstruction in the path of the vehicle [15]. Accidents and injuries have been reported by customers related to this issue which is definitely not desirable and could result in the loss of lives and cost a company billions.



(a) False curb detection (Blue)
& Soiling (green)



(b) False pedestrian detection
(Red bounding box)

Fig. 1. False detections on vehicle ego-body (Color figure online)

Secondly, each vehicle has different camera positions and configurations. Each SVS (Surround View System) in vehicles have four different camera views as seen in Fig. 2 and in each camera view the vehicle ego body is in different locations. This means if finding the ROI (Region of Interest) by finding the positions of the ego-body manually it would have to be done four times and then would have to be done for every vehicle model in which the cameras are installed which would be tedious and leave room for human error. This would then have to be repeated for every vehicle manufacturer that utilises the cameras. It is also hard to pinpoint where exactly the camera is going to be positioned in its housing by the manufacturer/assembler and there could be some variability from vehicle to vehicle of the same model.

Over the lifetime of the camera different issues can arise. First, the camera may move in its housing which changes the ego-body position within the camera view. Secondly, cameras could even become fully misaligned meaning that the camera may have to be re-positioned and re-calibrated.

The main objective of this paper is to tackle the issues mentioned above and mainly the problem of false detections on the vehicle's ego-body. In this paper we propose to solve these issues by detecting where the vehicle ego-body is in each image using semantic segmentation and using post processing we can extract the coordinates of the boundary between the ROI and the ego-body. With these known coordinates we can supply them to the main perception algorithms so they can focus on just the ROI.

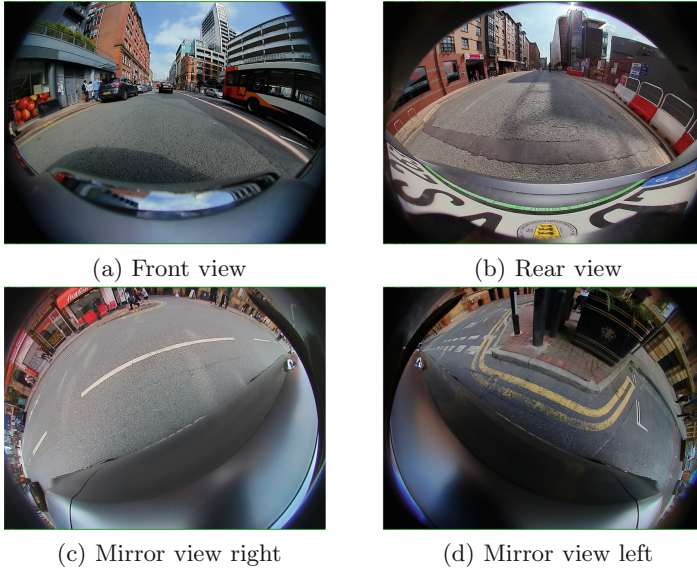


Fig. 2. Camera views on vehicle

3 Literature Review

Semantic Segmentation: Semantic segmentation plays a very important role for scene understanding in autonomous driving. Semantic segmentation involves classification of every pixel of an image into their relevant classes.

Yogamani et al. (2018) paper [2] carried out a comparative study of Real time semantic segmentation algorithms for autonomous driving. The study compared the performance of the combinations of different encoder and decoders. The encoders and decoders they trialled were SkipNet, MobileNet, ShuffleNet, UNet, ResNet18 and Dilation Frontend. The experiments were carried out using the Cityscapes dataset [12] and the mIoU scores for each of the relevant classes were recorded. One of the main takeaways from the experiment carried out in the paper was that the “UNet decoding method provides more accurate segmentation results” [2].

Fisheye: Currently there are very few studies which attempt to perform semantic segmentation directly on fisheye images using Deep Learning techniques and virtually no studies that could be found that use semantic segmentation for vehicle ego body detection/ROI extraction using raw fisheye images. This is mainly due to two reasons, firstly managing strong distortion in fisheye images and, secondly the lack of a large scale fisheye native dataset available [3]. In the past most studies based on fisheye datasets had to manually construct their own datasets by taking existing datasets and projecting the images and labels to fisheye format [3].

In 2019, Valeo released the Woodscape dataset which is the first extensive public automotive fisheye dataset including over 10,000 semantic segmented and annotated images for public usage [11], along with a paper [4]. In the paper the authors detail the distinct advantage of using fisheye cameras in automotive applications, because of their wide field of view they can get a full 360° surround view of the vehicle with a minimal number of sensors.

In a paper by Deng et al. [5] they propose CNN based semantic segmentation for urban traffic scenes using fisheye images. They first constructed a fisheye dataset constructed from the well known Cityscapes dataset [12]. To handle the complex scene in the fisheye image, local, global and pyramid local region features are integrated by an overlapping pyramid pooling (OPP) module. They found that as the OPP module allows arbitrary-sized input it keeps good translation invariant property and shows better performance than sub-region pyramid pooling module. In this study they also implemented zoom augmentation in which they change the focus length of the fisheye generated image and this showed improvement for generalization of the system.

Mariotti and Eising (2021) in their paper [13] “Spherical formulation of geometric motion segmentation constraints in fisheye cameras” attempt to solve the problem of motion detection for fisheye cameras by reformulating the problem in spherical coordinates which can address both the non-linearity and the large field of view. To solve the problem of motion segmentation using fisheye cameras, four geometric constraints were unified, namely, epipolar, positive depth, positive height and anti-parallel for the detection of moving obstacles in the scene. The results presented, based on dense optical flow, show that the geometric approaches described are effective at detecting arbitrary moving objects. They concluded that the integration of the geometric constraints as described in this paper into a neural framework would yield optimal results.

Detecting Reflections: There are not many solutions available right now to automatically find reflections in an image. Problematic mirrors may typically be disregarded in an applied computer vision by manually drawing the ROI. Labelling the ROI requires human effort and relies on the camera and mirror maintaining a fixed perspective. There has been some work on the automatic detection of reflecting planes using geometric models of the image and its reflection [6, 7], but this was not explored in the context of segmentation and had a number of barriers to practical use.

In 2019 a paper [8] was published where the authors attempt to solve the problem of false positive detections due to reflections using segmentation. In the paper they propose the use of semantic segmentation for better scene understanding and in order to reduce instance segmentation false positives. They found that in their Mask R-CNN model the fusion of both segmentation types decreased false positives in images by over half and that this method was not just limited to actual mirrors but can be applied to other glossy surfaces also. They also found that using this method the precision increased from 71% to 83% and meanwhile the increase in false negatives was very small.

Literature Review Conclusion. The problem at hand seems to be novel and has not been discussed in any literature or articles. The false detection of pedestrians and objects in the ego-body reflection could be a serious problem which needs addressing. From examining the papers mentioned about segmentation of fisheye images we have the advantage of having a native fisheye dataset rather than having to generate fisheye images from normal images. Also research shows that semantic segmentation should be relatively easy using a robust architecture like UNET.

4 Implementation

Data Processing: The dataset consists of 13,184 images and masks in total, a 54:46 train/validation split was implemented. The data used to train the model consists of 7134 native fisheye internal Valeo images and ego-body masks and the data for testing consists of 6050 native fisheye internal Valeo images and masks. The data contains images and ego-body masks from all vehicle surround view cameras: front view, rear view, mirror view left and mirror view right. The segmentation masks in the dataset are in RGB format. They were converted to ‘One hot’ encoding where the ego-body mask was encoded 1 and the ROI was encoded 0 as we are performing binary semantic segmentation. The images and masks in dataset are a mix of three resolutions: 1280×966 px, 1280×1536 px and 1280×1632 px. Transforms were then applied to each image and mask where they were resized to a resolution of 640×480 px and also normalised. Data augmentation was also implemented on the training data: rotation, horizontal flip, vertical flip and blur were all employed in the implementation to help improve model performance.

Architecture: The proposed architecture is based on a UNET model with a ResNET50 encoder pre-weighted on ImageNet. UNET is a semantic segmentation architecture that was developed originally for biomedical image segmentation. UNET consists of two paths, contracting and expanding. The contracting path (encoder) is made up of convolutional and max pooling layers for down-sampling while the expanding path (decoder) is for precise localisation using transposed convolutions for up-sampling. Finally, the output of the network produces a binary encoded semantic segmentation map [9].

In the implementation we propose some slight changes to the architecture. The original UNET encoder was replaced by a ResNet50 encoder pre-weighted on ImageNet to improve model accuracy. Residual networks or ResNet is a Convolutional Neural Network (CNN) architecture, made up of a series of residual blocks (ResBlocks) with skip connections differentiating ResNets from other CNNs [10].

The overall purpose of the proposed network is to use semantic segmentation to extract the location of the vehicle ego body, the generated mask is then post processed in order to extract the contour values. This information will then be provided to the other perception algorithms.

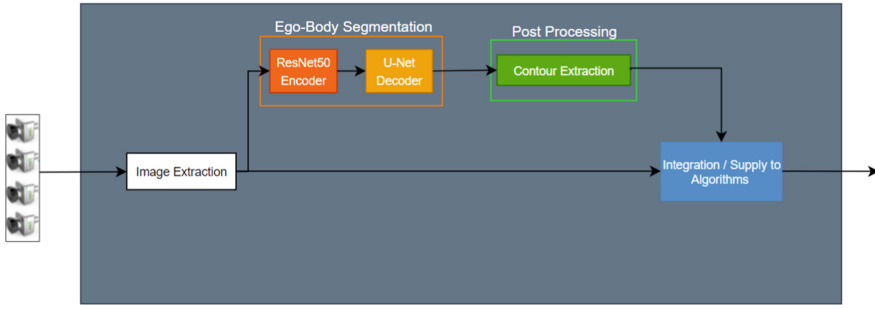


Fig. 3. Proposed architecture integration

Evaluation Metrics: Segmentation tasks require their own set of specific evaluation metrics as other metrics like pixel accuracy can give misleading information for segmentation tasks due to class imbalance.

Dice loss was chosen to measure the models loss. Dice loss is a loss function adapted from Dice Coefficient. Dice coefficient or F1 score, in simple terms is used to calculate the similarity between two images. The equation for the Dice coefficient D is shown in Fig. 4, where p_i and g_i stand for pairs of corresponding pixel values for the prediction and ground truth, respectively. In a boundary detection scenario, p_i and g_i values are either 0 or 1, indicating whether or not the pixel is a boundary. The Dice loss is then calculated by 1-(Dice coefficient).

$$D = \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}$$

Fig. 4. Dice coefficient formula

IoU, as seen in Fig. 5, is the area of union between the predicted segmentation and the ground truth divided by the area of overlap between the predicted segmentation and the ground truth. This metric ranges from 0–1 (0–100%) with 0 representing no overlap and 1 representing perfectly overlapping segmentation.

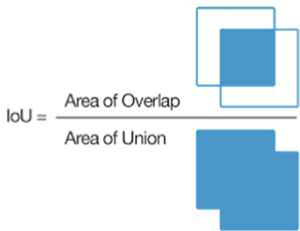


Fig. 5. Intersection over union formula

5 Evaluation and Results

The pre-trained ResNet50 encoder and UNet decoder was run for 20 epochs on the pre-processed data with a batch size of 4. Model parameters were optimised using the Adam optimizer with a learning rate of 0.0001. As mentioned previously the dataset contains 13,184 images and masks in total, a 54:46 train/validation split was implemented. Figure 6 below shows the IoU score plot over the 20 epochs, showing that the model performs well in both the training and validation sets with high IoU scores between 0.9750 and 0.981. The model was run for a greater number of epochs but there was minimal increase in IoU and minimal decrease in dice loss for epochs greater than 20, so it was decided 20 epochs was adequate for this proof of concept project.

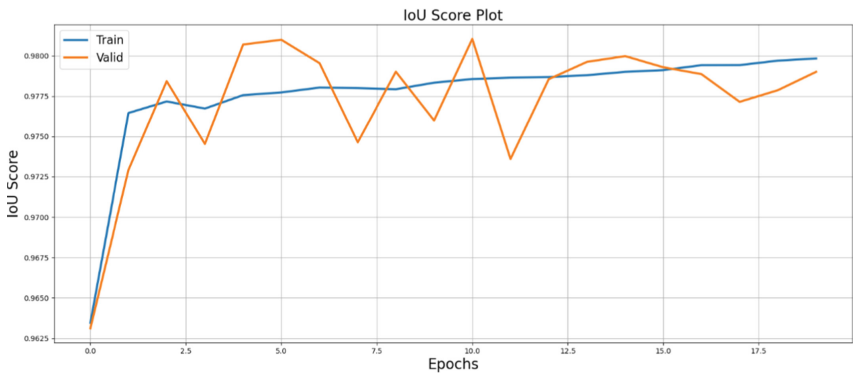


Fig. 6. IoU score plot

Figure 7 shows the dice loss plot over the 20 epochs as we can see loss drops quickly over the first few epochs and settles at the 0.01 mark. The high IoU score and low dice loss is possibly correlated to the low number of classes to segment and the large smooth boundary between the ROI and the ego-body which make it easier for the network to perform segmentation.

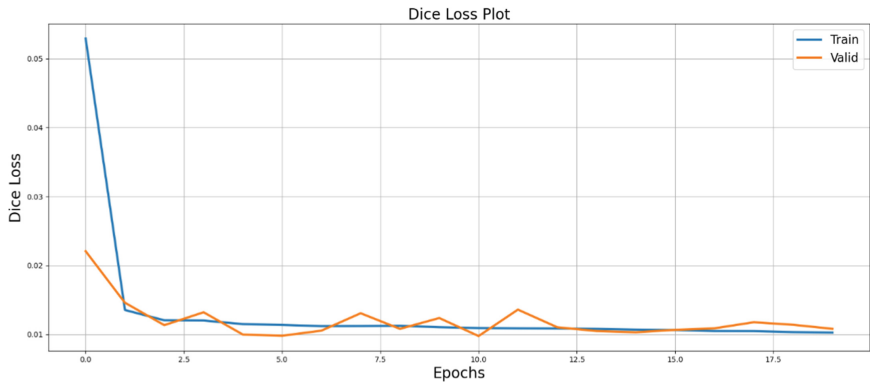


Fig. 7. Dice loss plot

The network model with the best validation IoU over the 20 Epochs achieved a IoU score of 0.981 and a dice loss of 0.01. This is a excellent IoU score which means highly accurate segmentation masks are being output from the network.

Table 1. Best network model

Type	Epoch	Batch size	Best IoU	Dice loss
Training	10	4	0.9786	0.01089
Validation	10	4	0.981	0.009736

Figure 8 shows the models inference run on unseen data. From left to right is the original image, the ground truth mask, the predicted mask and then the predicted mask overlay on the original image. From comparing the predicted masks and the ground truths in Fig. 8 it can observed that they are very close in appearance and there are not many misclassified pixels to be seen in the predicted mask.

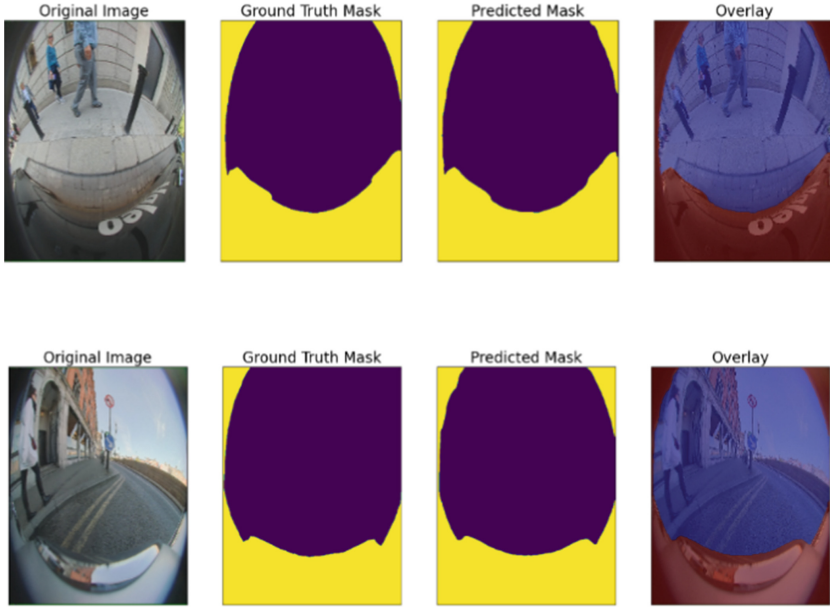


Fig. 8. Inference

6 Conclusion and Future Work

A simple binary semantic segmentation model was proposed in the paper to recognise the location of the vehicle ego-body in fish-eye format. Finding this information can be very useful and could potentially solve problems like false detections on the vehicle ego-body which would then improve overall vehicle safety, camera misalignment detection and reduce the amount of manual work it would take to find the ROI. The proposed model performed sufficiently well and the predicted masks it produces are of a high quality.

Future work would be to integrate the proposed system into a vehicle's main perception system. The system would be integrated like in Fig. 3 where the proposed network performs the semantic segmentation on the camera input, this is then post processed where the contours are extracted from the output and these contours are then passed on to the other perception algorithms which now have coordinates on the ROI that they should be focusing on. The proposed network could be run in set intervals or specific times to save computing power e.g. each time the car starts, when the car is shut off and in 2 min intervals while the car is on. Running the proposed network in short intervals and when the car is running, starts and shuts down serves the purpose of checking if the camera is misaligned. The system can store the previous contour values from the network and compare them with the new contour values and if they have changed over a certain threshold the system throws an error telling the driver that the camera is misaligned.

References

1. Kuutti, S., Bowden, R., Jin, Y., Barber, P., Fallah, S.: A survey of deep learning applications to autonomous vehicle control. *IEEE Trans. Intell. Transp. Syst.* **22**(2), 712–733 (2021). <https://doi.org/10.1109/TITS.2019.2962338>
2. Yogamani, S., Siam, M., Gamal, M., Abdel-Razek, M., Jagersand, M., Zhang, H.: A comparative study of real-time semantic segmentation for autonomous driving. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 587–597 (2018). <https://doi.org/10.1109/CVPRW.2018.00101>
3. Saez, A., Bergasa, L.M., Romeral, E., Lopez, E., Barea, R., Sanz, R.: CNN-based fisheye image real-time semantic segmentation. In: *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1039–1044 (2018). <https://doi.org/10.1109/IVS.2018.8500456>
4. Yogamani, et al.: WoodScape: a multi-task, multi-camera fisheye dataset for autonomous driving. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9307–9317 (2019). <https://doi.org/10.1109/ICCV.2019.00940>
5. Deng, L., Yang, M., Qian, Y., Wang, C., Wang, B.: CNN based semantic segmentation for urban traffic scenes using fisheye camera. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 231–236 (2017). <https://doi.org/10.1109/IVS.2017.7995725>
6. Bajcsy, R., Lee, S.W., Leonardis, A.: Detection of diffuse and specular interface reflections and inter-reflections by color image segmentation. *Int. J. Comput. Vis.* **17**, 241–272 (1996). <https://doi.org/10.1007/BF00128233>
7. DelPozo, A., Savarese, S.: Detecting specular surfaces on natural images. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2007). <https://doi.org/10.1109/CVPR.2007.383215>
8. Owen, D., Chang, P.L.: Detecting reflections by combining semantic and instance segmentation. *Umbo Comput. Vis.* (2019). <https://doi.org/10.48550/ARXIV.1904.13273>
9. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds.) *MICCAI 2015. LNCS*, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28, <https://doi.org/10.48550/arXiv.1505.04597>
10. He, K., Zhang, X., Ren, S., Sun, J. Deep residual learning for image recognition. *Microsoft Res.* (2015). <https://doi.org/10.48550/arXiv.1512.03385>
11. Yogamani, S., et al.: WoodScape: a multi-task, multi-camera fisheye dataset for autonomous driving. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9307–9317 (2019). <https://doi.org/10.1109/ICCV.2019.00940>
12. Cordts, M.: The cityscapes dataset for semantic urban scene understanding. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). <https://doi.org/10.1109/CVPR.2016.350>
13. Mariotti, L., Eising, C.: Spherical formulation of geometric motion segmentation constraints in fisheye cameras. *IEEE Trans. Intell. Transp. Syst.* **23**(5), 4201–4211 (2022). <https://doi.org/10.1109/TITS.2020.3042759>
14. Tesla Phantom breaking article. *The Verge* (2022). <https://www.theverge.com/2022/6/3/23153241/tesla-phantom-braking-nhtsa-complaints-investigation>. Accessed 20 Aug 2022
15. Nissan Emergency breaking failure article. *CNET* (2019). <https://www.cnet.com/roadshow/news/nissan-rogue-nhtsa-brakes-investigation/>. Accessed 20 Aug 2022

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Natural Language Processing and Recommender Systems



Recommendation Uncertainty in Implicit Feedback Recommender Systems

Victor Coscrato^(✉) and Derek Bridge

School of Computer Science and Information Technology,
University College Cork, Cork, Ireland
`vcoscrato@gmail.com, d.bridge@cs.ucc.ie`

Abstract. A Recommender System’s recommendations will each carry a certain level of uncertainty. The quantification of this uncertainty can be useful in a variety of ways. Estimates of uncertainty might be used externally; for example, showing them to the user to increase user trust in the abilities of the system. They may also be used internally; for example, deciding the balance of ‘safe’ and less safe recommendations. In this work, we explore several methods for estimating uncertainty. The novelty comes from proposing methods that work in the implicit feedback setting. We use experiments on two datasets to compare a number of recommendation algorithms that are modified to perform uncertainty estimation. In our experiments, we show that some of these modified algorithms are less accurate than their unmodified counterparts, but others are actually more accurate. We also show which of these methods are best at enabling the recommender to be ‘aware’ of which of its recommendations are likely to be correct and which are likely to be wrong.

Keywords: Recommender systems · Uncertainty · Neural networks

1 Introduction

Recommender Systems (RS) help users discover items in item catalogs. In general, for each user, the majority of the items are not relevant; therefore, the system’s task is to select and present personalized recommendation lists to each individual user. In most RS domains, the item catalog is huge, meaning that users will have interacted with only a tiny fraction of the items. This is known as *sparsity*. It means the system must infer the user’s preferences from a relatively small amount of information, leading sometimes to the generation of unsuccessful recommendations. Other factors that make it difficult to make good recommendations include changes in user mood and changes in user preferences over time. Due to the challenges of modelling ever-changing preferences from sparse feedback, there can be high uncertainty in the recommendations that an RS makes to its users [22].

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Artificial Intelligence under Grant No. 18/CRT/6223.

© The Author(s) 2023

L. Longo and R. O’Reilly (Eds.): AICS 2022, CCIS 1662, pp. 279–291, 2023.

https://doi.org/10.1007/978-3-031-26438-2_22

It is important for an RS to quantify its uncertainty. Estimates of recommendation uncertainty can help the RS detect which of its recommendations are more or less likely to be incorrect [16]. It can use these estimates in a variety of ways. The simplest is to expose them to the user: a recommendation can be accompanied by a number, a text or a visual that shows how sure or unsure the RS is that the user will like the recommendation. If a poor recommendation (one that the user does not like) is accompanied by a declaration of high uncertainty, for example, user trust may not be lost in the way that it might have been if there had been no declaration of uncertainty. As well as exposing uncertainty estimates to users, an RS may use the uncertainty estimates internally. For example, it may use them to decide how many uncertain recommendations to show (perhaps none when building trust, perhaps more when aiming for serendipity); or a hybrid RS may use them to decide when to call on different recommendation algorithms.

The literature on RS uncertainty quantification focuses almost exclusively on the rating prediction task, e.g. [22,31]. This is the case where the system has explicit feedback, usually in the form of numeric ratings (e.g. 1–5 stars) and where the task of the RS is to predict the rating for an unseen user-item pair. Of much more importance to the construction of usable RS is the top- K recommendation task and even more so in the case of implicit feedback, where the RS knows only which items a user has interacted with (e.g. which she has purchased or which she has clicked on). Yet there is almost no published research on uncertainty estimation in this more important setting. Not only that, but much of the work done on uncertainty estimation in the explicit rating prediction setting does not generalise to the implicit feedback setting. For example, in some explicit rating approaches, the ratings are assumed to follow a statistical distribution, whose dispersion is used to estimate uncertainty, e.g. [29]. This cannot generalise to the case where there are only implicit signals, with no rating values.

However, in other fields, dominated by neural models, such as computer vision [19] and natural language processing [30], methods for estimating the network’s predictive uncertainty have been explored. Among the most popular methods used for uncertainty quantification in neural networks are Bayesian Neural Networks (BNNs) [3], Monte-Carlo Dropout (MCDropout) [7] and Deep Ensembles [20].

We see an opportunity to adapt ideas from these other fields to the field of RS. This adaptation is made easier by the fact that the state-of-the-art in recommendation algorithms has changed in recent years. Now, several neural network-based recommenders have been proposed [4,15,23], with some achieving impressive results. Several properties of neural networks explain their increasing usage. First, the flexibility of neural architectures allows for a wide variety of data inputs, making it relatively straightforward to combine interaction data with item content data, user data and contextual data [13]. Second, neural networks are a great tool for latent representation learning, as shown by the success of variational autoencoder recommenders [21]. Up to now, uncertainty estimation in these neural RS remains unexplored.

In this paper, we introduce uncertainty estimation to implicit feedback RS. More specifically, we re-purpose several uncertainty estimation methods that

were successful in other tasks—either on explicit feedback recommenders or on neural networks in other fields—to make them suitable for the implicit recommender case. We compare these uncertainty estimation methods against each other, aiming to provide initial answers to two questions:

- Q1:** When implicit feedback RS are modified to perform uncertainty estimation, are there changes (gains or losses) in the accuracy of the RS?
Q2: Do the uncertainty estimates help the RS understand which of its recommendation are more likely to be right and which are more likely to be wrong?

The following Sect. 2 formulates the uncertainty estimation problem for implicit feedback RS. Section 3 proposes some techniques for solving this problem. Experiments are described in Sect. 4 along with their results. In Sect. 5, we review some related work that can be relevant to further work on RS uncertainty. Finally, Sect. 6 contains our conclusions.

2 Background

At its core, recommending is the task of selecting from many items those that are most relevant to the user. In this work, we focus on implicit feedback systems. In this case, the task of recommending can be seen as comprising at least two sub-tasks: first, estimating relevance scores for unobserved user-item interactions; and second, selecting the top- K items for a given user, guided by these relevance scores.

Formally, let $\mathcal{D} = \{(u, i) | u \in U, i \in I\}$ denote all the user-item pairs, where U and I are the set of users and items in the system, respectively. Users interact with items, e.g. purchasing them, clicking on them, and so on. We denote the set of observed user-item interactions by $\mathcal{D}^+ \subset \mathcal{D}$. Similarly, we denote the set of unobserved interactions $\mathcal{D}^- = \mathcal{D} - \mathcal{D}^+$. Then, for relevance scoring, the RS must learn a model $F_\theta(u, i) = r_{ui}$, parameterized by θ , from the observed interactions to predict the relevance of unseen user-item pairs.

With this setup, the implicit feedback task closely resembles a classification task, with observed interactions treated as ones, and non-observed ones treated as zeros. Therefore, the same objective functions used in classification tasks can be used [17]. In this work, we employ cross-entropy loss for every recommender. Nevertheless, we remark that ranking-based losses, such as Bayesian Personalized Ranking [27], could also be used instead, without affecting the uncertainty estimation methods that we will explore in Sect. 3.

The methods for uncertainty estimation herein can be applied to a wide class of recommender algorithms, that is, there are several possible choices for F_θ . Where possible, we will employ the well-known matrix factorization (MF) algorithm [17]. MF consists in learning a D -dimensional latent vector for each user and item. To predict r_{ui} , the user and item embeddings are combined, as follows,

$$F_\theta(u, i) = p_u^t q_i \quad (1)$$

where p_u and q_i are the user and item latent representations, respectively. In this case, $\theta = \{\{p_u\}_{u \in U}, \{q_i\}_{i \in I}\}$.

Furthermore, as explained in Sect. 1, we also want to use some uncertainty estimation methods that apply to neural models. For our neural recommender, we will use one of the simplest and most popular algorithms: He et al.’s Multi-Layer Perceptron (MLP) recommender [15]. In this case,

$$F_\theta(u, i) = \text{MLP}(p_u || q_i) \quad (2)$$

where $||$ is a concatenation operator and $\theta = \{\{p_u\}_{u \in U}, \{q_i\}_{i \in I}, \theta_{MLP}\}$. The MLP consists of a set of feed-forward layers f_1, \dots, f_L , such that,

$$f_0 = p_u || q_i \quad (3)$$

$$f_l = \text{ReLU}(W_l f_{l-1}), \quad \text{for } l \in 1, \dots, L-1; \quad (4)$$

$$f_L = \text{Sigmoid}(w_L^t f_{L-1}) \quad (5)$$

where W_l is the weight matrix for hidden layer l and w_L is the output layer’s weight vector. The Sigmoid activation in the output layer scales the output to $[0, 1]$.

For both the MF and the MLP, the parameters are learned by mini-batch gradient descent, minimizing the binary cross-entropy loss. On each training epoch, the training data consists of the observed interactions D^+ and an N -sized randomly-selected sample of non-observed interactions from D^- , where N is a hyperparameter.

We now turn our attention to uncertainty estimation methods. We use σ_{ui} to denote the uncertainty associated with the predicted relevance r_{ui} .

3 Uncertainty Estimation Methods

Recommendation uncertainty has several causes, including sparsity of data, modeling choices, and stochastic learning algorithms. For this reason, methods for uncertainty estimation in the field are very diverse. In this section, we present several methods for uncertainty estimation, making clear which recommender algorithms they can be used with.

One of the most notable sources of uncertainty is sparsity. For this reason, the amount of available data offers good baseline estimates of recommendation uncertainty [22]. Furthermore, these estimates can be used with any recommendation algorithm. In the past, they have been used for explicit feedback recommenders, but here we use them in the implicit setting. These estimates can be user-centric or item-centric. Hence, following [22], we define the following uncertainty metrics,

$$\text{NEG-USER-SUPPORT: } \sigma_{ui} = -\#u \quad (6)$$

$$\text{NEG-ITEM-SUPPORT: } \sigma_{ui} = -\#i \quad (7)$$

where $\#u$ and $\#i$ denote the number of observed interactions for the user and the item, respectively. The clear drawback of these uncertainty estimates is that they are either at user-level or at item-level, that is σ_{ui} is defined solely based on the user, or the item, but not on the user-item interaction. Nevertheless, they have the advantage of needing no additional learning and can be easily plugged into any system.

Beyond uncertainty introduced by the data, every recommender algorithm has its own uncertainty issues. Consider, for example, models that are based on representation learning, such as MF, where vector embeddings are learned as a latent representation for each user and item. For such models, the uncertainty surrounding the learning of such representations will affect the system recommendations. In fact, MF is known to suffer from learning instability [5, 25].

In the case of explicit feedback, ensembles have been successful at estimating the uncertainty of MF rating predictions [22]. But, explicit feedback MF is only one of many algorithms that can benefit from ensembling. In fact, an ensemble can be used to estimate uncertainty for any model that relies on a stochastic mechanism, such as random parameter initialization or stochastic learning protocols. This is the case for implicit feedback MF (Eq. 1) and also any neural network model, and in particular the MLP model (Eq. 2).¹

Formally, the principle is to train several models $F^{(k)}$, for $k = 1, \dots, n$ using a different random initialization each time, and then calculate interaction relevance and uncertainty as follows:

$$r_{ui} = \frac{\sum_{k=1}^n F_{\theta}^{(k)}(u, i)}{n} \quad (8)$$

$$\sigma_{ui} = \frac{\sum_{k=1}^n (F_{\theta}^{(k)}(u, i) - r_{ui})^2}{n} \quad (9)$$

Bayesian Neural Networks (BNN) are another major tool tailored to uncertainty quantification in neural models. BNNs differ from their deterministic counterpart by treating the parameters as random variables [10], which are assumed to follow some prior distribution $p(\theta)$. Given some training data \mathcal{D} , the posterior weight distribution, according to Bayes rule, is as follows,

$$p(\theta|\mathcal{D}) = \frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})} \quad (10)$$

Calculating the posterior directly from Eq. 10 is generally not possible, because the data evidence, $p(\mathcal{D})$, is unknown. For this reason, inference methods such as Monte-Carlo Markov Chains (MCMC) [9] and Variational Inference (VI) [11] are applied to approximate the exact posterior. More recently, Bayes By Back-propagation (BBB) has been proposed [3], a method that allows for the posterior weights distribution to be learned through back-propagation, just as the weights of a non-Bayesian network are learned by conventional back-propagation. Predictions can then be made using the estimated posterior.

¹ An ensemble of neural models is often referred to as a Deep-Ensemble [20].

Table 1. Methods we compare.

Name	Prediction model (F)	Uncertainty estimator
MF-NUS	MF (Eq. 1)	NEG-USER-SUPPORT (Eq. 6)
MF-NIS	MF (Eq. 1)	NEG-ITEM-SUPPORT (Eq. 7)
MF-Ensemble	MF (Eq. 1)	Ensemble (Eqs. 8–9)
MLP-NUS	MLP (Eq. 2)	NEG-USER-SUPPORT (Eq. 6)
MLP-NIS	MLP (Eq. 2)	NEG-ITEM-SUPPORT (Eq. 7)
BayesianMLP	MLP (Eq. 2)	Bayesian inference (BBB) (Eqs. 11–12)
MCDropout	MLP (Eq. 2)	Monte-Carlo Dropout (Eqs. 8–9)
MLP-Ensemble	MLP (Eq. 2)	Ensemble (Eqs. 8–9)

More precisely, the output’s expected value $\mathbb{E}[F_{\theta}(u, i)]$ is a point prediction for the interaction relevance r_{ui} , and its variance $\text{Var}[F_{\theta}(u, i)]$ is an estimate of relevance uncertainty σ_{ui} . In practice, the values are estimated using samples $\theta_k, \dots, \theta_k$ from the posterior, as follows,

$$r_{ui} = \frac{\sum_{k=1}^n F_{\theta_k}(u, i)}{n} \quad (11)$$

$$\sigma_{ui} = \frac{\sum_{k=1}^n (F_{\theta_k}(u, i) - r_{ui})^2}{n} \quad (12)$$

Another uncertainty estimation method that is tailored to neural networks is MCDropout [7]. The method, which can be thought of as an approximation of a Bayesian network, consists of taking multiple forward passes with dropout enabled at prediction time.² Formally, let $F^{(k)}$, for $k = 1, \dots, n$ denote k predictions calculated with dropout enabled. Then, the final estimates for relevance and uncertainty follow according to Eqs. 8 and 9.

4 Experiments

In this section, we compare the uncertainty estimation methods proposed in the previous section, with the goal of answering the two research questions raised in Sect. 1. More specifically, we will compare RS that combine MF and MLP presented in Sect. 2 with the uncertainty estimators discussed in Sect. 3. In Table 1, we list all the models and uncertainty estimation methods that we consider.

4.1 Datasets

We evaluate our models and uncertainty estimation methods on two popular datasets: an implicit version of the Movielens 1M dataset [14]³ and one Pinterest dataset [8]. Table 2 presents some summary statistics.

² Conventionally, dropout is enabled at training time and combats overfitting. In MCDropout, it is enabled at prediction time to sample a space of predictions.

³ To make this dataset implicit, we simply treat every given rating as an implicit signal (1), ignoring the numeric rating value.

Table 2. Datasets we use.

Dataset	Users	Items	Interactions
MovieLens	6040	3416	1 Million
Pinterest	55187	9643	1.5 Million

For both datasets, we use a user-based random data splitting method: 60% of the interactions for each user are for training, 20% for validation and 20% for testing.

4.2 Tuning

MF and MLP have hyperparameters that need to be chosen. First, we set the user and item latent embeddings size D to 128. Setting them to the same size gives a fair comparison. While it has been shown that both MF and MLP can benefit from even higher dimensions [28], $D=128$ gives us reasonable computational cost. Furthermore, to suppress the need to tune the number of training iterations, we employ early-stopping to end the learning phase when the MAP@5 (see Eq. 13) on the validation set does not improve for three consecutive iterations.

We tuned our models using a Bayesian parameter search, assisted by Optuna [1]. We train each model 20 times by sampling the hyperparameters from the following:

- For all models, learning rate is sampled from $[0.0001, 0.01]$ and N , the number of negative training instances per positive training instance, from $\{1, 2, \dots, 20\}$.
- MF: The L2 penalty factor applied to the user and item factors was taken from $[10^{-6}, 10^{-4}]$.
- MLP: We use the same three-layer MLP as in [15]. We also employ dropout on the training stage. The dropout rate is tuned in $[0, 0.2]$.
- BayesianMLP: We use a Bayesian MLP with the same architecture as our deterministic MLP. We use the same prior and tune the hyperparameters related to it across the exact same grid as used in [3].
- Ensemble and sample sizes: We use $n = 5$ in Eqs. 8–9. We experimented with larger and smaller sample sizes, but found they all produced similar results.

4.3 Evaluation

To evaluate a model’s recommendations, we obtain the top- K recommendation list for each user, which we denote by Z_u^K . These are the K candidate items that have highest predicted relevance score r_{ui} for the user. Candidate items exclude those that the user has interacted with in the training and validation sets; candidates are therefore items that either the user has not interacted with or items that the user has interacted with but the user’s interaction with the item is recorded in the test set.

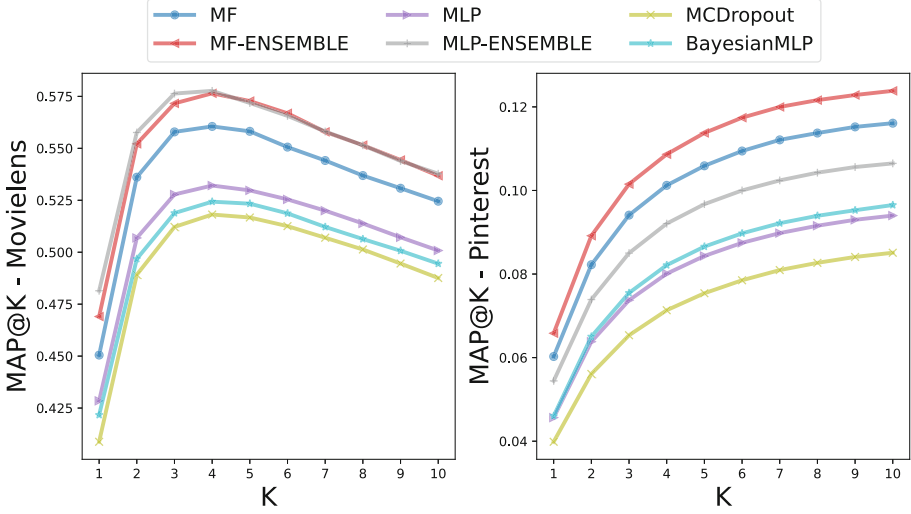


Fig. 1. $MAP@K$ for $K = 1, 2, \dots, 10$ for the Movielens (left) and Pinterest (right) datasets.

Let rel_u be the items that u has interacted with that are in the test set. Then, we evaluate a recommendation list according to its Mean Average Precision (MAP), averaged over all users:

$$MAP@K = \frac{1}{\#U} \sum_{u \in U} \sum_{j=1}^K Precision@j_u \times \delta(Z_u^K(j) \in rel_u) \quad (13)$$

where

$$Precision@j_u = \frac{\#\{Z_u^j \cap rel_u\}}{j} \quad (14)$$

and where $\delta(Z_u^K(j) \in rel_u) = 1$ if $Z_u^K(j)$, which is the j -th item in Z_u^K , is in rel_u and 0 otherwise.

4.4 Results

To answer **Q1** from Sect. 1, we compute the accuracy of the top- K recommendations for different recommendation list sizes $K = 1, 2, \dots, 10$. Figure 1 shows the $MAP@K$ obtained in both datasets. Note that MF-NUS & MF-NIS and MLP-NUS & MLP-NIS are omitted because their MAP is the same as MF or MLP.

The ensemble models, MF-ENSEMBLE and MLP-ENSEMBLE, show a remarkable MAP improvement over the baselines, MF and MLP. On the other hand, the BayesianMLP has similar performance to the deterministic MLP, and MCDropout has the worst performance on both datasets. Therefore, we found

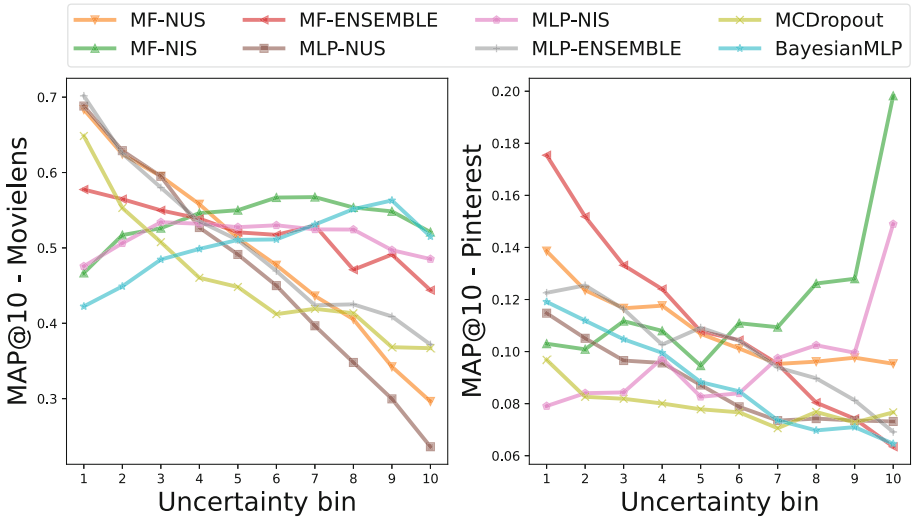


Fig. 2. MAP@10 for users grouped according to the average uncertainty of their recommendation. The higher the bin index, the higher is uncertainty.

that some models that perform uncertainty estimation improve accuracy, others worsen it. Clearly, ensembling emerges as the most beneficial method with respect to accuracy.

To answer **Q2**, we analyze the accuracy of the models for users, grouped according to their average recommendation uncertainty. More precisely, we calculated the average uncertainty on each user's recommendation list, and split the users into 10 equal-sized uncertainty bins, where bin 1 will have the 10% of users with the smallest average recommendation uncertainty and bin 10 will have those with the highest. Our intuition is that accuracy will fall as uncertainty grows. Figure 2 shows the results.

In line with our intuition, we see that MAP has a strong negative correlation with some of the uncertainty estimation methods. In particular, MF-Ensemble, MLP-Ensemble, MF-NUS and MLP-NUS appear to be those more strongly reflecting the expected behaviour. The results for NEG-USER-SUPPORT show that mature users tend to get more accurate recommendations. In fact, NEG-USER-SUPPORT has the strongest correlation to MAP in the MovieLens dataset. On the other hand, in the Pinterest data, the MF-Ensemble is the one to achieve the strongest correlation. This, together with the earlier results in Fig. 1, show that ensembling is not only a technique that can boost the accuracy of recommenders, but can also offer uncertainty estimates that correspond with the expected recommendation accuracy.

Other models to follow the expected behaviour are MCDropout in both datasets, and the BayesianMLP in the Pinterest dataset. Oddly, the BayesianMLP shows a growing MAP curve in the MovieLens data, meaning that users

with higher recommendation list uncertainties are getting higher accuracy, which is a result that needs further investigation.

Models using NEG-ITEM-SUPPORT do not show a very strong correlation between uncertainty and MAP with the exception of the last uncertainty bins in the Pinterest case. This too is a result that needs further investigation.

Largely, we believe that we have obtained a positive answer to **Q2**. For this research question, ensembling has, again, proven to be a great tool. Nevertheless, the simple and cheap NEG-USER-SUPPORT metric can also provide good value with no computational cost added, in contexts where user-centric estimates suffice.

5 Related Work

In this section, we briefly describe some related work that could be further explored in RS research.

Bernardis et al. showed that there is a strong correlation between the eigenvalues of the item similarity matrix and the accuracy of item-based recommenders [2]. Because of this, they propose an eigenvalue confidence index to measure the confidence level of the recommendations given to each user. Their method is suitable for both explicit and implicit recommendation tasks, and confidence can be thought of as the inverse of uncertainty. However, their method is applicable only to systems based on item similarity. Furthermore, like NEG-USER-SUPPORT, their confidence index is a user-centric measure, and therefore it lacks the granularity that is needed to differentiate the uncertainty of the individual items being recommended to a user.

Another method, which is superficially related to the ones employed herein but is actually quite different in its purpose, is the use of Gaussian embeddings for collaborative filtering. Gaussian embeddings are a generalisation of the embeddings used in many recommender algorithms. In [6], Gaussian embeddings are learned by a matrix factorization algorithm; in [18], they are learned by a convolutional neural model. Gaussian embeddings give us non-deterministic user and item representations, capturing the uncertainty that there is in learning these representations. However, they do not quantify the uncertainty of item relevance to a user.

Neupane et al. [24] propose a method for quantifying the amount of evidence available when providing recommendations to cold-start users. They propose a meta-evidential method for doing so. We believe that, in future research, uncertainty could be inferred in a similar way.

Finally, in the related field of Information Retrieval, Penha & Hauff explore uncertainty in neural learning-to-rank models [26]. They obtain uncertainty estimates for a BERT ranker with the usage of Monte-Carlo Dropout and Deep-Ensembles, which we explained earlier. Their uncertainty-aware ranking method combines the predicted interaction relevance with their estimated uncertainties. They found that ‘shrinking’ the relevance of interactions with high relevance can sometimes improve the system’s recommendation accuracy. In a similar vein, but

now using models based on Gaussian Processes, Guiver & Snelson proposed to either shrink or increase the relevance of items based on their uncertainty to make the model more conservative or more risk-taking [12]. However, their results are largely negative: they did not find this form of ranking to be more accurate. We leave the exploration of uncertainty-aware recommendation strategies similar to these for future research.

6 Conclusion

In this work, we explored methods for uncertainty estimation for implicit feedback recommender systems, exploring how the uncertainty estimates affect accuracy (Q1) and intelligibility regarding the recommender accuracy (Q2). Some of the methods had a positive impact on accuracy, others a negative impact. In particular, ensembling was the method showing the greatest accuracy improvements. Similarly, ensembling also was one of the top contenders when it came to correlation between accuracy and uncertainty, together with NEG-USER-SUPPORT, suggesting that these methods can help to identify which users are prone to receive the most or least accurate recommendations.

In addition, in the previous section, we highlighted some related work that can be useful for further exploration in the area. We hope that these, together with the promising results shown by our experiments will foment new research in this largely unexplored field.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)
2. Bernardis, C., Ferrari Dacrema, M., Cremonesi, P.: Estimating confidence of individual user predictions in item-based recommender systems. In: Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization, pp. 149–156 (2019)
3. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural networks. In: Proceedings of the 32nd International Conference on Machine Learning, pp. 1613–1622 (2015)
4. Cheng, H.T., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 7–10 (2016)
5. D’Amico, E., Gabbolini, G., Bernardis, C., Cremonesi, P.: Analyzing and improving stability of matrix factorization for recommender systems. *J. Intell. Inf. Syst.* **58**, 255–285 (2022)
6. Dos Santos, L., Piwowarski, B., Gallinari, P.: Gaussian embeddings for collaborative filtering. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1065–1068 (2017)

7. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1050–1059 (2016)
8. Geng, X., Zhang, H., Bian, J., Chua, T.S.: Learning image and user features for recommendation in social networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4274–4282 (2015)
9. Geyer, C.J.: Practical Markov chain Monte Carlo. *Stat. Sci.* 473–483 (1992)
10. Goan, E., Fookes, C.: Bayesian neural networks: an introduction and survey. In: Mengersen, K.L., Pudlo, P., Robert, C.P. (eds.) *Case Studies in Applied Bayesian Data Science*. LNM, vol. 2259, pp. 45–87. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-42553-1_3
11. Graves, A.: Practical variational inference for neural networks. In: *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pp. 2348–2356 (2011)
12. Guiver, J., Snelson, E.: Learning to rank with SoftRank and Gaussian processes. In: *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 259–266 (2008)
13. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: DeepFM: a factorization-machine based neural network for CTR prediction. In: *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1725–1731 (2017)
14. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. *ACM Trans. Interact. Intell. Syst.* **5**(4), 1–19 (2015)
15. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: *Proceedings of the 26th International Conference on the World Wide Web*, pp. 173–182 (2017)
16. Hernando, A., Bobadilla, J., Ortega, F., Tejedor, J.: Incorporating reliability measurements into the predictions of a recommender system. *Inf. Sci.* **218**, 1–16 (2013)
17. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: *Proceedings of the 8th IEEE International Conference on Data Mining*, pp. 263–272 (2008)
18. Jiang, J., Yang, D., Xiao, Y., Shen, C.: Convolutional Gaussian embeddings for personalized recommendation with uncertainty. In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 2642–2648 (2020)
19. Kendall, A., Gal, Y.: What uncertainties do we need in Bayesian deep learning for computer vision? In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5580–5590 (2017)
20. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: *Proceedings of the 31st International Conference in Neural Information Processing Systems*, pp. 6405–6416 (2017)
21. Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T.: Variational autoencoders for collaborative filtering. In: *Proceedings of the 2018 World Wide Web Conference*, pp. 689–698 (2018)
22. Mazurowski, M.A.: Estimating confidence of individual rating predictions in collaborative filtering recommender systems. *Expert Syst. Appl.* **40**(10), 3847–3857 (2013)
23. Naumov, M., et al.: Deep learning recommendation model for personalization and recommendation systems. [arXiv:1906.00091](https://arxiv.org/abs/1906.00091) (2019)
24. Neupane, K.P., Zheng, E., Yu, Q.: MetaEDL: meta evidential learning for uncertainty-aware cold-start recommendations. In: *Proceedings of the IEEE International Conference on Data Mining*, pp. 1258–1263 (2021)

25. Peña, F.J., et al.: Combining rating and review data by initializing latent factor models with topic models for Top-N recommendation. In: Proceedings of the 14th ACM Conference on Recommender Systems, pp. 438–443 (2020)
26. Penha, G., Hauff, C.: On the calibration and uncertainty of neural learning to rank models. [arXiv:2101.04356](https://arxiv.org/abs/2101.04356) (2021)
27. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint [arXiv:1205.2618](https://arxiv.org/abs/1205.2618) (2012)
28. Rendle, S., Krichene, W., Zhang, L., Anderson, J.: Neural collaborative filtering vs. matrix factorization revisited. In: Proceedings of the 14th ACM Conference on Recommender Systems, pp. 240–248 (2020)
29. Wang, C., et al.: Confidence-aware matrix factorization for recommender systems. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, pp. 434–442 (2018)
30. Xiao, Y., Wang, W.Y.: Quantifying uncertainties in natural language processing tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7322–7329 (2019)
31. Zhu, B., Ortega, F., Bobadilla, J., Gutiérrez, A.: Assigning reliability values to recommendations using matrix factorization. *J. Comput. Sci.* **26**, 165–177 (2018)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Graph-Based Diffusion Method for Top-N Recommendation

Yifei Zhou^(✉) and Conor Hayes

Data Science Institute, University of Galway, Galway, Ireland
y.zhou4@nuigalway.ie, conor.hayes@universityofgalway.ie

Abstract. Data that may be used for personalised recommendation purposes can intuitively be modelled as a graph. Users can be linked to item data; item data may be linked to item data. With such a model, the task of recommending new items to users or making new connections between items can be undertaken by algorithms designed to establish the relatedness between vertices in a graph. One such class of algorithm is based on the random walk, whereby a sequence of connected vertices are visited based on an underlying probability distribution and a determination of vertex relatedness established. A *diffusion kernel* encodes such a process. This paper demonstrates several diffusion kernel approaches on a graph composed of user-item and item-item relationships. The approach presented in this paper, *RecWalk**, consists of a user-item bipartite combined with an item-item graph on which several diffusion kernels are applied and evaluated in terms of *top-n recommendation*. We conduct experiments on several datasets of the *RecWalk** model using combinations of different item-item graph models and personalised diffusion kernels. We compare accuracy with some non-item recommender methods. We show that diffusion kernel approaches match or outperform state-of-the-art recommender approaches.

Keywords: Top-n recommendation · Web-mining · Random walk · Diffusion kernels

1 Introduction

The recommendation task has conventionally been cast as a matrix completion task - the data consists of sparse user-item matrix and the recommendation task is to make predictions for the missing elements. Multiple varieties of collaborative filtering (CF) have been proposed for this - nearest-neighbour based approaches (e.g. based on Pearson Correlation Coefficient), Matrix Factorisation [10] such as PureSVD and SVD++. More recently, He et al. [7] proposed a Neural Collaborative Filtering Framework and Ning et al. [14] proposed the SLIM method which builds a model using an elastic net regression approach. Item-based approaches convert the user-item matrix into an item-item matrix, typically using a measure of similarity, which can be mined for *if you liked that,*

you may also like this type recommendations. Both user-item, and item-item matrices are sparse and this has negative impact on recommendation coverage and quality.

However, user-item and item-item data can intuitively be modelled as a graph. With such a model, the task of recommending new items to users, or making new connections between items can be undertaken by algorithms designed to establish the relatedness between vertices in a graph. The sparsity issues are lessened due to the transitivity relations between vertices. One such class of algorithm is based on the random walk diffusion process, whereby a sequence of connected vertices are visited by random agents based on the transition probabilities of the links. The diffusion is a time-controlled stochastic process controlled by specific equations governing the traversal. As they move from vertex to vertex, the agents record a trace of their walk through the network that will reflect the probability of reaching a specific vertex from another vertex. We can identify patterns in the trajectories of multiple agents and use these to estimate the proximity or relatedness of the vertices in the graph. A *kernel* is a generalisation of such relatedness between vertices in the graph. The visitation decisions of the agents are underpinned by the rules of the diffusion. For instance, some diffusions will allow an agent to stop at random, or to restart the diffusion from another randomly selected vertex. Applying a kernel to network data provide a measures of similarity or relatedness between vertices that are not directly connected in the graph. Therefore, we can use kernels to make recommendations of related vertices based on an set of item vertices from a user profile.

The followings are the contributions of this paper. Firstly, we present a graph-based recommender method - *RecWalk**. Our method adopts the RecWalk approach of Nikolakopoulos Karypis [13] that combines a user-item interaction component with an item-item interaction component representing the similarities between items. Our approach uses a different weight initializing scheme built from the primary database. After that, we apply and evaluate several diffusion-based random walk algorithms on the network to discover the user-item or item-item relationships that are not found in the traditional item-based CF approaches. For instance, the standard item-based approaches only consider one-neighbour-connected items, while our proposed method will exploit two-hop or multi-hop neighbours within a network to improve recommendation accuracy. We apply three different diffusion kernels on eight public datasets and compare the performance with state-of-art recommender algorithms.

2 Notations and Definitions

In this paper, some notations are used to denote specific technical terms. Vectors, which are assumed to be column vectors, are represented by bold lower-case letters (e.g., \mathbf{p}). Matrices are denoted by bold upper-case letters (e.g., \mathbf{M}). Specifically, the i^{th} column and the j^{th} row of the matrix \mathbf{M} are depicted as \mathbf{p}_i and \mathbf{q}_j^T respectively. Besides, a boldface $\mathbf{1}$ is used to represent a column-wise vector where all values are ones. In addition, the diagonal values of a matrix \mathbf{M}

are denoted as $\text{Diag}(\mathbf{M})$. Furthermore, $\|*\|$ is used to denote the Euclidean norm. Sets are represented using the calligraphic upper-case letters (e.g. $\{\mathcal{U}, \mathcal{I}\}$). Finally, we use ‘=’ to express a definition statement.

A set of users (\mathcal{U}) and a set of items (\mathcal{I}) within a dataset are represented by $\{U_1, U_2, \dots, U_n\}$ and $\{I_1, I_2, \dots, I_m\}$ respectively. Let $\mathbf{R}^{U \times I}$ be the user-item matrix. Each entry of the matrix is the value of the user-item rating, provided that user (u) has rated the item (i); otherwise the entry is zero. Each user (u) is modelled by a row-vector $\mathbf{r}_u^T \in \mathbf{R}^I$, which is obtained from the user-item interaction matrix \mathbf{R} ; Similarly, each item (i) will be expressed as a column vector $\mathbf{r}_i \in \mathbf{R}^U$. Finally, the item-model is defined as a matrix $\mathbf{W} \in \mathbf{R}^{I \times I}$, which gives a measure of similarity or relatedness between items i and j .

3 Related Works

3.1 Item Models

Item models are one of the most popular and essential components used in collaborative recommender methods (e.g., FISM [8]). Such methods aim to build an item-item interaction matrix (\mathbf{W}) to capture the relations between items. An item model may also be represented as a graph in which pair of items are linked by their relatedness (e.g., similarity scores) in the item-item interaction matrix.

3.2 Random Walk

Graph-based approaches enable items that are not directly linked in the item-item graph to be considered as relevant recommendation candidates. Karypis and Nikolakakopoulos [13] propose a simple random walk (SRW) approach on an item-item graph. A *random walk* is a graph-based algorithm, defined as a stochastic process that begins a graph traversal at a vertex and moves to another connected vertex randomly at each time step with a probability proportional to the edge value in a transition probability matrix (\mathbf{P}) [6]. Karypis and Nikolakakopoulos’s approach applies a finite-step random walk on an item-item network starting with the item nodes rated by a user. The terminated state of SRW scores item nodes (excluding those the user has rated in the past) with cumulative landing probabilities. Items with the highest landing probabilities will be recommended to the user.

$$\omega_u^T = \frac{\mathbf{r}_u^T}{\|\mathbf{r}_u^T\|_1}, \mathbf{e}_u^T = \omega_u^T \mathbf{P}^K \quad (1)$$

Formula (1) above illustrates the linear mathematical operation form of the K-step SRW algorithm for a particular user (u). We use a state variable (\mathbf{e}_u^T) to record the landing probability of each item in the database after each step and use a row-vector (\mathbf{r}_u^T), which is obtained from the user-item matrix (\mathbf{R}), to represent the user’s past behaviour in the database. The start state vector (ω_u^T) is initialised as the normalised row-vector (\mathbf{r}_u^T). After each move, the new

state is updated by the product of the current state vector and the transition probability matrix (\mathbf{P}). The final state will be the product result of ω_u^T and the power (K) of \mathbf{P} .

3.3 Diffusion Kernels

Diffusion is a concept that refers to the net movement of a substance from an area of higher concentration to an area of lower concentration [9]. In the domain of computer science, a *diffusion kernel* is a matrix used to measure the relatedness or proximity between a pair of nodes within a graph. The relatedness score is based on the probability of information diffusing from one node to another, which is determined by a function known as a *kernel function*. There are different types of diffusion kernels [2], such as Markov-based kernels: Personalised PageRank (PPR), exponential-based kernels: Communicability (DR), and Laplacian-based kernels: Regularised Laplacian (LAP). Table 1 below describes three diffusion kernels (PPR, DR, and LAP) in detail. Specifically, the PPR and LAP diffusion kernels are given with both infinite series and linear forms as shown in their mathematical expressions Eq. (1.1), Eq. (1.2) and Eq. (1.5), Eq. (1.6), respectively. For the PPR kernel, \mathbf{S} is a stochastic matrix of an adjacency matrix (\mathbf{W}) with the PageRank scores between pair of nodes as shown in Eq. (1.3); k indicates the number of steps required in a random walk; p is a damping factor from 0–1 to control the strength and speed of energy propagation. DR is one of the exponential kernels that cannot be written in a linear form, thereby keeping the series form only in Eq. (1.4). The Laplacian matrix (\mathbf{L}) of an adjacency matrix is represented as the subtraction of the degree matrix (\mathbf{D}) and the adjacency matrix (\mathbf{W}) in Eq. (1.7).

Table 1. Diffusion kernels

Kernel Name	Expression
PPR	$\omega_u^T = e_u^T \sum_{k=0}^{\infty} (1-p)p^k S^k$ (1.1)
	$\omega_u^T = e_u^T (1-p)[I - pS]^{-1}$ ($0 < p < 1$) (1.2)
	$S = \text{Diag}(\mathbf{W}\mathbf{1})^{-1}\mathbf{W}$ (1.3)
DR	$\omega_u^T = e_u^T \sum_{k=0}^{\infty} p^k \frac{W^k}{k!}$ (1.4)
LAP	$\omega_u^T = e_u^T \sum_{k=0}^{\infty} p^k (-L)^k$ (1.5)
	$\omega_u^T = e_u^T [I + pL]^{-1}$ (1.6)
	$D = \text{Diag}(\mathbf{W}\mathbf{1})^{-1}; L = D - W$ (1.7)

4 Proposed Method

4.1 RecWalk*

This paper extends the *RecWalk* approach, a combined-random-walk framework, proposed by Nikolakakopoulos and Karypis [13] by adding and evaluating two different diffusion kernels on several real datasets. RecWalk* builds a network consisting of two sub-components: a user-item bipartite graph whose weights are initialised from a user-item matrix, denoted as (2), and an item-item interaction graph. Figure 1 illustrates the RecWalk* graph construction from a user-item co-rated table to a user-item combination graph.

$$\mathcal{G} : (\mathcal{V} = \{\mathcal{U}, \mathcal{I}\}, \mathcal{E}) \quad (2)$$

The adjacency matrix of \mathcal{G} is expressed as $\mathbf{A}_G \in \mathbf{R}^{(U+I) \times (U+I)}$ and is given by (3)

$$\mathbf{A}_G = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{bmatrix} \quad (3)$$

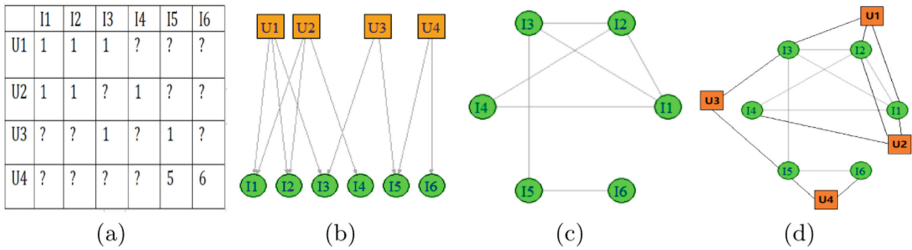


Fig. 1. RecWalk* model construction. (a) shows an example of a user-item co-rated matrix, and (b) is the graph representation of the matrix correspondingly. (c) illustrates an example of the item model built up from the user-item matrix in (a) using Cosine Similarity. The combination graph comprises a user-item-bipartite component and an item-item component, as shown in (d).

To investigate whether the network properties will influence the recommendation result, RecWalk* adopts a different weight-initializing strategy in the transition probability matrix of the user-item subgraph. RecWalk* defines a single bidirectional transition probability between a user node and an item node. This is the reciprocal value of the number of items rated by the user. This is a simplification of the original RecWalk weighting, which initialises a transition probability from a user node to an item node as well as from an item node to a user node.

There are two preconditions to build up our proposed framework. 1) A random walk always starts at a user node and ends with an item node; 2) The item-item component must be a connected graph.

Algorithm 1: RecWalk* Model**Input:** user-item co-rated matrix \mathbf{R} , item-model \mathbf{W} , parameter: a **Output:** \mathbf{P}

- 1 Construct \mathbf{M}_I : $\mathbf{M}_I = \frac{1}{W_\infty} \mathbf{W} + \text{Diag}(\mathbf{1} - \frac{1}{W_\infty} \mathbf{W} \mathbf{1})$
- 2 Construct \mathbf{A}_G : $\mathbf{A}_G = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{bmatrix}$
- 3 $\mathbf{R}_x = \frac{\mathbf{R}}{\mathbf{R} \mathbf{1}}$
- 4 $\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{R}_x \\ \mathbf{R}_x^T & \mathbf{0} \end{bmatrix}$
- 5 Construct \mathbf{P} : $\mathbf{P} = a\mathbf{H} + (1 - a) \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_I \end{bmatrix}$

Algorithm 1 shows the implementation of the transition probability matrix in the RecWalk* model. We adopt the same random walk strategy used in RecWalk. Each move is determined by a biased coin-toss. Assuming that the walker currently occupies a node $c \in \mathcal{U} \cup \mathcal{I}$, the next move is determined by a biased coin-toss that yields heads with probability (a) and tails with probability ($1 - a$). Provided that the current node (c) is a user node ($c \in \mathcal{U}$): i) if the coin-toss yields heads, the walker jumps to one of the items rated by the user randomly; ii) if the coin-toss yields tails, the walker stays put. While the current node (c) is an item node ($c \in \mathcal{I}$): i) if the coin-toss yields heads, the walker jumps to one of the users that have rated the item randomly; ii) if the coin-toss yields tails, the walker moves to a connected item node in the inter-item component. This algorithm accepts three inputs: a user-item co-rated matrix \mathbf{R} , an item model \mathbf{W} , a parameter ' a ' that denoted the biased probability. Lines 1–2 declare two adjacency matrices for the item-item component (\mathbf{M}_I) and the user-item component (\mathbf{A}_G), respectively. Lines 3–4 initialises the user-item transition probability matrix \mathbf{H} symmetrically, and line 5 constructs the final transition probability matrix \mathbf{P} by a linear combination of these two components with the biased probability a .

4.2 Communicability Kernel (DR)

The implementation of the communicability kernel (DR) with the RecWalk* model is shown in Algorithm 2. DR is an exponential-based kernel, thereby providing the truncated way only. The truncated way, described by Torres [16], means that a diffusion process finishes (truncates) the random walk after the required number of moves. Such an approach is computed by the iterative matrix multiplication and is applicable to all diffusion-based random walks. This algorithm accepts three inputs: a user ' u ' from all existing users in the database, a damping factor β used to control the diffusion intensity, and a constant number ' k ' indicating the number of moves of a random walk. Initially, a state variable $\boldsymbol{\lambda}^T$ is used to record the diffusion scores of all items after each move, and the primary state $\boldsymbol{\lambda}_{(0)}^T$ is a row-wise vector \mathbf{e}_u^T with the size of $\|\mathcal{U}\| + \|\mathcal{I}\|$ where the

Algorithm 2: RecWalk*+DR

Input: user $u \in \mathcal{U}$, damping factor: β , step number: k
Initialize: State Variable $\lambda^T : \lambda_{(0)}^T = e_u^T, e_u^T \in \mathcal{R}^{U+I}$
Transition Matrix $P : P = \text{RecWalk}^*(R, W, a)$
Output: ω_u^T

```

1  $t=1$ 
2 while  $t \leq k$  do
3    $\lambda_{(t)}^T = \lambda_{(t-1)}^T + \lambda_{(0)}^T \beta^t \frac{P^t}{t!}$ 
4   Normalise  $\lambda_{(t)}^T : \lambda_{(t)}^T = \frac{\lambda_{(t)}^T}{\sum \lambda_{(t)}^T}$ 
5 end
6  $\omega_u^T = \lambda_{(t)}^T$ 

```

element one on the position that corresponds to user u and zeros elsewhere. \mathbf{P} is the transition probability constructed by the RecWalk* model. Lines 2–5 give the update procedure of DR in each move followed by the Eq. (1.4) in Table 1, and the state variable is normalised after each iteration to ensure that all entries sum up to be 1. ω_u^T is the output vector that contains the final diffusion score of each item as shown in line 6.

4.3 Regularised Laplacian Kernel (LAP)

Algorithm 3 gives the details of RecWalk* with the regularised Laplacian diffusion kernel. The initialisation procedure is the same as the DR kernel. However, it lists two alternative ways: truncating the infinite series ('Truncated') or using the linear system ('Linear'). A linear solver is a powerful and direct way to reach the convergent state in the shortest time, but not effective with a large sparse matrix. Meanwhile, a truncated approach works for any type of kernel but would be memory-consuming for the large matrix computation in each iteration. In short, for both 'Truncated' and 'Linear' ways, each one has its own strengths and weakness. Lines 1–2 aim to obtain the Laplacian matrix (\mathbf{L}) of the transition probability matrix (\mathbf{P}). Lines 3–9 is the truncated implementation based on the Eq. (1.5) in Table 1 and lines 10–12 is the linear solver implementation based on the Eq. (1.6) in Table 1.

When a diffusion process finishes or reaches a convergent state, each item node will receive a final diffusion score. All items will be sorted by their diffusion scores in descending order, and those with top rankings which are not included in the rated item lists of the user will be returned as recommendation candidates.

5 Experiments

5.1 Datasets Preparation

Table 2 provides details of eight well-known rating datasets from different domains that have explicit and implicit feedback. The *Movielens!1M* [11] and

Algorithm 3: RecWalk*+LAP

Input: user $u \in \mathcal{U}$, Mod='Truncated', damping factor: β , step number: k
Initialize: State Variable $\lambda^T : \lambda_{(0)}^T = e_u^T, e_u^T \in \mathcal{R}^{U+I}$, Transition Matrix
 $P : P = \text{RecWalk}^*(R, W, a)$
Output: ω_u^T

```

1  $D = \text{Diag}(P1)$ 
2  $L = D - P$ 
3 if Mod=='Truncated' then
4   t=1
5   while  $t \leq k$  do
6      $\lambda_{(t)}^T = -\beta \lambda_{(t-1)}^T L + e_u^T$ 
7     Normalise  $\lambda_{(t)}^T : \lambda_{(t)}^T = \frac{\lambda_{(t)}^T}{\sum \lambda_{(t)}^T}$ 
8   end
9    $\omega_u^T = \lambda_{(t)}^T$ 
10 else if Mod=='Linear' then
11    $\omega_u^T = e_u^T [I + \beta L]^{-1}$ 
12   Normalise  $\omega_u^T : \omega_u^T = \frac{\omega_u^T}{\sum \omega_u^T}$ 
13 end

```

Yahoo!Movie [17] datasets represent film ratings. The Amazon product datasets provided by McAuley [1]: *Baby, Cell Phones and Accessories, Apps for Android*, and *Health and Care* represent consumer ratings. *Book-Crossing* (explicit rating version), collected by Ziegler [3] represents user rating of books. *Steam Video Game* represents users' purchasing records from a popular PC Gaming hub [15]. We applied some filtering to each dataset (except MovieLens) as the sparsity of some user and items records caused problems when sampling data for training and testing purposes. Our filtering approach ensuring that each user had at least three rated items, and each item was rated by at least by one user. Table 2 gives the statistics for the Unfiltered vs Filtered data.

5.2 State-of-Art Algorithms

We apply the RecWalk* model using three diffusion kernels (PPR, DR, and LAP) as functional kernels on two standard item-based models: (SLIM and NNCosine). State-of-the-art methods, such as Matrix Factorisation (PureSVD, EigenRec) and random walk (P^n), were evaluated as baselines.

Baseline

- **PureSVD**: one of the classical Matrix Factorisation recommender methods, aiming to reduce the dimensionality of the matrix [10]
- **EigenRec**: an extension of PureSVD that adds a scaling component for each item [12]
- **P^n** : an n -step random walk model on the user-item bipartite network, starting at a user node and ending with an item node where n must be an odd number [4]

Table 2. Statistics for filtered vs unfiltered data sets

	#User	#Item	N	ρ		#User	#Item	N	ρ
	MovieLens!1M					Yahoo!Movie			
Unfiltered	6,040	3,706	1,000,209	4.47%		7,642	11,916	221,367	0.24%
Filtered	6,040	3,706	1,000,209	4.47%		7,613	3,787	207,747	0.72%
	Baby					Apps for Android			
Unfiltered	531,890	64,426	915,446	0.0027%		1,323,884	61,275	2,638,172	0.0032%
Filtered	4,266	7,009	64,690	0.22%		3,297	7,040	106,985	0.46%
	Health and Care					Cell Phones			
Unfiltered	1,851,132	252,331	2,982,326	0.0006%		2,261,045	319,678	3,447,249	0.0005%
Filtered	3,615	8,214	79,757	0.27%		5,142	7,916	55,657	0.14%
	Book-Crossing					Steam Video Games			
Unfiltered	7,642	11,916	221,367	0.24%		12,393	5,155	129,511	0.20%
Filtered	3,716	7,256	85,370	0.32%		5,218	5,124	120,854	0.44%
Parameters Settings: #User and #Item: The number of users and items respectively. N: The number of interactions between users and items. ρ : The density level: $N/(User \times Item)$									

Item Model

- **NNCosine**: a neighbour-based similarity recommender method (ItemKNN) [5]
- **SLIM**: a recommender method which solves an item model by using ElasticNet Regression [14]

5.3 Evaluation Metrics

To evaluate the performances of our model, Leave-one-out Cross Validation (LOOCV) [13] was adopted in our experiments. For each user, an item rated by the user in the past was sampled and put into a test set (\mathcal{P}), and remaining data were put into a training set (\mathcal{T}). We repeated this process 3 times, and recorded the average. The standard sampling strategy, *Random Sampling* - One of the items rated by a user were selected randomly, was used to partition the dataset. For a particular user, we consider their corresponding test item together with 1000 randomly sampled unrated items. These 1001 items are ranked according to their prediction scores generated by each recommender method. To the experimental settings, we only consider the implicit feedback where the real value of rating was set to 1, and use Hit Rate (HR@10) [14] to show the accuracy of prediction result.

5.4 Running Environment

Our all experiments were implemented by using Python 3.8.8 and Numpy 1.21.3 running on Windows 10 operating systems.

6 Results

For baseline algorithms, experimental results showed that EigenRec performed consistently better than PureSVD, and P^n ($n = 3$) gained the best performance against all different n values¹. Therefore, EigenRec and P^3 were selected as the representative baseline algorithms for each dataset. For the RecWalk* model, we made experiments on two item models (NNCosine and SLIM) with three diffusion kernels (+PPR/+DR/+LAP) and reported results.

Table 3. Comparison of baseline (EigenRec and P^3), item (Cosine and SLIM) and diffusion-based approaches (PPR, DR, and LAP). Evaluation metric: HR@10. Paired T-test (SLIM and SLIM+LAP) at the significance level (0.05)

Dataset	Baseline		Item \ RecWalk* \ Diffusion								
			NNCosine				SLIM				
	EigRec	P^3	Base	+PPR	+DR	+LAP	Base	+PPR	+DR	+LAP	p value
MovieLens	43.51	24.29	27.10	33.41	33.53	39.62	44.45	45.08	45.07	45.10	6.73×10^{-7}
Yahoo!M	50.86	53.28	49.86	53.89	52.90	56.71	58.91	59.14	59.03	59.35	8.90×10^{-5}
AZ-Baby	14.60	13.55	10.41	14.67	14.96	15.14	15.35	15.82	16.06	16.10	5.77×10^{-5}
AZ-Apps	24.72	23.54	23.32	25.27	24.39	25.66	25.11	26.57	25.93	27.02	2.23×10^{-22}
AZ-Cell	21.88	23.40	16.78	24.41	25.71	26.14	25.59	26.41	26.18	26.64	1.53×10^{-7}
AZ-Health	22.74	22.49	18.06	22.27	22.68	24.04	24.18	24.51	24.45	25.64	1.18×10^{-22}
BX	14.83	16.47	7.67	16.82	16.25	17.12	19.35	20.05	19.99	20.16	9.29×10^{-8}
Steam	63.61	67.71	61.59	68.63	66.62	69.13	71.62	71.79	71.92	72.27	1.69×10^{-7}

Parameters Settings: RecWalk*: $\alpha \in \{0.005, 0.1, \dots, 0.9\}$. Diffusion Kernels: (+PPR/+DR/+LAP) $\beta \in \{0.1, \dots, 0.9\}$, DR (The step number) $k \in \{3, \dots, 10\}$. EigenRec: $f \in \{10, 20, \dots, 50\}$, $d \in \{0.1, \dots, 0.9\}$. P^3 : #Step = 3. NNCosine: (The nearest neighbour) $k=200$. SLIM: $l1 - norm \in \{0.025, 0.2, 0.25\}$, $l2 - norm = 0.0001$. **Bold** number: The best performance of diffusion-based recommender methods

Table 3 illustrates the results of baseline algorithms and item models with their diffusion-based RecWalk* in HR@10. The EigenRec and P^3 columns give the scores using the EigenRec and P^3 recommendation algorithms, respectively, as the baseline results. NNCosine and SLIM are two baseline item models, and the result of each algorithm are presented in the Base column. For each item-based approach, the results of three diffusion-based approaches are presented in the +PPR, +DR, and +LAP columns, respectively, and we use the bold font to mark the kernel with the best performance.

Table 3 shows that the diffusion-based approaches consistently outperform the baseline item models – NNCosine and SLIM – in terms of accuracy, and this difference is most noticeable in NNCosine. Despite a much smaller improvement for SLIM, a paired ‘T-test’ was conducted for all datasets and found that the difference was significant at a p value of the significance level of 0.05. (The p values for LAP are shown in the table). Therefore, SLIM+RecWalk*+LAP consistently performed better than SLIM. Furthermore, the diffusion-based approaches behaved better than EigenRec and P^3 in accuracy except for the MovieLens dataset.

¹ The full results are available at the appendix: <https://github.com/easternbob2019/>.

In addition, we compared the result (NDCG@10 [13]) of RecWalk* and RecWalk with NNCosine to examine whether the simplified weight initialising scheme of RecWalk* had any affect. Table 4 shows that RecWalk* and RecWalk performed similarly on PPR and DR, but that RecWalk* was better than RecWalk on LAP.

Table 4. Comparison of RecWalk* (+PPR/+DR/+LAP) and RecWalk (+PPR/+DR/+LAP) on NNCosine (MovieLens1M). Evaluation Meric: NDCG@10

Model (+NNCosine)	+PPR	+DR	+LAP
RecWalk	20.33	19.82	17.84
RecWalk*	19.60	19.98	22.25

7 Conclusion and Future Work

Diffusion approaches on networks have recently been used to determine relatedness between vertices [16] and to produce recommendations [13]. The work in this paper evaluates three different diffusion kernels on eight recommender datasets, represented as graphs. For each user in the data set, the diffusion process produces a ranking of related item nodes that are treated as recommendation candidates. Our experiments have determined that the diffusion kernel approaches are at least as good as state of the art of techniques for the top-n recommendation task, with the LAP diffusion kernel out performing other diffusion kernels.

Currently, our work focuses on the standard datasets only where the roles of users and items are well-defined with one type of action (e.g., Purchasing or Rating history). For each item in the database, we aim to add their semantic attributes as additional features so that the user would gain cross-domain recommendation results. Therefore, we will link each item to its corresponding entity node in a knowledge graph (e.g., DBpedia), extract a local subgraph and combine it with a CF-item model (e.g., NNCosine) as a semantic CF-item model. We will conduct experiments on the new semantic CF-item model with the RecWalk* and RecWalk model to examine the performances.

Acknowledgement. This publication describes research supported in part by a grant from Science Foundation Ireland under Grant number SFI/12/RC/2289_P2 and from the Hardiman scholarship provided by the College of Science and Engineering at the University of Galway. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

1. Aamazon product data (2018). <https://jmcauley.ucsd.edu/data/amazon/>
2. Avrachenkov, K., Chebotarev, P., Rubanov, D.: Kernels on graphs as proximity measures. In: Bonato, A., Chung Graham, F., Prałat, P. (eds.) WAW 2017. LNCS, vol. 10519, pp. 27–41. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67810-8_3
3. Book crossing (2022). <https://grouplens.org/datasets/book-crossing/>
4. Cooper, C., Lee, S.H., Radzik, T., Siantos, Y.: Random walks in recommender systems: exact computation and simulations. In: Proceedings of the 23rd International Conference on World Wide Web, pp. 811–816 (2014)
5. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 39–46 (2010)
6. Eksombatchai, C., et al.: Pixie: a system for recommending 3+ billion items to 200+ million users in real-time. In: Proceedings of the 2018 World Wide Web Conference, pp. 1775–1784 (2018)
7. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
8. Kabbur, S., Ning, X., Karypis, G.: Fism: factored item similarity models for top-n recommender systems. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2013)
9. Kondor, R.I., Lafferty, J.: Diffusion kernels on graphs and other discrete structures. In: Proceedings of the 19th International Conference on Machine Learning, vol. 2002, pp. 315–322 (2002)
10. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434 (2008)
11. Movielens 1m dataset (2022). <https://grouplens.org/datasets/movielens>
12. Nikolakopoulos, A.N., Kalantzis, V., Garofalakis, J.D.: Eigenrec: an efficient and scalable latent factor family for top-n recommendation, p. 47. arXiv preprint [arXiv:1511.06033](https://arxiv.org/abs/1511.06033) (2015)
13. Nikolakopoulos, A.N., Karypis, G.: Recwalk: nearly uncoupled random walks for top-n recommendation. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. pp. 150–158 (2019)
14. Ning, X., Karypis, G.: Slim: Sparse linear methods for top-n recommender systems. In: 2011 IEEE 11th International Conference on Data Mining. pp. 497–506. IEEE (2011)
15. Steam video games (2022), <https://www.kaggle.com/tamber/steam-video-games/data>
16. Torres-Tramón, P.: Diffusion-based models for semantic relatedness. Ph.D. thesis, NUI Galway (2020)
17. Yahoo!movie (2022), <https://webscope.sandbox.yahoo.com/>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





CouRGe: Counterfactual Reviews Generator for Sentiment Analysis

Diego Carraro^(✉)  and Kenneth N. Brown 

Insight Centre for Data Analytics, School of Computer Science and IT,
University College Cork, Cork, Ireland
{diego.carraro,ken.brown}@insight-centre.org

Abstract. Past literature in Natural Language Processing (NLP) has demonstrated that counterfactual data points are useful, for example, for increasing model generalisation, enhancing model interpretability, and as a data augmentation approach. However, obtaining counterfactual examples often requires human annotation effort, which is an expensive and highly skilled process. For these reasons, solutions that resort to transformer-based language models have been recently proposed to generate counterfactuals automatically, but such solutions show limitations.

In this paper, we present CouRGe, a language model that, given a movie review (i.e. a seed review) and its sentiment label, generates a counterfactual review that is close (similar) to the seed review but of the opposite sentiment. CouRGe is trained by supervised fine-tuning of GPT-2 on a task-specific dataset of paired movie reviews, and its generation is prompt-based. The model does not require any modification to the network's architecture or the design of a specific new task for fine-tuning.

Experiments show that CouRGe's generation is effective at flipping the seed sentiment and produces counterfactuals reasonably close to the seed review. This proves once again the great flexibility of language models towards downstream tasks as hard as counterfactual reasoning and opens up the use of CouRGe's generated counterfactuals for the applications mentioned above.

Keywords: Natural language processing · Sentiment analysis · Language models · Counterfactual reasoning · Data augmentation

1 Introduction

Under the framework of example-based reasoning [20], *counterfactual examples* are widely-adopted as a proxy for investigating causality relationships between events [16]. Their usefulness is well-established in the machine learning literature as they have been employed in many settings and domains, for example, to boost model generalisation, provide explanations and to enrich datasets (e.g. [7, 22, 23] respectively). In Sect. 2 we briefly review different types of counterfactuals but, in this work, we focus on counterfactuals in the Natural Language Processing (NLP) domain - specifically in sentiment analysis.

As a demonstrating and relevant example, consider the four textual movie reviews in Table 1. Literature has proposed approaches to generate counterfactual reviews of types **a**, **b** and **c** from the seed review **s**. Review **a** is a *task-specific* counterfactual because its generation is targeted to apply a specific different *counterfactual label* to the review, i.e. the negative sentiment. Generations of this kind can be found in [8, 14], for example. Instead, review **c** is a *general-purpose* counterfactual because its generation isn’t tailored to any downstream task, i.e. the sentiment label does not necessarily change¹. Generations of this kind can be found in [17, 27], for example.

Table 1. Example of a seed review **s** with three corresponding counterfactual reviews (**a**, **b**, **c**) where edits are highlighted in blue.

Id	Review	Sentiment	Generation type
s	“Titanic” is a good movie because of the original plot and the fascinating cast.	Positive	–
a	“Titanic” is a bad movie because of the expected plot and the low-performing cast.	Negative	Task-specific
b	“Titanic” is a bad movie because of the expected plot (really, I could predict every single minute of it, not kidding) and the horrible soundtracks .	Negative	Task-specific
c	“Titanic” is a good movie because of the original cast and the fascinating plot	Positive	General-purpose

A counterfactual review should be close to the seed review so that minimal changes allow causality assessments [16]. For example, while review **a** and **b** lead to the same negative sentiment, the former is much closer to **s** than the latter. In this paper, we focus on counterfactual reviews of type **a**, i.e. close to **s** but of different sentiment.

Also, generation can be manual or automatic (or hybrid [27]). When manual, human annotators are required to edit the seed review manually to generate counterfactuals. The editing process is generally accurate but expensive: human annotators are required to be “experts” in the task, and the effort dedicated to each generation can be quite high (e.g. 4–5 min in average [8]). Also, resorting to the manual approach might be a limitation in applications where online single-generation is required rather than offline batch-generation. On the other hand, automatic generation is generally cheaper and is fast enough to be suitable for interactive use, thus being appropriate for many modern data-hungry settings.

Although automatic generation is a way of obtaining a large number of cheap counterfactuals, we believe the approach is still under-investigated in the NLP domain. The most successful applications leverage recent progress on transformer-based [24] language models (LMs). By modification to the model’s

¹ When the generation is task-specific but the counterfactual label and the seed label are the same, the generated instance is known as semi-factual, e.g. the counterfactual explanations in [9].

architecture and/or fine-tuning, some works apply a controlled generation to a specific task, e.g. [14, 17] and some others to a specific part of the text, e.g. [21, 27]. Our solution to automatic counterfactuals generation is inspired in particular by [1, 17, 27] and targets the sentiment analysis task. Indeed we design a generator, which we name CouRGe, that, given a textual seed review and a counterfactual sentiment, produces a textual counterfactual review close to the seed review and displaying the target sentiment. We implement CouRGe by fine-tuning GPT2 [18] with a task-specific dataset of paired examples, and we leverage a prompt-based generation framework [12]. We run experiments² on a movie review dataset where we investigate different training scenarios for CouRGe. Results show that CouRGe can generate counterfactuals that belong to the target sentiment and that are diverse and fairly close to the seed review.

The remainder of the paper is structured as follows: Sect. 2 reviews related work in the literature; Sect. 3 outlines the counterfactuals generation framework we employ and describes how we train CouRGe; Sect. 4 presents the experiments and analyse results; and Sect. 5 draws conclusions and illustrates future plans.

2 Background and Related Work

2.1 Counterfactual Examples: Applications

Counterfactual examples have been used for a variety of goals: to explain the outputs of a model for increasing interpretability and trust for both users and AI practitioners in (e.g. [6, 7, 25]); to obtain more robust models that (hopefully) capture not only spurious correlation relationships, but also causal relationships between inputs and outputs of a model (e.g. [23, 26]); to increase fairness (e.g. [5, 10]); or simply for data augmentation purposes (e.g. [13, 28]).

Counterfactual and adversarial examples are related but different in nature [3]. Indeed, adversarial examples (also known as adversarial attacks) are test inputs created with the purpose of *fooling* a model to misclassify such inputs. They are designed with the specific goal of testing the robustness of a model to unexpected and out-of-distribution inputs. Also, counterfactuals are used to test a model in some settings (e.g. [4, 14]), but their use is more related to the interpretability and the analysis of the causal effects between the inputs and the outputs of the model [3]. Although generation algorithms in the literature work with similar principles for both counterfactuals and adversarials, the former typically hold additional properties such as *plausibility* (i.e. generated examples are realistic and in-distribution) and *human-perceptibility* (i.e. changes on the generated examples need to be perceptible by a human evaluator) [14, 28].

2.2 On Generating Counterfactuals for NLP

In the NLP domain, manual approaches to generate counterfactuals have been proposed, for example, in [4, 8, 17]. Similarly, the authors employ human crowd

² The code is available at <https://github.com/cdiego89phd/counterfactuals-generation>.

workers to generate counterfactual reviews from original textual movie reviews. This editing process instructs workers to apply minimal perturbations to the seed text (i.e. closeness constraint) but at the same time ensure that the generated text remains coherent and fluent (i.e. coherence-fluency constraint) and that the counterfactual label applies (i.e. label-flip constraint, when applicable). Generations of this kind are generally very expensive and often impractical: for this reason, in this paper we propose a cheaper alternative, i.e. automatic generation. In the remainder of this section, we review literature that is closest to and inspired our work.

PPLM [1] and GYC [14] are LM-based tools able to generate text entailed to one or more controllable attributes, such as class labels, for example. In practice, the generation is controlled by specific attribute models that are plugged in on top of the LM so that the generation does not require any further training of the LM. While GYC is designed to produce counterfactuals from a seed text, PPLM is a general-purpose text generator. MiCE is a tool that resorts to a two-stage process to generate counterfactuals as a proxy for interpretability [21]. In the first step, MiCE identifies portions of the seed text that are associated with the example’s label; in the second step, such portions are minimally perturbed to obtain a text matching a specific counterfactual label. POLYJUICE [27] is a general-purpose conditional counterfactual generator for text sentences. It is a GPT-2 version fine-tuned on various paired-sentences datasets that allow for control over perturbation types and locations through pre-defined control codes. Finally, Counterfactual Story Rewriting (CSR) is a system able to perform counterfactual narrative reasoning and revision by fine-tuning an LM with a task-specific dataset [17].

CouRGe is inspired by PPLM, GYC and MiCE because generation is controlled towards a specific label; it is close to CSR because the training is performed with a task-specific dataset (and we propose a different training scenario); and it uses prompting, which resembles the use of control codes in POLYJUICE.

3 Training CouRGe

3.1 Framework

Our goal is to build a generator G with parameters θ , i.e. G_θ , able to perform the following task: *given a seed review with its sentiment label and a counterfactual target sentiment, generate a counterfactual review as close as possible to the seed review and of target sentiment*. More formally, given a seed review x of sentiment s and a counterfactual opposite sentiment \bar{s} , we require G_θ to learn the function g_θ , that returns the counterfactual review \hat{x} , as close as possible³ to x and of sentiment \bar{s} :

$$g_\theta(x, s, \bar{s}) = \hat{x} \quad (1)$$

where a sentiment is either positive ($s, \bar{s} = 1$) or negative ($s, \bar{s} = 0$).

³ We use different distance metrics to measure the closeness, see Sect. 4.

3.2 Training Scenarios

In this section, we describe different training scenarios for our task. We use two variants of the GPT-2 pre-trained language model [18] as base models, i.e. GPT2 and GPT2-m (124 and 355 million parameters respectively), leading to 12 different trained model versions. However, such training scenarios are general, and other pre-trained models could be used with little modification (e.g. the BERT family [2], the T5 family [19]). In some training scenarios below, we also assume the availability of a dataset of n paired reviews $\mathcal{D} = \{x_i, s_i, \bar{x}_i, \bar{s}_i\}$ where x is a seed review with sentiment s_i and a ground truth counterfactual review \bar{x}_i with sentiment \bar{s}_i (we will use the counterfactually-augmented dataset from [8]).

Zero-Shot (ZS). There is no training in this scenario, i.e. we employ GPT2 and GPT2-m to assess the generation capabilities that these models gained from the pre-training.

Unsupervised Fine-Tuning (UFT). In this scenario, we expose GPT2 and GPT2-m to a movie-specific corpus to drive the models’ text generation toward the target domain and vocabulary (sometimes, this type of training is also known as continual pre-training). In this setting, the model is fine-tuned to maximize the log-likelihood of the reviews in the corpus C :

$$\mathcal{L}^{UFT}(\Theta) = \log g_{\theta}(C) \quad (2)$$

Supervised Fine-Tuning (SFT). We use the task-specific dataset from [8] (and formally described in Sect. 3.1) to fine-tune GPT2 and GPT2-m so that the text generation will be specific to our task. Informally, this setting is equivalent to a supervised scenario where ground-truth counterfactual reviews are the target labels. We perform prompt-based fine-tuning [12], where we design two specific manual prompts. The log-likelihood is the following:

$$\mathcal{L}^{SFT}(\Theta) = \log g_{\theta}(f_{pt}(x, s, \bar{x}, \bar{s})) \quad (3)$$

and f_{pt} is a function that encapsulates the input into the prompt (Table 2).

Unsupervised and Supervised Fine-Tuning (UFT + SFT). In this scenario, we sequentially combine UFT first (Eq. 2) and SFT afterwards (Eq. 3), in order to leverage the advantages of both training steps.

3.3 Generation Step

At generation time, we feed the models from scenarios ZS and UFT with s, x, \bar{s} (separated by the special separation token [SEP]) and we ask them to generate \bar{x} . For scenarios SFT and (UFT + SFT) we apply prompt-base inference so that we query the models with the encapsulated input $f_{pt}(x, s, \bar{s})$ to generate \bar{x} .

Table 2. The close prompts used for training and generation. The design of P1 and P2 is inspired by [12]. To note, we fill the sentiments s and \bar{s} with the strings accordingly to the sentiment map reported. Also, we use special tokens in square brackets for the prompts: [SEP] is a separator; [BOS] and [EOS] indicate the beginning and the end of the generation, respectively.

Id	Prompt (f_{pt})	Sentiment map	
		$s, \bar{s} = 1$	$s, \bar{s} = 0$
P1	“[BOS] s review:[SEP] x [SEP] \bar{s} review:[SEP] \bar{x} [EOS]”	“Positive”	“Negative”
P2	“[BOS]The movie is s . [SEP] x [SEP] The movie is \bar{s} . [SEP] \bar{x} [EOS]”	“good”	“bad”

4 Experiments

4.1 Datasets Preprocessing

Because our target domain is the movie domain, for the UFT setting, we use the Rotten Tomatoes movies and critic reviews dataset⁴. We randomly split the dataset into training and validation sets (with 80%-20% ratio).

CAD-IMDb⁵ is the movie reviews dataset we employ for the SFT scenario. The dataset accounts for 2440 examples: each example is a pair of reviews where one review is the seed review x and the other is the counterfactual review \bar{x} ⁶. We randomly split the dataset into training, validation and test sets (with 70%-12%-18% ratio).

4.2 Experimental Methodology

When training the different versions of CouRGe in the various scenarios, we use the validation set to tune the hyperparameters (we optimise for the perplexity metric [18] with early stopping); we consider the tuning of the learning rate, weight decay, adam epsilon, warmup steps and accumulation steps.

After a model is trained, i.e. at test time, we run the generation step (Sect. 3.3) three times, so that the model generates three counterfactuals for each seed review in the test set. Similarly, we perform the generation step for the baseline models (see details in the next section) and obtain three counterfactuals per seed review in the test set. For the baselines and our CouRGe models, we randomize the generation so that, instead of selecting the next token with

⁴ <https://www.kaggle.com/datasets/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset>.

⁵ <https://github.com/acmi-lab/counterfactually-augmented-data/tree/master/sentiment>.

⁶ It is not clear which of the two reviews is the original and which one is the manually-crafted counterfactual: we randomly assign one review to be the seed review and the other to be the counterfactual review.

the highest probability, we select among multiple tokens with the highest probability. After the generation is completed, we assess the performances of each generator, computing the metrics described in Sect. 4.4.

Tuning of Generation’s Hyperparameters. At the generation step, LMs can control the generation by setting hyperparameters such as the number of beams, repetition penalty, n-gram repetitions, top-k and top-p. To assess the impact of such hyperparameters, we run further experiments (denoted by SFT*) where we take the models from the SFT scenario and we tune hyperparameters on the validation set before running the generation (and we optimize for BLEU, see Sect. 4.4).

Out-Of-Domain (OOD) Test. To assess the generalisation capabilities of our generator, we evaluate CouRGe on two additional test sets, i.e. movies’ reviews⁷ from the IMDb website and businesses’ reviews⁸ from the Yelp website.

4.3 Baselines

Among the generators presented in Sect. 2, we selected two baseline generators to compare the performances of our CouRGe. We resort to the trained models made available in their repositories and do not perform any hyperparameter tuning (we use the default values).

PPLM [1]: for each seed review in the test set, PPLM uses a context, a Bag of Words (BoW) and a sentiment discriminator to generate a counterfactual. The context is the first three words of the seed review (similarly to [14]); the BoW is composed of the words in the seed review; and the discriminator guides the generation towards the counterfactual label.

POLYJUICE [27]: we run the generator on the full-automatic setting. Thus, for each seed review in the test set, we randomly select k sentences to perturb. Each of the selected sentences is entirely blanked (which means that we randomly select the perturbation type), leaving the rest of the seed review as it is. To note, POLYJUICE has been trained with the same task-specific dataset presented in Sect. 4.1 (including the test set portion), which is a considerable advantage over PPLM and our CouRGe.

We do not employ GYC [14] and MiCE [21] as baselines for our experiments. Regarding the former, there is no open implementation available, and its approach is similar to PPLM. We omit the latter because its generation process would unfairly favour the performances on the LFS metric (see next section).

⁷ The polarity dataset v2 at www.cs.cornell.edu/people/pabo/movie-review-data.

⁸ https://huggingface.co/datasets/yelp/_polarity.

4.4 Evaluation Metrics

We evaluate each generator by applying a wide range of automatic metrics that measure the generated counterfactuals’ effectiveness, closeness and diversity. For each metric below, we first average the metric scores across the three generated counterfactuals and then across all the test instances.

Effectiveness. Ensure that the counterfactual label applies to the generated text. We choose to employ the Label-Flip Score (**LFS**), which scores 1 when the counterfactual sentiment is the opposite of the seed sentiment. To predict each label, we use a version of DistilBERT, a sentiment classifier fine-tuned on the SST-2 sentiment dataset⁹ (selected as the most accurate classifier among different candidates through a small experiment run on the CAD-IMDb of [8]).

Closeness. We measure Levenshtein edit distance (**LEV**) [11] and the syntactic closeness with the tree-edit distance (**TED**) [29], and we do that by comparing each counterfactual with its corresponding seed review. Also, we compute corpus-level **BLEU** from Papileni et al. [15], widely-used to measure the performance of translation machines, which calculates the overlap between the generated counterfactuals and their respective reference counterfactuals in the test set.

Diversity. We use the Self-BLEU (**S-BLEU**) proposed by Zhu et al. [30]. For each seed review, we compute the metric between the three corresponding counterfactuals (the lower the metric’s value, the better).

4.5 Results

The first set of results is reported in Table 3. POLYJUICE’s counterfactuals (when $k = 2$) are close to their seed review (best performance for LEV and BLEU) and diverse, but they are not effective (worst performance for LFS). This is as expected, considering the nature of the generator. Indeed, because POLYJUICE’s counterfactual reasoning is applied at a sentence level, then closeness is ensured (perturbations are minimal); at the same time, there is no such reasoning at an inter-sentence level, which makes the label flip difficult to achieve for multi-sentences reviews. For $k \in \{3, 4\}$ we have similar outcomes. When $k = 1$, closeness metrics improve (e.g. LEV= 0.09, TED= 10.1) but LFS drops to 0.19. (Results for $k \in \{1, 3, 4\}$ are not reported due to space constraints.)

PPLM’s performances are surprisingly low: despite PPLM being able to control the sentiment and the content of the generated text, it fails to generate good counterfactuals accordingly to all the metrics (except for diversity). A possible explanation is that we do not tune the extensive range of the model’s hyperparameters. We leave this task for future work.

⁹ <https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english>.

Table 3. Results of the evaluation, where the test set is composed by 488 instances. We do not report performances for the ZS scenario, as they are very similar to the ones in UFT. For POLYJUICE, we report results for $k = 2$, being the version with the highest LFS. In bold, we highlight the best-performing value of each metric.

Model	Training scenario	Prompt	LFS \uparrow	LEV \downarrow	TED \downarrow	BLEU \uparrow	S-BLEU \downarrow
POLYJUICE-2	–	–	0.27	0.18	17.3	0.71	0.84
PPLM	–	–	0.44	1	59.2	0.01	0.07
CouRGe-GPT2	UFT	–	0.54	1	70.1	<0.01	0.28
CouRGe-GPT2-m	UFT	–	0.53	1	68.8	<0.01	0.26
CouRGe-GPT2	SFT	P1	0.88	0.3	23.6	0.45	0.84
CouRGe-GPT2	SFT	P2	0.88	0.3	25.2	0.44	0.83
CouRGe-GPT2-m	SFT	P1	0.89	0.32	23.5	0.43	0.83
CouRGe-GPT2-m	SFT	P2	0.87	0.31	23.2	0.43	0.82
CouRGe-GPT2	UFT+SFT	P1	0.85	0.3	26.7	0.45	0.84
CouRGe-GPT2	UFT+SFT	P2	0.85	0.35	28.3	0.39	0.79
CouRGe-GPT2-m	UFT+SFT	P1	0.88	0.32	25.2	0.43	0.84
CouRGe-GPT2-m	UFT+SFT	P2	0.85	0.35	28.3	0.4	0.88
CouRGe-GPT2	SFT*	P1	0.84	0.2	15.8	0.57	0.89
CouRGe-GPT2	SFT*	P2	0.85	0.22	17.3	0.54	0.88
CouRGe-GPT2-m	SFT*	P1	0.87	0.23	16.5	0.54	0.89
CouRGe-GPT2-m	SFT*	P2	0.87	0.23	16.4	0.55	0.85

Results for the training scenarios ZS and UFT of CouRGe (we only report the latter as they are similar to the former) show that counterfactual reasoning is a challenging task that cannot be successfully addressed without proper fine-tuning. In particular, performances are poor accordingly to all metrics, even when the LM is shifted towards the domain-specific distribution (UFT scenario).

For the SFT scenario, CouRGe produces effective and reasonably close counterfactuals (best value for LFS while BLEU is the metric where performance is not outstanding). Disproving what is found in [17], models trained in the (UFT+SFT) do not benefit from the UFT training, as results are very similar to the ones in SFT. As expected, when we optimize for closeness, performances improve for LEV, TED and BLEU, while LFS suffers a small drop. Also, diversity is relatively poor in all scenarios (and it is comparable to POLYJUICE’s diversity). As a final remark on Table 3, CouRGe built on GPT2-m does not perform better than the one built on GPT2 and training with the two different prompts also leads to similar performances, contrary to what is found in [17].

We also found that CouRGe can generalise fairly well on unseen and out-of-domain data, see Table 4. This is true in particular for the out-of-domain Yelp test, where performances are comparable to the ones reported in Table 3. For the IMDb test, performance degrades despite the fact that reviews are in the same movie domain used for training CouRGe. A possible cause for this is the average length of the seed review given as input to the generator, which is significantly higher than the one in Yelp or in the training set (i.e. 901 characters).

Table 4. Results of the ODD evaluation, where each test set is composed of 250 instances. We employ the best performing model in terms of LFS, i.e. CouRGe-GPT2-m from SFT. We do not measure BLEU as reference counterfactuals are not available in the datasets.

ODD test	Avg. seed review length	LFS	LEV	TED	S-BLEU
IMDb	3892 chars	0.66	0.84	71.2	0.43
Yelp	723 chars	0.81	0.35	21.0	0.77

Table 5. Average computational time for each model’s generation. Experiments were run on a NVIDIA A40 48 GB GPU.

Model	Generation time
POLYJUICE-2	2 s per seed review
PPLM	164 s per seed review
CouRGe-GPT2	11 s per seed review
CouRGe-GPT2-m	13.77 s per seed review

Also, Table 5 reports the average times spent by the models for generating the three counterfactuals from the seed review: PPLM takes the largest amount of time and therefore, its generation can only fit batch/offline settings. Instead, the other three might be suitable for both online and offline settings (in particular, POLYJUICE stands out with 2 s per review).

5 Conclusion and Future Work

In this paper, we have designed and trained CouRGe, a GPT2-based text generator able to generate counterfactual reviews for the sentiment analysis task. We have proven that GPT2 is an excellent learner because it can be fine-tuned to perform counterfactual reasoning with no modifications to the training procedure or the model’s architecture. Based on our experiments that compare CouRGe with PPLM and POLYJUICE (two state-of-the-art generators), our model is much more effective (i.e. the counterfactual label applies more often), while closeness and diversity are comparable or better than the ones shown by POLYJUICE (the best baseline for these metrics). One limitation of CouRGe is the computational expense in terms of time. Indeed, despite being an order of magnitude faster than PPLM on average for a single instance generation, our model might not be suited to operate in some online settings but only in offline settings. Also, we are aware that our automatic evaluation should be complemented with a proper manual evaluation, as done in [14, 27], for example. We leave the investigation to reduce the computational time and the manual evaluation as future work.

To further improve CouRGe’s counterfactual reasoning, a few options are available. For example, we could look into prompt engineering, i.e. design

further manual prompts and automatic prompts [12]. Also, because our training framework enjoys generality, we could employ bigger language models from the GPT family (e.g. GPT3); or employ different families of models such as T5 [19] and BERT [2] in place of GPT2.

This work can be extended in some other ways. For example, we might use CouRGe’s counterfactuals to augment the training set of a sentiment classifier and increase generalisation (like in [8, 27]); we could reproduce the same study of this paper, but framed for a different downstream task like Natural Language Inference (similarly to what is done in [8] for example).

Acknowledgements. This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2 which is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

1. Dathathri, S., et al.: Plug and play language models: a simple approach to controlled text generation. In: International Conference on Learning Representations (2020)
2. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018)
3. Freiesleben, T.: The intriguing relation between counterfactual explanations and adversarial examples. *Minds Mach.* **32**(1), 77–109 (2022)
4. Gardner, M., et al.: Evaluating models’ local decision boundaries via contrast sets. arXiv preprint [arXiv:2004.02709](https://arxiv.org/abs/2004.02709) (2020)
5. Garg, S., Perot, V., Limtiaco, N., Taly, A., Chi, E.H., Beutel, A.: Counterfactual fairness in text classification through robustness. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, pp. 219–226 (2019)
6. Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., Lee, S.: Counterfactual visual explanations. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 2376–2384. PMLR (2019)
7. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. In: Data Mining and Knowledge Discovery, pp. 1–55 (2022)
8. Kaushik, D., Hovy, E., Lipton, Z.: Learning the difference that makes a difference with counterfactually-augmented data. In: International Conference on Learning Representations (2019)
9. Kenny, E.M., Keane, M.T.: On generating plausible counterfactual and semi-factual explanations for deep learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 11575–11585 (2021)
10. Kusner, M.J., Loftus, J., Russell, C., Silva, R.: Counterfactual fairness. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017)
11. Levenshtein, V.I., et al.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet Physics Doklady, vol. 10, pp. 707–710. Soviet Union (1966)

12. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. arXiv preprint [arXiv:2107.13586](https://arxiv.org/abs/2107.13586) (2021)
13. Liu, Q., Kusner, M., Blunsom, P.: Counterfactual data augmentation for neural machine translation. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 187–197 (2021)
14. Madaan, N., Padhi, I., Panwar, N., Saha, D.: Generate your counterfactuals: towards controlled counterfactual generation for text. In: AAAI (2021)
15. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: ACL (2002)
16. Pearl, J., Mackenzie, D.: The Book of Why: The New Science of Cause and Effect, 1st edn. Basic Books Inc., New York (2018)
17. Qin, L., Bosselut, A., Holtzman, A., Bhagavatula, C., Clark, E., Choi, Y.: Counterfactual story reasoning and generation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (2019)
18. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
19. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**(140), 1–67 (2020)
20. Rissland, E.L.: Example-based reasoning. In: Informal Reasoning and Education, pp. 205–226. Routledge (2012)
21. Ross, A., Marasović, A., Peters, M.E.: Explaining nlp models via minimal contrastive editing (mice). arXiv preprint [arXiv:2012.13985](https://arxiv.org/abs/2012.13985) (2020)
22. Temraz, M., Keane, M.T.: Solving the class imbalance problem using a counterfactual method for data augmentation. In: Machine Learning with Applications (2022)
23. Teney, D., Abbasnejad, E., van den Hengel, A.: Learning what makes a difference from counterfactual examples and gradient supervision. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12355, pp. 580–599. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58607-2_34
24. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017)
25. Verma, S., Dickerson, J., Hines, K.: Counterfactual explanations for machine learning: a review. arXiv preprint [arXiv:2010.10596](https://arxiv.org/abs/2010.10596) (2020)
26. Wang, Z., Culotta, A.: Robustness to spurious correlations in text classification via automatically generated counterfactuals. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 14024–14031 (2021)
27. Wu, T., Ribeiro, M.T., Heer, J., Weld, D.S.: Polyjuice: generating counterfactuals for explaining, evaluating, and improving models. arXiv preprint [arXiv:2101.00288](https://arxiv.org/abs/2101.00288) (2021)
28. Yang, F., Liu, N., Du, M., Hu, X.: Generative counterfactuals for neural networks via attribute-informed perturbation. *ACM SIGKDD Explor. Newsl.* **23**(1), 59–68 (2021)
29. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.* **18**, 1245–1262 (1989)
30. Zhu, Y., et al.: Txygen: a benchmarking platform for text generation models. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 1097–1100 (2018)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Meme Sentiment Analysis Enhanced with Multimodal Spatial Encoding and Face Embedding

Muzhaffar Hazman¹(✉) , Susan McKeever² , and Josephine Griffith¹

¹ University of Galway, Galway, Ireland

{m.hazman1,josephine.griffith}@universityofgalway.ie

² Technological University Dublin, Dublin, Ireland

susan.mckeever@TUDublin.ie

Abstract. Internet memes are characterised by the interspersing of text amongst visual elements. State-of-the-art multimodal meme classifiers do not account for the relative positions of these elements across the two modalities, despite the latent meaning associated with where text and visual elements are placed. Against two meme sentiment classification datasets, we systematically show performance gains from incorporating the spatial position of visual objects, faces, and text clusters extracted from memes. In addition, we also present facial embedding as an impactful enhancement to image representation in a multimodal meme classifier. Finally, we show that incorporating this spatial information allows our fully automated approaches to outperform their corresponding baselines that rely on additional human validation of OCR-extracted text.

Keywords: Multimodal deep learning · Sentiment analysis · Internet memes

1 Introduction

The sentiment polarity classification task traditionally entailed analysing a piece of natural language text to classify its sentiment as negative, positive, or neutral. Sentiment analysis was initially performed on text. The growth of user-generated multimodal content (e.g., videos, image-caption pairs) has motivated the extension of affective computing techniques to input types beyond text [9]. Multimodal sentiment analysis poses the same questions as its text-only predecessor, but is extended to inputs comprising multiple modalities simultaneously. When faced with multimodal inputs, Poria et al. [9] describe unimodal encoders as crucial building blocks of multimodal systems, each encoder directly contributing to the resultant performance. Furthermore, the fusion of unimodal representations also plays a key role by providing “surplus information” to the classifier [9].

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224.

© The Author(s) 2023

L. Longo and R. O'Reilly (Eds.): AICS 2022, CCIS 1662, pp. 318–331, 2023.

https://doi.org/10.1007/978-3-031-26438-2_25

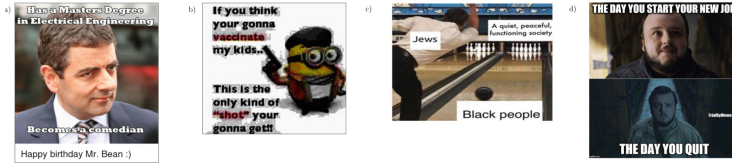


Fig. 1. Sample memes with a) *Positive* sentiment [8] and b) *Negative* sentiment [8], c) hateful spatial analogies [13], and d) spatial segments [14].

Along with the advent of other multimodal formats of user-generated content, Internet memes (or simply “memes”) have proliferated. Memes are commonly found in various online communities to communicate ideas, incite humour, and express emotions. Automated analysis of memes allows for: including memes in automated opinion mining processes [9], taking action against meme-based hate speech [6, 13], identifying disinformation campaigns [1], and investigating social and political cultures [5]. This work contributes to the underlying problem of **sentiment polarity classification of a meme**: “Given a meme in a visual format, comprising an image I with embedded text T , classify the meme as having the overall sentiment of either *Negative* (e.g., Fig. 1b), *Positive* (e.g., Fig. 1a), or *Neutral*”.

Mememes are challenging input in automated affective classification problems, as they typically exhibit very brief texts, references to popular culture, subtle intermodal semantic relations, and dependence on background context [11, 13, 13, 17]. Thus, solutions must consider the semantics of each, the textual and visual modalities, and their combinations [6]. The breadth of this challenge spans various affective goals, including sentiment polarity [8, 14], offensiveness [6, 8, 14], sarcasm [8, 14], and motivational intent [8, 14].

Recent work has shown that incorporating additional relevant information improves the performance of meme affective classifiers [11], amongst which is positional information of words within text and visual objects within an image [13, 17]. Unlike many other forms of multimodal content, the text within a meme is interspersed into its image, often either superimposed on the image or comprising a segment of the meme image, creating a shared visual medium. Meme authors intentionally position a grouping of words (“text clusters”) to convey meaning, such as implying hateful analogies [13] (e.g., Fig. 1c); text clusters can be paired with image segments, with each pair signifying a different sentiment (e.g., Fig. 1d). Current approaches that use positional information in meme sentiment classification opt to omit intermodal positional relations, i.e. they consider the position of a word amongst text but not its position in relation to the meme image or vice versa.

This work proposes injecting the spatial information of features from both modalities of a meme into a deep learning multimodal classifier to improve sentiment classification performance. Crucially, we account for the interspersing of

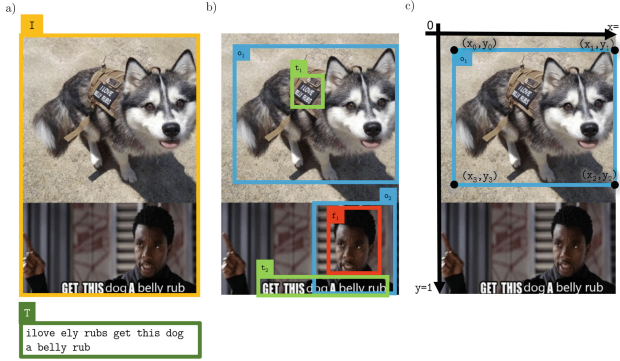


Fig. 2. Sample meme [8] a) showing the image and text modalities, I and T , as given in the dataset; b) bounding boxes generated for local features: text clusters (t_1 and t_2), objects (o_1 detected as “Dog” and o_2 as “Person”), and faces (f_1); and c) the coordinate system used to generate the spatial encoding for each bounding box (e.g. the vertices of o_1 , p_{o_1}).

visual objects and text clusters by representing the spatial position of each on a shared coordinate system (“spatial encoding”). We append the spatial encoding of visual objects (e.g. o_1, o_2 in Fig. 2b), faces (e.g. f_1 in Fig. 2b), and text clusters (e.g. t_1, t_2 in Fig. 2b) to their local representations prior to multimodal fusion and classification. The performance implication of spatial encodings and local representations are systematically evaluated on two benchmark datasets using the seven models described in Sect. 3.2. To the best of our knowledge, this work is the first to use shared coordinate spatial encoding and deep representation of faces to tackle the sentiment classification of memes.

2 Related Works

2.1 Meme Affective Classifiers

Memes are distinct from other multimodal user-generated content types in several key ways. First, the text and image of a meme share a common visual medium, unlike the more common image-caption pairs. Text in memes is often intentionally located amongst other visual content to create meaning [13]. Second, memes use short text pieces and few foreground visual objects, relying on intermodal relations to convey meaning. Kiela et al. [6] show how harmless images and texts could be combined to create hateful memes. Furthermore, slight changes in either modality can change a hateful meme into a harmless one and vice versa. Therefore, meme classifiers must be able to learn subtle intermodal relationships with very limited input.

Architecturally, the current literature suggests that various affective classification tasks can be applied to memes without requiring entirely distinct

approaches. Most apparently, Bucur et al.’s [3] winning submission of the Memotion 2022 Challenge [8], was trained to simultaneously classify sentiment polarity, offensiveness, sarcasm, humour, and motivational intent. Their findings suggest that meme classification architectures exhibit adaptability across different affective computing tasks. Furthermore, Pramanick et al. [10], who reported the best-performing sentiment classification solution to the Memotion 1.0 dataset [14], showed that the same architecture outperforms all, or all but one, competing solution when individually trained on eight affect dimensions.

A typical approach to building a multimodal meme classifier is to generate unimodal representations of each modality before fusing these representations into a multimodal representation of the meme, such as in [3, 10, 11, 13]. Furthermore, the literature presents a wide range of deep learning representations used for each visual and textual modality [6, 8, 14], with no clear evidence that any of the options would consistently outperform all others.

2.2 Positional Encoding

Positional encoding plays a central role in the Transformers architecture [15] and has seen wide adoption in tackling various natural language tasks. It describes the position of tokens, such as a word in a sentence or a region in an image, within the input. However, since most multimodal meme classifiers employ unimodal encoders, the positions of text and visual elements are encoded separately.

To the best of our knowledge, a positional encoding that is shared between the text and image modalities on a common spatial coordinate system (a “spatial encoding”) has not been applied to classifying meme sentiment. None of the architectures reportedly used to learn meme sentiment classification in [14] and [8] did so using a positional information from a coordinate system shared between modalities. Further, we were not able to find a pre-trained multimodal Transformer that readily supports such a shared encoding.

In this task, Pramanick et al. [10] showed performance gains by segmenting the text modality into text clusters but did not explicitly represent the spatial position of each cluster. To classify hateful memes, Zhu [17] employed a patch detector to divide each meme into “image regions”. They then appended each text token with a representation of its surrounding image patch. However, they did not present the performance gains solely attributable to this approach. Further, we posit that such a patch-based definition of position would not be suitable where multiple text clusters are placed within the same image patch (e.g., Fig. 1c) or where a patch consists only of text (e.g. Fig. 1a).

Shang et al. [13] proposed a more general representation of spatial position by appending the spatial encoding of extracted visual objects and text clusters prior to input into an intermodal co-attentive pooling module based on a design from [7]. They attributed their model’s outperforming of other leading hateful meme classifiers to its “awareness” of offensive intermodal analogies: the purposeful superimposing of a text cluster near to a visual object is used to represent an offensive conceptual comparison. While their approach is predicated solely on offensive spatial analogies, we posit that this approach could capture

a broader category of intermodal spatial relationships, including those captured by Pramanick et al.’s [10] and Zhu’s [17] approaches.

2.3 Visual Feature Representations

While the image modality is commonly represented by passing the entire meme image through an image encoder [8], enhancing this representation with that of extracted visual objects has proven beneficial in classifying hateful memes [11, 13, 17]. One such approach is to input the meme image into Google Cloud Vision API’s Web Entity Detection to create a corresponding description or set of attributes in text format [11, 17]. Zhu [17] also demonstrated further performance improvement with the inclusion of Race and Gender tags for each face using a pre-trained FairFace classifier. Pramanick et al. [11] also showed improved performance by representing cropped images of visual objects and faces with VGG-19. Shang et al. [13] also found that their multimodal classifiers perform best when global and local visual feature representations are available.

The use of faces to convey sentiment is neither new nor unique to memes. Firstly, visual sentiment analysis [16] points to facial expressions as a valuable mid-level feature in classifying the sentiment conveyed by images from social networks. Second, facial expression emojis have been shown to be informative in supporting the sentiment classification of textual social media [2]. In memes, Zhu [17] argues that expecting a global image encoding to sufficiently recognise facial features that are predictive of hatefulness is unreasonable given the size of current meme datasets. Although we agree with Zhu’s argument, we posit that their approach omits other information conveyed by faces that may indicate a meme’s sentiment, such as emotion, expression, and identity.

3 Methodology

In this work, we evaluate the performance of seven novel multimodal classifier models. These models are separately trained on two competition datasets, Memotion 1.0 [14] and Memotion 2.0, [8], to classify the sentiment polarity of memes. We first designed and evaluated a multimodal deep learning model to establish baseline performance. This model is then repeatedly augmented to answer our research questions. Augmentations include incorporating spatial information of faces, visual objects, and text clusters and are described for each model in Table 3. Evaluation is conducted based on the differences in macro-averaged and weighted-averaged F1 scores – metrics prescribed by the authors of the datasets [8, 14] – between pairs of models that respectively include and exclude each augmentation. This section presents details of the datasets and models used.

3.1 Dataset and Feature Extraction

This work utilises datasets presented in the SemEval 2019 Memotion 1.0 [14] (“**Memo1**”) and AAAI 2022 Memotion 2.0 [8] challenges (“**Memo2**”). Both

Table 1. Samples per dataset.

Dataset	Memo1			Memo2		
	Train	Val	Test	Train	Val	Test
Original						
Positive	4,156	–	1,099	1517	325	78
Neutral	2,204	–	584	4510	975	971
Negative	631	–	172	973	200	451
Total	6,991	–	1,855	7,000	1,500	1,500
Filtered & Filtered-OCR						
Positive	3,450	609	1,067	1,453	192	76
Neutral	1,837	324	572	4,363	951	939
Negative	518	92	169	941	317	442
Total	5,805	1,025	1,808	6,757	1,460	1,457

are collections of user-generated memes labelled with one of three exclusive sentiment classes. The authors of the datasets extracted text from each meme with an automated OCR tool and then manually corrected any erroneous text extraction. For our experiments, the samples from Memo1 and Memo2 are kept separate. Without filtering or pre-processing, these samples comprise our **Original** datasets that we use to compare our **Baseline** model to leading solutions.

For each meme in these datasets, we localised, extracted, and represented its text clusters, faces, and visual objects using the tools listed in Fig. 3. The maximum counts of text clusters, visual objects, and faces are set to 18, 10,

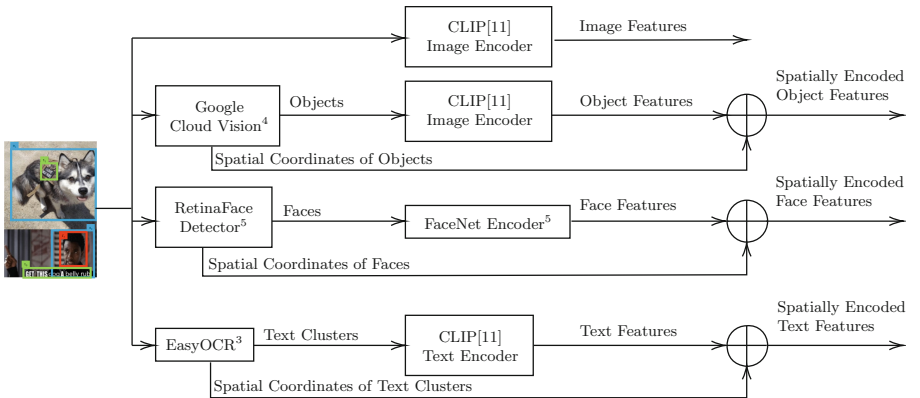


Fig. 3. Localisation and representation process applied to each meme to extract its Image, Object, Face and Text features. ³<https://github.com/JaiedAI/EasyOCR>; paragraph option set to true. ⁴<https://cloud.google.com/vision/docs/object-localizer>. ⁵Using DeepFace wrapper from <https://github.com/serengil/deepface>.

and 5, respectively, with padding used for memes with fewer. Padding for text clusters is defined by passing an empty string into the CLIP text encoder, while that for visual objects is the CLIP encoding of a blank image, and zero-padding is used for faces.

Since this work applies to memes that contain identifiable visual objects and text clusters, we removed meme samples that do not meet these criteria to make up the **Filtered** datasets. This filtering is performed on all subsets of Memo1 and Memo2. As Memo1 did not contain a designated validation set, we defined one by splitting the training set – as reported by the authors of the Memo1 dataset and used in submissions to their competition [14] – with a random 85:15 sampling, weighted by the sentiment class, to maintain the target distribution. We maintained the train-validation-test splits defined for Memo2 [8]. Meme samples with identifiable visual objects but no detected faces are given face feature representation made up entirely of padding.

Finally, the **Filtered-OCR** datasets replace the text of each meme in **Filtered** with that returned in our feature extraction OCR step. Unlike in [8, 10, 14], we excluded any additional human validation during the OCR extraction process. All models are trained, validated, and tested on the resultant **Filtered-OCR** datasets. The counts of memes in each dataset and sentiment labels are shown in Table 1.

3.2 Models

This section describes the architectural characteristics of our models as listed in Table 2 and illustrated in Fig. 4. Each was built using PyTorch and trained with a triangular cyclical learning rate schedule ranging between $1e-4$ and $1e-3$ with a step size of 52 mini-batches of 512 samples. During training, validation performance was monitored for overfitting or until each model was trained for 100 epochs. Training is carried out using AdamW optimiser with weight decay of $5e-1$, betas of 0.1 and 0.25 to minimise negative log-likelihood loss with class weights inversely proportional to its sample count in the training dataset. All non-pretrained weights are initialised with a zero-mean Gaussian distribution with standard deviation 0.02, while pretrained weights are not fine-tuned. The same hyperparameter settings are maintained across all models as they are separately trained on the datasets.

Leading meme sentiment classifiers use a variety of architectures with little indication of which is most optimal. For our **Baseline** model, we drew inspiration from the typical overall approach used in leading solutions to the Memotion 2.0 Challenge [8]: each modality is represented using a pretrained encoder. Then, these representations are fused, often with a multimodal attention mechanism, and finally passed to a fully connected layer.

To encode the meme image and text (see I and T in Fig. 2a) in our **Baseline** model, we opted to use the pretrained image and text encoders of CLIP [12], respectively, which has shown comparable performance to other multimodal approaches [11]. In addition, CLIP image encodings have been shown to outperform various other image encoders in the zero-shot classification of hateful

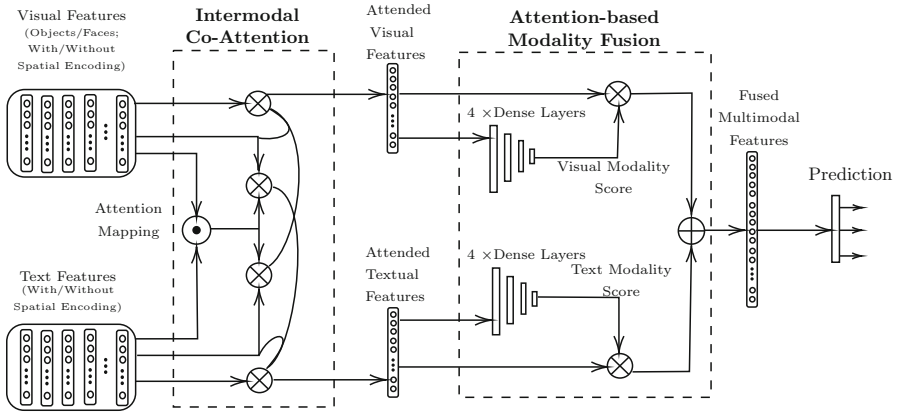


Fig. 4. Architecture of our **Obj-NoSpatial**, **Obj-Spatial**, **Face-NoSpatial**, and **Face-Spatial** models. The Image features used in **Img-Obj-Spatial** and **Img-Face-Spatial** models bypasses the Intermodal Co-Attention module and requires the Attention-Based Modality Fusion module to be expanded with another set of dense layers. This work’s **Baseline** model does not include the Intermodal Co-Attention module. Sources: Intermodal Co-Attention [7,13]; Attention-based Modality Fusion [4,10]

memes [12] and are used by the winning solution of the Memo2 challenge [3]. We chose the ViT-B/16 variant of CLIP while Pramanick et al. [11] and Bucur et al. [3] did not report their chosen variant.

Since attentive fusion has been shown to perform well on several meme problems [10], we included one in our models. Our **Baseline** model fuses the CLIP representations of the meme image and text using Gu’s [4] attentive modality fusion mechanism, as used in [11]. We defined the sizes of the four dense layers as 256, 64, 8, and 1, which produces an attention score for each modality. The attention-weighted representation of each modality is concatenated and passed into a GeLU-activated dense layer followed by a log-softmax activation to output predicted logits of each sentiment class.

This model is trained on the **Original** dataset to allow performance comparisons with previously published works. We then evaluated this model on the **Filtered** and **Filtered-OCR** datasets. In the latter, the content of all text clusters t_n is concatenated and entered into the text encoder. The difference in the performance of this model on these two datasets allows us to measure the performance impact resulting from our OCR-based text extraction output relative to the human-curated approach used by the authors of the datasets [8,14].

The **Obj-NoSpatial** and **Face-NoSpatial** models remove the meme image and text, I and T per **Baseline**. As inputs, the former takes CLIP-encoded visual objects, o_1, o_2, \dots, o_j , and text clusters extracted from a meme, t_1, t_2, \dots, t_i . Instead of objects, the **Face-NoSpatial** model takes the FaceNet representation of faces, f_1, f_2, \dots, f_k . Then, the j visual objects or k face representations are

passed through co-attentive weighted pooling against i text clusters as used in [13] but without spatial encodings. This step allows the models to learn attention maps between each object/face and each text cluster; producing a one-dimensional vector representing each modality. This representation replaces that of the image modality as input into the attentive fusion mechanism described for the **Baseline** model.

The **Obj-Spatial** and **Face-Spatial** models introduce the spatial encodings of each text cluster, p_{t_i} , as well as for visual objects, p_{o_j} , and faces, p_{f_k} , respectively. We augment the co-attentive pooling module in **Obj-NoSpatial** and **Face-NoSpatial** into the co-attentive analogy alignment module proposed in [13]. This is performed by appending each object’s and cluster/face’s representation vector with its spatial encoding. The padding for spatial encodings is defined as zeros for all coordinates.

The **Img-Obj-Spatial** and **Img-Face-Spatial** models each combine the CLIP representation of the meme image, I , into **Obj-Spatial** and **Face-Spatial**, respectively. Since these models make use of three representations per meme – image, text clusters and objects/faces – we extend Gu’s [4] fusion mechanism to accommodate three inputs by introducing a third set of dense layers.

Table 2. Goals of each experimental model.

Model	Dataset	Goal
Baseline	Original	Benchmarks our chosen modality encodings and fusion mechanism against leading solutions
	Filtered	Establishes baseline performance on samples with detectable text clusters and visual objects
	Filtered-OCR	Measures the impact of replacing human-curated text replaced with text clusters returned by automated OCR. Also, establishes a baseline for our fully automated approaches
Obj-NoSpatial	Filtered-OCR	Measures the performance of representing the image modality using only CLIP-encoded localised visual objects without spatial encodings
Obj-Spatial	Filtered-OCR	Measure the performance impact of including spatial encodings of objects and text clusters
Img-Obj-Spatial	Filtered-OCR	Maximises available visual information by augmenting image input with objects and text clusters and respective spatial encodings
Face-NoSpatial	Filtered-OCR	Measures the performance of representing the image modality using only embeddings of localised faces without spatial encodings
Face-Spatial	Filtered-OCR	Measure the performance impact of including spatial encodings of faces and text clusters
Img-Face-Spatial	Filtered-OCR	Augments image input with faces and text clusters and respective spatial encodings

Table 3. Performance of our **Baseline** model against leading solutions on the Memo1 dataset. Sources: [10, 14].

Solution	Macro-F1
MHA-Meme ^a [10]	0.3762
Vkeswani IITK	0.3547
Our Baseline	0.3546
Guoym	0.3520
Aihaiara	0.3502
Sourya Diptadas	0.3476
Irina Bejan	0.3469

^a Not a competition submission; results based on subset of the original dataset

Table 4. Performance of our **Baseline** model against leading solutions on the Memo2 dataset. Source: [8].

Solution	Weighted-F1
BLUE	0.5318
BROWALLIA	0.5255
Yeti	0.5088
Little Flower	0.5081
Greeny	0.5037
Our Baseline	0.5035
Amazon PARS	0.5025

4 Results

Evaluating the **Baseline** model on the **Original** datasets places it within the top six highest performing solutions on each respective dataset; see Tables 3 and 4.

The performance of the **Baseline** model on the **Original**, **Filtered** and **Filtered-OCR** datasets are shown in Table 5. The lower performance of the model on the **Filtered** dataset than on the **Original** dataset likely stems from the removal of samples that contain only text on an object-less background. Classifying such samples is similar to discerning the sentiment of unimodal text inputs and is beyond the scope of this work. We attribute the performance decrease of the **Baseline** model on the **Filtered-OCR** vs. **Filtered** datasets to the lower quality of the text extracted with our automated OCR process relative to human-curated text. Despite this, our spatially aware models are able to overcome this performance penalty. The model that performs best on each dataset – as seen in Table 6 – constitutes **fully automated approaches** that outperform their respective **Baseline** models trained on the human-curated text

Table 5. Weighted F1 (F1-W) and Macro F1 (F1-M) for the **Baseline** model on all datasets.

Dataset	Memo1		Memo2	
	F1-W	F1-M	F1-W	F1-M
Original	0.481	0.355	0.504	0.325
Filtered	0.475	0.327	0.503	0.314
Filtered-OCR	0.462	0.326	0.439	0.283

Table 6. Weighted F1 (F1-W) and Macro F1 (F1-M) for all models on the Memo1 and Memo2 **Filtered-OCR** datasets. **Rel.** indicates relative performance to model stated in the **Comparison** column on each given dataset.

Model	Comparison	Memo1			Memo2		
		F1-W	F1-M	Rel.	F1-W	F1-M	Rel.
Baseline	-	0.462	0.326	-	0.439	0.283	-
Obj-NoSpatial	vs. Baseline	0.452	0.307	↓	0.427	0.271	↓
Obj-Spatial	vs. Obj-NoSpatial	0.481	0.317	↑	0.482	0.305	↑
Img-Obj-Spatial	vs. Obj-Spatial	0.489	0.336	↑	0.499	0.300	↑↓
Face-NoSpatial	vs. Baseline	0.476	0.340	↑	0.471	0.298	↑
	vs. Obj-NoSpatial			↑			↑
Face-Spatial	vs. Face-NoSpatial	0.485	0.341	↑	0.496	0.310	↑
	vs. Obj-Spatial			↑			↑
Img-Face-Spatial	vs. Face-Spatial	0.473	0.332	↓	0.509	0.314	↑
	vs. Img-Obj-Spatial			↓			↑

from the **Filtered** datasets. By removing the need for manual intervention, fully automated models improve the feasibility of conducting sentiment classification of memes at scale, and reduce the effort necessary for creating future meme datasets.

The results show that spatial encoding improves performance. **Obj-Spatial** and **Face-Spatial** each outperforms **Obj-NoSpatial** and **Face-NoSpatial** respectively. These results point to intermodal spatial information being informative for the problem task and not sufficiently represented by the CLIP encodings of the whole meme image. This finding holds significance to applying deep learning solutions on memes in particular, as the text modality is incorporated and interspersed within the image. Although the importance of token positions in leading solution architectures has been well established, the lack of a shared visual medium for image and text modalities in many other vision-language tasks has resulted in leading multimodal architectures with separate positional representations for each modality. Based on our results, we argue that spatial encodings should also be considered for other vision-language tasks where visual objects and text share a common visual medium.

The performance benefit of representing the image modality with localised visual feature representations depends on whether the features are defined as objects or faces. CLIP-encoded object representation performs worse than **Baseline**. This results from a reduction in the visual information available to the image encoder. However, **Face-NoSpatial**, which uses FaceNet embeddings to represent faces, outperforms both **Obj-NoSpatial** and **Baseline** while also suffering from the same, if not greater, reduction in available visual information. Furthermore, **Obj-Spatial** showed mixed results against **Baseline**, while **Face-Spatial** outperforms **Baseline** in both datasets. Notably, faces are not entirely excluded from models based on visual objects, as many meme sam-

ples had “Person” as a detected object. Thus, we believe that the performance difference between the two approaches arises from the more fine-grained facial embedding provided by FaceNet and the inherent exclusion of non-face visual objects that emphasises the contribution of faces to the sentiment of a meme.

We found that augmenting the meme image with local representations of either objects or faces and their spatial encodings consistently outperforms models that rely on the image alone. However, choosing between CLIP-encoded objects versus FaceNet-encoded faces as augmentations to the meme image proved inconsistent and dependent on the dataset. Although **Img-Obj-Spatial** and **Img-Face-Spatial** perform the best in the Memo1 and Memo2 datasets, respectively, their performance relative to **Obj-Spatial** and **Face-Spatial** appears to depend on the dataset. Drops in performance here may stem from redundant intermodal information (e.g. between global image and objects-based representations). Unlike in [10], we did not employ any form of learned cross-modal filtering.

5 Conclusions

In this work, we addressed spatial encoding and facial embedding in classifying sentiment polarity of internet memes. We developed seven novel architectures, and evaluated each on two challenge datasets. For both datasets, our proposed baseline multimodal classifier ranked within the top six of leading state-of-the-art solutions on both datasets. While we found that representing the image modality with visual objects alone does not consistently offer performance benefits, a face-based representation does. Furthermore, the incorporation of spatial information of these visual features grants performance improvements over both image-only and faces-/objects-only approaches. For each of the Memotion datasets, our top performing solution comprises augmenting the image modality with spatially encoded visual features and text clusters. We propose these solutions as fully automated competitive alternatives to current state-of-the-art solutions that rely on manual validation of OCR-based text extraction.

References

1. Al-Rawi, A.: Political memes and fake news discourses on instagram. *Media Commun.* **9**(1), 276–290 (2021). <https://doi.org/10.17645/mac.v9i1.3533>
2. Ayvaz, S., Shiha, M.: The effects of emoji in sentiment analysis. *Int. J. Comput. Electr. Eng.* **9**, 360–369 (2017). <https://doi.org/10.17706/ijcee.2017.9.1.360-369>
3. Bucur, A.M., Cosma, A., Iordache, I.: BLUE at memotion 2.0 2022: you have my image, my text and my transformer. In: *De-Factify @ AAAI 2022. First Workshop on Multimodal Fact-Checking and Hate Speech Detection*. *CEUR Workshop Proceedings*, AAAI (2022)
4. Gu, Y., Yang, K., Fu, S., Chen, S., Li, X., Marsic, I.: Hybrid attention based multimodal network for spoken language classification. In: *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA, pp. 2379–2390. Association for Computational Linguistics (2018)

5. Joshi, A., Buntain, C.: Exploiting the right: inferring ideological alignment in online influence campaigns using shared images. In: Workshop Proceedings of the 16th International AAAI Conference on Web and Social Media (2022). <https://doi.org/10.36190/2022.45>
6. Kiela, D., Firooz, H., Mohan, A., Goswami, V., Singh, A., Fitzpatrick, C.A., et al.: The hateful memes challenge: competition report. In: Proceedings of the NeurIPS 2020 Competition and Demonstration Track. Proceedings of Machine Learning Research, vol. 133, pp. 344–360. PMLR (2021)
7. Lu, J., Yang, J., Batra, D., Parikh, D.: Hierarchical question-image co-attention for visual question answering. In: Advances in Neural Information Processing Systems, vol. 29. Curran Associates, Inc. (2016)
8. Patwa, P., Ramamoorthy, S., Gunti, N., Mishra, S., Suryavardan, S., Reganti, A., et al.: Findings of memotion 2: sentiment and emotion analysis of memes. In: DeFactify @ AAAI 2022. First Workshop on Multimodal Fact-Checking and Hate Speech Detection. CEUR Workshop Proceedings, AAAI (2022)
9. Poria, S., Cambria, E., Bajpai, R., Hussain, A.: A review of affective computing: from unimodal analysis to multimodal fusion. *Inf. Fusion* **37**, 98–125 (2017). <https://doi.org/10.1016/j.inffus.2017.02.003>
10. Pramanick, S., Akhtar, M.S., Chakraborty, T.: Exercise? I thought you said ‘extra fries’: leveraging sentence demarcations and multi-hop attention for meme affect analysis. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 15, no. 1, pp. 513–524 (2021)
11. Pramanick, S., Sharma, S., Dimitrov, D., Akhtar, M.S., Nakov, P., Chakraborty, T.: MOMENTA: a multimodal framework for detecting harmful memes and their targets. In: Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, pp. 4439–4455. Association for Computational Linguistics (2021). <https://doi.org/10.18653/v1/2021.findings-emnlp.379>
12. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
13. Shang, L., Zhang, Y., Zha, Y., Chen, Y., Youn, C., Wang, D.: AOMD: an analogy-aware approach to offensive meme detection on social media. *Inf. Process. Manag.* **58**(5), 102664 (2021). <https://doi.org/10.1016/j.ipm.2021.102664>
14. Sharma, C., Bhageria, D., Scott, W., PYKL, S., Das, A., Chakraborty, T., et al.: SemEval-2020 task 8: memotion analysis- the visuo-lingual metaphor! In: Proceedings of the Fourteenth Workshop on Semantic Evaluation, Barcelona, pp. 759–773. International Committee for Computational Linguistics (2020). <https://doi.org/10.18653/v1/2020.semeval-1.99>
15. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, Nips 2017, pp. 6000–6010. Curran Associates Inc., Red Hook (2017)
16. Yuan, J., Mcdonough, S., You, Q., Luo, J.: Stribute: image sentiment analysis from a mid-level perspective. In: Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining. Wisdom 2013. Association for Computing Machinery, New York (2013). <https://doi.org/10.1145/2502069.2502079>
17. Zhu, R.: Enhance Multimodal Transformer With External Label And In-Domain Pretrain: Hateful Meme Challenge Winning Solution (2020). <https://doi.org/10.48550/arxiv.2012.08290>



Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Intelligent Image Compression Using Traffic Scene Analysis

David Bowden¹(✉)  and Diarmuid Grimes² 

¹ Dell Technologies, Ovens, Ireland
david.bowden@dell.com

² Munster Technological University, Cork, Ireland

Abstract. The quantity of images generated at the edge of the Cloud is growing year-on-year, which puts an increasing strain on existing telecommunications infrastructure. There is also an associated increased cost for transmission bandwidth and storage of video images in the Cloud. In our modern society we tend to accumulate data, and are reluctant to throw it away, without asking “what is the value of this data?” and “do we need it?”. One of the major sources of video streams are the increasing number of traffic cameras, used to maintain the efficient flow of vehicles on our roads. In this work we focus on images taken from road traffic cameras, and show how their transmission bandwidth and storage requirements can be reduced. By analysing video feeds on a simulated edge device, we have shown that it is possible to extract objects of interest from the image, and discard or dramatically reduce irrelevant information in the content. Our technique also generates associated meta-data, in the form of JSON-LD, which annotates the original image and maintains its semantic fidelity and provenance after compression. Our technique is compatible with conventional compression techniques, and thus the potential bandwidth savings would be incremental. We present the potential savings that can be made in the transmission and storage of unstructured data, as well as some of the challenges still to be overcome.

1 Introduction

In a white paper CISCO forecasted that Cloud traffic from 2015 to 2020 would almost quadruple from 3.9 Zettabytes to 14.1 Zettabytes [15], and Forbes claimed in 2018 that “90% of the worlds data has been created in the last 2 years... accelerating with the growth of the Internet of Things” [11]. One of the primary sources of unstructured data are video cameras [23]. Originally video cameras were implemented as Closed-Circuit TV systems, and the transmission of data was limited. With the emergence of IoT and the reduction in video camera costs, the number and distribution of cameras has mushroomed in areas such as traffic management [9].

Typically, video camera feeds are transmitted back to control rooms as sequences of discrete JPEG images, where they are displayed on large video walls of monitors. As the number of traffic cameras has increased, this strategy

has proved impractical. In the last two decades, video image processing has been an active area of research, to try and automate many of the mundane activities currently performed by humans [12].

To understand a scene, we must first extract the objects of interest within it. Humans perform this task easily with little prior knowledge – for computers the task is more challenging. They need to identify foreground objects as Regions of Interest (RoI), and associate a classification to the object: car, bus, motorcycle, etc. Additional attributes and sub-objects can also be detected and associated to a parent object, such as the model of a car [13] or its licence plate. Xu et al. [29] proposed a technique for annotating objects in images using RDF graphs, which describes the road traffic scene in the form of semantic metadata. However, in our work we focus on encoding metadata for transmission using JSON-LD.

In this paper we introduce our approach for implementing the intelligent compression of images, and empirically demonstrate the benefits of such an approach on a number of well-known benchmark datasets. We utilise the concept of dynamically reducing the resolution of traffic images, based on an analysis of their content and relative value to downstream systems. The concept is to reduce the resolution of the irrelevant parts of the image, whilst maintaining the vehicles at a higher resolution, thus reducing the total transmission bandwidth.

Additionally, we consider the value of images based on their content, and whether they could be, wholly or partially, replaced by semantic metadata. We demonstrate how the technique can reduce the transmission and storage requirements of video images, from the road traffic domain. Our technique can be incorporated into conventional compression techniques to further reduce bandwidth requirements, whilst maintaining semantic fidelity to the original image.

The remainder of this paper is structured as follows. Background on semantic representations and JPEG compression is presented in Sect. 2. The problem and proposed solution are discussed in Sect. 3. The description of the experimental setup is given in Sect. 4. The experimental results are discussed in Sect. 5. Finally, we conclude our work in Sect. 6.

2 Background

The value of road traffic video images can be greatly enhanced by adding metadata, such as traffic counts and conditions, at the point of collection, which adds context to the image content. Whilst traffic management systems provide this functionality, and there is ongoing research in this area [1, 16], they rely heavily on centralised processing and manual operators. As the scale of video collection and processing increases, it places an increasing burden on centralised systems. By moving initial analysis to the cameras and triaging the video images at the edge, the burden on central traffic management systems can be greatly reduced, as well as reducing video transmission and storage costs.

To achieve some level of understanding, machine vision must cross the “Semantic Gap” [24] into some form of knowledge representation; where objects in the scene are classified and related to each other in a semantic way, in our case

through a knowledge graph. Representing metadata as a knowledge graph is a powerful technique, as it is possible for computers to navigate the graphs and draw inferences from them. Therefore, a traffic jam can be conceptually inferred from the presence of many slow-moving cars over a sequence of images [10].

Recently, a number of Internet search companies have proposed the adoption of an extended form of JSON to annotate web pages, called JSON Linked Data (JSON-LD) [19], which can be directly converted into RDF semantic graphs, and vice versa. JSON-LD is starting to gain interest in the IT community, because, similar to JSON, it is both human and machine readable, and is supported over the HTTP protocol [7]. Maarala et al. considered several RDF formats for transmitting IoT data [10]. Whilst they preferred Entity Notation [21], JSON-LD had comparable performance, and its adoption by Internet companies makes it a better choice in our opinion for this case.

Gonzalez and Woods [5] outlined several areas where lossy JPEG compression [25] savings can be made, one of which being the removal of irrelevant information. Irrelevant information is information in the image that the human eye can't see, or the mind ignores; it is the latter principal that we exploit in our method. If the image is to be used for road traffic scene analysis, then other extraneous parts of the image are of secondary importance and may only provide general context information to interpret the scene. The intelligent image compression process focuses the user on the relevant information in the image, in this case the vehicles, whilst smoothing out the irrelevant information.

A similar approach to our work was proposed by Wu et al. [28]. Their technique identified vehicles in a sequence of frames, reduced the resolution of the background, before then adding the higher resolution vehicle images back into the video stream. Standard video encoding, such as MPEG-4, then compresses the stream. Our work extends some of these concepts but compresses the vehicle and background images separately using JPEG compression, before reconstructing the image at the remote server by using semantic relationships in the accompanying JSON-LD metadata.

3 Problem Definition

In this work we investigate methods for compressing images from road traffic cameras that retain a high level of image fidelity for the road traffic domain, whilst reducing the quality of unwanted/irrelevant elements from the original image. Silverstein and Farrell [20] demonstrated that “at most [there is] a weak relationship between fidelity and quality”. They defined image fidelity as that “*inferred by the ability to discriminate between two images*”, and image quality as that “*inferred by the preference for one image over another*”. By this definition, **fidelity** is the human ability to discriminate between images, i.e. the ability to identify information in one image that is not in the other.

Most contemporary image compression research focuses on maintaining image **quality** whilst using fewer Bits Per Pixel (e.g. [18, 22]); thus, they rely on some form of Image Quality Assessment (IQA) [6] when evaluating their results,

but this depends on comparative measures of image quality. However, Wang et al. showed that images with different types, but similar levels of distortion, can have very different “perceptual quality”. From Fig. 2 of their paper “Image Quality Assessment: From Error Visibility to Structural Similarity” [26], the five distorted images have the same Mean Squared Error (MSE), but different levels of information, e.g. the boat’s name can clearly be read in images (b), (c) and (f), but not in (d) and (e).

Our work demonstrates that by extracting objects in the image that are of value to the domain of interest (road traffic management), and highly compressing the background image, we can selectively reduce the average quality of the image whilst maintaining its domain specific fidelity. Therefore, as our method relies on maintaining information at the expense of quality, many conventional image quality metrics are not applicable. Instead, we propose a size equalization approach, where the same image is aggressively compressed and a side-by-side comparison is then made to assess the domain level information in each image, performed using a Machine Learning (ML) object detector to count the number of domain objects that are distinguishable in each image at the server.

If the standard compressed image size is S_c , and the intelligent compression image size is S'_c . Then for S'_c we define: S''_c is the size of a low-quality version of the background image, S_m is the size of the associated metadata¹, O_i is the size of the i^{th} extracted object sub-image, and n is the number of objects extracted from the image.

$$S'_c = S''_c + S_m + \sum_{i=0}^n O_i \quad (1)$$

Assuming an aggressive level of compression, if S_c is the size of a low-quality image using standard compression techniques, and we make:

$$S_c = S'_c \quad (2)$$

Then the relationship between information in the image using a standard compression technique, I_c , and the information in the image using our proposed compression technique, I'_c , is:

$$I_c < I'_c \quad (3)$$

And in the special case where $n = 0$:

$$S'_c = S_m \quad (4)$$

$$I_c << I'_c \quad (5)$$

Therefore, the two objectives of our research are to examine reductions in image transmission bandwidth by reducing/removing irrelevant content, com-

¹ The metadata is the information that spatially and contextually links the extracted object images to the original image.

pared to existing bandwidth reduction techniques; and to examine the benefits of augmenting/replacing video images with high-level semantic metadata. A quantitative experimental approach was taken to evaluate the concept, a prototype was developed to compare the fidelity of images using both standard compression and the proposed intelligent compression techniques, and a series of experiments were performed to compare the two.

3.1 Experimental Approach

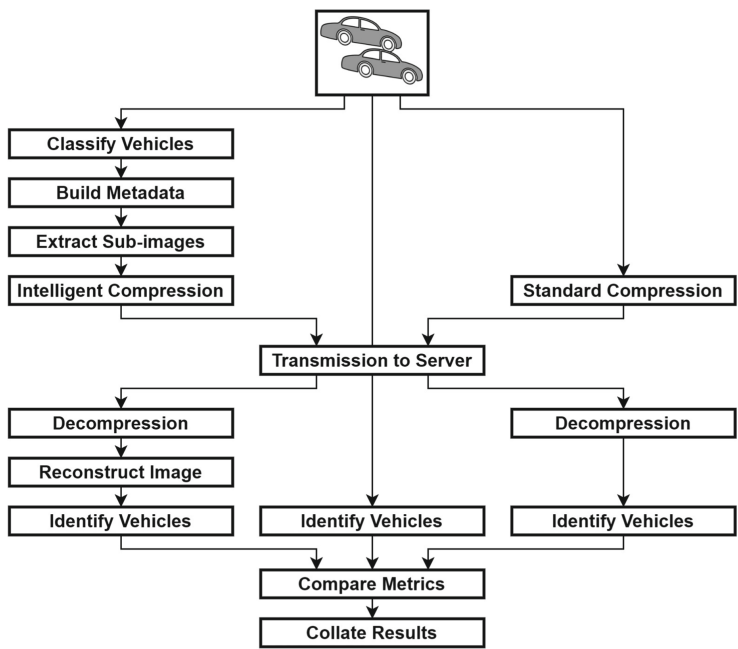


Fig. 1. Experimental approach workflow

The experimental approach is illustrated in the workflow in Fig. 1. On the right is the standard compression approach, on the left the intelligent compression approach, and in the middle the definition of the ground truth. The prototype is fed images from road traffic scenes, with varying vehicle densities, weather and lighting conditions. We simulate a video stream through the use of a sequence of individual road traffic images from publicly available datasets.

The top part of the pipeline in Fig. 1 comprises the steps to compress the image, which would be performed close to the camera at the edge; and the bottom part the steps to decompress and compare the images, which are performed at the server. The quantity being measured is the loss of information, or, conversely, the

fidelity to the original image, caused by the compression technique under high bandwidth constraints. The road traffic images are compressed to approximately 10% of their original size.²

The standard compression workflow is relatively straightforward. The road traffic image at the top is compressed using standard JPEG compression, but the quality of the image is reduced until the resulting file is approximately 10% of its original size, around 18KB. It is then transmitted to the server over HTTP, where it is decompressed. To evaluate the loss of information an ML algorithm is used to identify vehicles still recognisable in the decompressed image.

The intelligent compression workflow is a little more complex. The road traffic image at the top is analysed by an ML algorithm to identify the objects of interest in the scene, the vehicles. The results of the object detection process are used to define the scene in the form of a knowledge graph, the metadata. The metadata describes the scene and where the vehicles are within it. The next step creates a sub-image containing just a vehicle; it does this for every vehicle defined in the metadata. Lastly the images are compressed, using standard JPEG compression, so that their aggregate size is approximately 10% of its original size, around 18KB. The sub-images are maintained at a higher resolution than the original background image, by a ratio of at least 20:1.

The metadata, background image and sub-images are then transmitted to the server over HTTP, where they are decompressed. The image is then reconstructed by using the information in the metadata to overlay the higher quality sub-images, of the vehicles, onto the background, to form a composite image. Again, to evaluate the loss of information an ML algorithm is used to identify vehicles still recognisable in the decompressed image. The last two steps compare the number of vehicles identified, after compression, to the ground truth from the original image. The results are then collated and compared.

4 Evaluation

4.1 Experimental Method

The experiments evaluate our hypothesis that, under severe bandwidth constraints, intelligent image compression will retain more information than standard JPEG image compression. The bandwidth was constrained to approximately 3.7 Mbps, which equates to an average image size of 18 KB. To meet this bandwidth constraint, the original image was progressively reduced in quality until the standard compression yielded an image size of approximately 18 KB. The process was repeated for the intelligent compression technique, except that in this case, all the resultant files, background image, vehicle sub-images and

² During experimentation, it was determined that 10% of the original file size, approximately 18 KB, was the practical minimum limit as the JPEG image overheads become a significant factor below this point, and the images start to become excessively pixelated.

metadata had to add up to approximately 18KB. This meant that the bandwidth constraint applied equally to both forms of compression (see Eq. 2). After the intelligently compressed image had been reconstructed at the server, the amount of information in both images was compared to the original image, and the results were compared to calculate any loss in information, i.e. the fidelity of the image to the original.

4.2 Experiment Dataset

The UA-DETRAC Benchmark dataset [27] was selected as the most suitable, and closely resembles the image feeds that would be obtained from actual cameras. A small subset of the other datasets were included in the experiments for comparison, such as KITTI [4] and 2020 AI City Challenge [14]. The UA-DETRAC dataset has approximately 10 h of video footage collected from two cities in China, Beijing and Tianjin, showing 24 different road traffic scenes. The videos are captured at a rate of 25 frames per second, a resolution of 960×540 pixels, and have three 8-bit colour channels. The video sequences were taken in a number of different conditions: sunny, cloudy, rainy, and at night; and from several camera perspectives.

4.3 Prototype Design

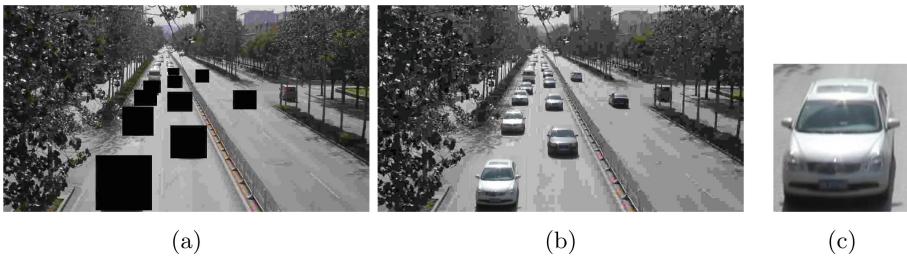


Fig. 2. Redacted vs reconstructed image, sample vehicle sub-image

A prototype of the intelligent image compression technique was developed to validate the concept and measure the potential reduction in image fidelity at highly restricted transmission bandwidths. It consists of a pipeline of components that implements the workflow illustrated in Fig. 1, using Python 3 and the OpenCV image manipulation library. The connection between the edge and the server is implemented as a HTTP RESTapi using JSON-LD [7] as the metadata message format.

The test-rig uses the YOLO [17] object detector to identify the vehicles in the scene. The detector has been pre-trained on the COCO [8] image dataset.

The COCO dataset consists of 80 classifications of objects, but only 4 of them relate to the road traffic domain.

Figure 2 presents different aspects of the process for a sample image. Figure 2a shows the original image with the vehicles removed, leaving the background image. The background image is reduced in quality, yielding a more compressed file. However, the vehicles, such as the sample show in Fig. 2c, are sent as separate sub-images.

Lastly, Fig. 2b shows the image reconstructed at the server. As can be seen from the image the secondary objects in the scene, road, tree, buildings, etc. have been reduced in quality and become pixelated, but the vehicles remain at their original quality. To improve the visual appearance to the user, a Gaussian filter is applied to smooth the image to reduce the visual impact of pixelation. This has the effect of focusing the user's attention on the vehicles, with the more blurred background providing context for the scene.

With regard to the sub-images, ideally these will be sent at the original resolution, to provide the maximum amount of information at the server. However, if there are a large number of sub-images or specific sub-images of a large size, their aggregate size may be greater than the constrained bandwidth. In these cases, the sub-images are also reduced, but a minimum ratio of 20:1 is maintained between the quality of the sub-images and the background image. The spatial relationship between the background image and the sub-images is maintained in the JSON-LD metadata, which accompanies the images. For the purposes of road traffic analysis, certain parts of the image are ignored, such as the mini-bus on the far right, which is on a side road. These are specified in the UA-DETRAC dataset as “*Ignore Regions*”.

5 Experimental Results

In order to compare the compression methods with respect to fidelity to the original, we assess the ability of the YOLO object detector to correctly identify the objects in each of three variants of an image, the standard compressed, the intelligently compressed, and the original. From this we can compile the true positive (TP), false negative (FN) and false positive (FP) scores for each of the images in a dataset: TP is a vehicle found in the ground truth that was also found in the compressed image; FN is a vehicle found in the ground truth that was not found in the compressed image; and FP is a vehicle found in the compressed image that was not found in the ground truth.

We then use these to generate F1 scores [2] as our metric for comparing the standard and intelligently compressed images to the ground truth for each frame. The score is in the range 0.0–1.0, where 1.0 indicates that there was no loss of information, i.e. all vehicles were correctly identified from the original image. This provides a measure of fidelity to the original images by averaging the F1 score across all frames. For the MVI_39031 dataset we obtained a standard compression score of 0.697, and an intelligent compression score of 0.865.

This indicates that, at very high compression ratios, the intelligent image compression technique retained approximately 24% more information over standard compression.

The experiment was repeated for the 11 other datasets, and the results are presented in Table 1. The “Diff.” column gives the percentage improvement of intelligent compression over standard compression for all experimental datasets. Overall the intelligent compression technique outperformed standard compression in 10 of the 12 experiments, with standard compression being only marginally better in the other 2 experiments.

Table 1. Average F1 scores for standard and intelligent compression

Dataset Name	Frame Count	Vehicles per Frame	Standard Comp.	Intelligent Comp	Diff.	Comments
MVL39051	1120	1.4	0.538	0.812	50.9%	Isometric view
2011-09-26 [4]	179	2.1	0.441	0.655	48.5%	Front view at traffic lights
MVL39031	1470	3.5	0.697	0.865	24.1%	Front view
MVL63544	1160	0.9	0.687	0.799	16.3%	Isometric view in rain
MVL40992	2160	1.2	0.56	0.651	16.3%	Isometric view at night
MVL63525	985	2	0.837	0.907	8.4%	Front view in rain
MVL40211	1950	2.9	0.708	0.751	6.1%	Iso. view with motion blur
S05 C028 [14]	1965	2.4	0.892	0.935	4.8%	Junction with traffic lights
MVL40131	1645	6.3	0.875	0.894	2.2%	Front view in rain
S04 C019 [14]	856	0.6	0.837	0.85	1.6%	Front view sunny day
MVL40892	1790	9	0.494	0.485	-1.8%	Busy junction in rain
MVL40963	1720	5.1	0.838	0.765	-8.7%	Front view at night

6 Conclusions

Our experiments have shown that intelligent compression can outperform standard JPEG compression in the majority of datasets tested, and in a quarter of the cases by over 24%. The technique performed well when the number of images or the size of the images was small; but as the number or sizes increased, the additional overhead of transmitting multiple images became a significant factor, and the fidelity of the composite image, to the original, dropped.

This suggests that there is an inflection point where the total size of the sub-images becomes too large and excessive pixelation occurs, causing a loss in the composite image fidelity. Intelligent image compression will outperform standard compression below the inflection point, but above standard compression performs better. Since we can calculate the inflection point before transmission, a dual approach could be taken, using standard compression above it, and intelligent compression below.

The JSON-LD metadata size is relatively small compared to the images, and spatially identifies the vehicles in the scene. Thus, it provides intrinsic value for both intelligent compression, and standard compressed images. The image is

what the camera sees, and the metadata is a model of the scene. If the scene has no vehicles, the semantic metadata alone may be sufficient to describe it.

Another technique might be caching vehicle sub-images between frames, as illustrated in Fig. 3, which depicts images five frames apart. As there are 25 fps, the car only moves a small distance between frames, and is likely to look very similar. The sub-image caching could use OpenCV's template matching feature, together with the inverse Euclidean distance between the bounding box centroids, to calculate the probability that this is the same vehicle from one frame to the next.



Fig. 3. Caching vehicle sub-images between frames

Then if the sub-image in the next frame, compared to the image in the previous frame, is above a certain probability, the sub-images in subsequent frames can reuse the image from the first frame. If the sub-image can be reused, there is no need to send successive sub-images, and instead the metadata from subsequent frames substitutes the ID of the first image in the sequence. When the frame's metadata is transmitted to the server, the server takes the sub-image from its cache of previous images.

In this way, the number of sub-images transmitted to the server, for any one given frame, can be reduced, and thus the freed-up bandwidth could be used to increase the JPEG quality of the other sub-images. It is difficult to predict the precise bandwidth savings, but we have estimated that savings of approximately 80% could be made for vehicle sub-images.

There are several other secondary benefits for implementing this technique, e.g. vehicle tracks, which is useful for calculating the vehicles speed and possible future trajectory. It is also an accurate method for counting vehicles, and enhancing the overall value of the metadata [12]. A small red dot in Fig. 3 marks the centroid of the vehicle in a given frame. Over time this sequence of dots form a track, which visualizes the trajectory of the vehicle. The metadata about a vehicle can be enhanced by connecting it to the metadata in previous frames. Thus, this metadata becomes a streamed knowledge graph, which semantically describes the road traffic scene over time, and, over which, an approach such as temporal stream reasoning [3] could be performed.

References

1. Chakraborty, P., Adu-Gyamfi, Y.O., Poddar, S., Ahsani, V., Sharma, A., Sarkar, S.: Traffic congestion detection from camera images using deep convolution neural networks. *Transp. Res. Rec.* **2672**(45), 222–231 (2018)
2. Chinchor, N., Sundheim, B.M.: MUC-5 evaluation metrics. In: Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, 25–27 August 1993
3. Della Valle, E., Ceri, S., Barbieri, D.F., Braga, D., Campi, A.: A first step towards stream reasoning. In: Domingue, J., Fensel, D., Traverso, P. (eds.) FIS 2008. LNCS, vol. 5468, pp. 72–81. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00985-3_6
4. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the KITTI dataset. *Int. J. Robot. Res.* **32**(11), 1231–1237 (2013)
5. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Pearson Education, London, United Kingdom (2007)
6. Gu, K., Lin, W., Zhai, G., Yang, X., Zhang, W., Chen, C.W.: No-reference quality metric of contrast-distorted images based on information maximization. *IEEE Trans. Cybern.* **47**(12), 4559–4565 (2016)
7. Lanthaler, M., Gütl, C.: On using JSON-LD to create evolvable RESTful services. In: Proceedings of the Third International Workshop on RESTful Design, pp. 25–32. ACM (2012)
8. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
9. Loce, R.P., Bernal, E.A., Wu, W., Bala, R.: Computer vision in roadway transportation systems: a survey. *J. Electron. Imaging* **22**(4), 041121 (2013)
10. Maarala, A.I., Su, X., Riekkki, J.: Semantic data provisioning and reasoning for the internet of things. In: 2014 International Conference on the Internet of Things (IOT), pp. 67–72. IEEE (2014)
11. Marr, B.: How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read. <https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#507c97c960ba>
12. Mehboob, F., Abbas, M., Rauf, A., Khan, S.A., Jiang, R.: Video surveillance-based intelligent traffic management in smart cities. In: Intelligent Video Surveillance. IntechOpen (2019)
13. Munroe, D.T., Madden, M.G.: Multi-class and single-class classification approaches to vehicle model recognition from images. In: Proceedings of the AICS, pp. 1–11 (2005)
14. Naphade, M., et al.: The 4th AI city challenge. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 2665–2674, June 2020
15. Networking, C.V.: Cisco global cloud index: forecast and methodology 2015–2020. White paper (2016)
16. Ozkurt, C., Camci, F.: Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. *Math. Comput. Appl.* **14**(3), 187–196 (2009)
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)

18. Rippel, O., Bourdev, L.: Real-time adaptive image compression. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 2922–2930. JMLR. org (2017)
19. Sikos, L.: Mastering Structured Data on the Semantic Web: From HTML5 Microdata to Linked Open Data. Apress, New York (2015)
20. Silverstein, D.A., Farrell, J.E.: The relationship between image fidelity and image quality. In: Proceedings of 3rd IEEE International Conference on Image Processing, vol. 1, pp. 881–884. IEEE (1996)
21. Su, X., Riekkki, J., Haverinen, J.: Entity notation: enabling knowledge representations for resource-constrained sensors. *Pers. Ubiquit. Comput.* **16**(7), 819–834 (2012)
22. Theis, L., Shi, W., Cunningham, A., Huszár, F.: Lossy image compression with compressive autoencoders. arXiv preprint [arXiv:1703.00395](https://arxiv.org/abs/1703.00395) (2017)
23. Thomas, S.S., Gupta, S., Subramanian, V.K.: Smart surveillance based on video summarization. In: 2017 IEEE Region 10 Symposium (TENSYP), pp. 1–5. IEEE (2017)
24. Tousch, A.M., Herbin, S., Audibert, J.Y.: Semantic hierarchies for image annotation: a survey. *Pattern Recogn.* **45**(1), 333–345 (2012)
25. Wallace, G.K.: The JPEG still picture compression standard. *IEEE Trans. Consum. Electron.* **38**(1), xviii–xxxiv (1992)
26. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
27. Wen, L., et al.: UA-DETRAC: a new benchmark and protocol for multi-object detection and tracking. *Comput. Vis. Image Underst.* **193**, 102907 (2020). Elsevier
28. Wu, W., Bernal, E.A., Loce, R.P., Hoover, M.E.: Multi-resolution video analysis and key feature preserving video reduction strategy for (real-time) vehicle tracking and speed enforcement systems, 10 February 2015. uS Patent 8,953,044
29. Xu, Z., Zhi, F., Liang, C., Mei, L., Luo, X.: Generating semantic annotation of video for organizing and searching traffic resources. *Int. J. Cognit. Inform. Nat. Intell. (IJCINI)* **8**(1), 51–66 (2014)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Aerial Object Detection for Water-Based Search & Rescue

Eoghan Mulcahy^(✉), Pepijn Van de Ven, and John Nelson

Electronic and Computer Engineering Department, University of Limerick,
Limerick, Ireland

{eoghan.mulcahy, pepijn.vandeven, john.nelson}@ul.ie

Abstract. Responding to a water rescue situation is challenging. First responders need access to data as quickly as possible to increase the likelihood of a successful rescue. Using aerial imagery systems is especially useful in a search and rescue scenario because it provides a higher dimensional view of the search environment. Unmanned aerial vehicles can be easily used to acquire aerial image data. During water-based search and rescue scenarios, first responders sometimes deploy an inflatable marker called a rescue danbuoy. The danbuoy is fitted with a small conical sack known as a drogue, this ensures that the marker is not blown off course by the wind and instead follows the flow of the body of water. Tracking the danbuoy as it moves is of utmost importance in a water rescue. We present a new data-set “VisBuoy” with imagery containing instances of danbuoy markers and boats in real-world water-based settings. We also show how using various deep learning-based computer vision techniques, we can autonomously detect danbuoy instances in aerial imagery. We compare the performance of four state-of-the-art object detectors Faster RCNN Retinanet, Efficientdet and YOLOv5 on the “VisBuoy” data-set, to find the best detector for this task. We then propose a best model with a precision score of 74% which can be used in search and rescue operations to detect inflatable danbuoy markers in water-based settings.

Keywords: Deep learning · Convolutional neural network · Object detection · Search and rescue

1 Introduction

Accurate and timely access to location based insights is key to successful search and rescue (SAR) operations. The most efficient situational awareness is achieved through aerial assessment [7]. Unmanned aerial vehicles (UAVs) are agile, fast and can be programmed to operate autonomously [25]. While aerial data acquisition alone helps obtain a bird’s-eye view during a rescue scenario, it presents a major challenge in processing a large amount of data to identify objects of interest in real-time [17]. Dealing with this data in real-time as a human is non-trivial, however, computer vision based object detection models provide a way to automatically search this data for objects of interest. This could be helpful for SAR first responders who can be guided by a sufficiently accurate algorithm to objects of interest visible in the UAV data.

Object detection is a computer technology related to computer vision that deals with detecting instances of semantic objects of various classes in digital imagery. Computational detection of objects of interest in a SAR mission is useful, it removes the need to manually review large amounts of data and allows for autonomous operations if required. In the recent past, deep-learning based object detection models have risen to prominence due to their higher performance compared to classical computer vision methods. Convolutional neural networks (CNNs) are state-of-the-art for object detection tasks and are used to great effect in many domains such as, medicine, automotive and space.

In this paper, we compare several state-of-the-art object detection models for performance on our novel data-set “VisBuoy”. We use the standardized detection performance metrics mean average precision and mean average recall. We find the most accurate object detector from this set and produce a model which can be used to detect danbuoy inflatable markers in a SAR scenario.

The paper is structured as follows: Sect. 2 details some related work. We outline our research methodologies in Sect. 3. We share the results of our experiments in Sect. 4 and we conclude with a summation of our results.

2 Related Work

Research into the use of UAVs for SAR has been popular in recent years. A number of studies have been conducted in disaster management [6] where UAV technology has been explored across all three disaster stages; pre-disaster preparedness [24], disaster assessment [8] and post-disaster response and recovery [10].

A subset of this research area comprises work on aerial image capture for UAV-assisted SAR missions [13]. Specifically, the task of automated object detection has been explored extensively. Approaches range from classical object detection methods such as edge detection and classification [4], to modern deep learning-based approaches, the latter achieving more accurate detections [2]. This research mainly focuses on the detection of people [5] on land rather than in water-based settings [11]. Our research takes a novel approach, instead detecting danbuoy inflatable markers via aerial imagery in water-based settings during SAR missions.

Many approaches take the route of examining the accuracy of one architecture on a public data-set. There are several drone-specific data-sets such as VisDrone [27] which are commonly used. We create a custom data-set as we are unaware of any publicly available danbuoy data-set at this time. There has been some research into the comparison of multiple state-of-the-art aerial image-based object detectors for vehicle [1] and person [20] detection. Our work focuses on a similar approach i.e. the comparison of multiple detectors in search of the best approach, but on the novel task of danbuoy inflatable marker detection in a water-based environment.

3 Methodology

3.1 Data-Set Generation

We gathered a custom data-set (Table 1) of danbuoy inflatable markers using a DJI Mavic Enterprise drone. We deployed a “Force 4 SOS Inflatable Danbuoy” (Fig. 1) into a river setting (Fig. 2) via a small boat. We captured a video through several UAV fly-overs at various altitudes, angles of approach and speeds resulting in a data-set of instance sizes (Fig. 3). Finally, we split the video into 1,279 frames using video to image conversion software [23] and labelled the images with the label-studio annotation tool [22].

Table 1. Data-set metrics

Metric	Danbuoy	Boat
Number of instances	532	387
Mean bounding box area	$9302px^2$	$24782px^2$
Standard deviation	8121	25835
Mode bounding box area	$2607px^2$	$2576px^2$
Median bounding box area	$5600px^2$	$14062px^2$
Min bounding box area	$1150px^2$	$1748px^2$
Max bounding box area	$54250px^2$	$118872px^2$



Fig. 1. Danbuoy inflatable marker



Fig. 2. Danbuoy deployed

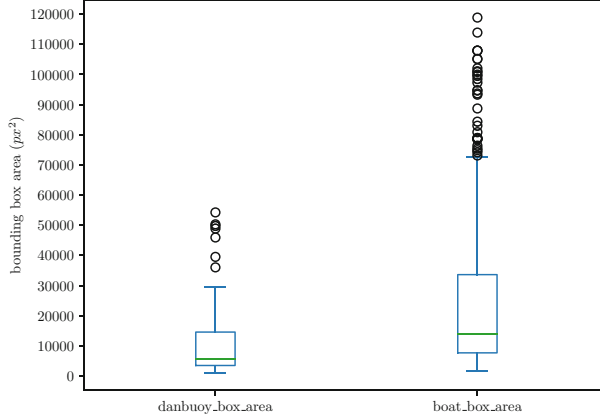


Fig. 3. Data-set instance bounding box area distributions

3.2 Model Development

To computationally detect instances of inflatable markers, four CNN based models were trained with an 80/20 train-validation split, on an NVIDIA GeForce RTX 2080 SUPER. Code was written to ensure all approaches could be validated against each other on the mean average precision and mean average recall metrics.

Intersection over union (IoU) (Fig. 4) is an important concept when evaluating the average precision. An IoU of 1 means that the ground truth and predicted bounding box are perfectly overlaid, while an IoU of 0 means the prediction has no overlap with the ground truth. We calculate the average precision (AP) (Eq. 1) by finding the area under the interpolated precision recall curve. Next, we calculate the average recall (AR) (Eq. 2) by finding the area under recall curve at each IoU level. To get the means (MAP and MAR) we average the AP/AR over all classes. For AP50 and AP75 we set and hold the IoU threshold at 50% and 75% respectively.

$$AP = \sum_n (R_{n+1} - R_n) P_{interp}(R_{n+1}) \quad (1)$$

where R_n is a unique recall value

$P_{interp}(R_{n+1})$ is the interpolated precision value

$$AR = 2 \int_{0.5}^1 recall(o) do \quad (2)$$

where o is IoU $[0.5:1]$ and $recall(o)$ is the corresponding recall

$$IoU = \frac{\text{area-of-overlap}}{\text{area-of-union}} = \frac{\text{Diagram of two overlapping squares}}{\text{Diagram of the union of two overlapping squares}}$$

Fig. 4. Intersection over union

Four state-of-the-art models were trained with Pytorch [16] as follows: Faster RCNN, Retinanet, Efficientdet and YoloV5. The models were configured as outlined in (Table 2). The learning rate, optimizer, image size, number of epochs and the batch size were kept constant to ensure a fair comparison. Commonly used backbone architectures were used for each model respectively. A short description of each model follows.

Faster RCNN is a two-stage detector [19] which consists of a deep fully convolutional neural network with a region proposal network and a detector that uses these proposals to generate predictions. It can be extended to return segmentation masks by adding another branch [12]. Faster RCNN has slower inference speeds than other detectors due to its large network parameter size.

Retinanet is a single-stage object detector which is widely used in satellite and aerial imagery. This detector was created as a competitor to two-stage detectors e.g. Faster RCNN which generally has higher accuracy at the cost of slower inference speeds. It utilizes a focal loss function [14] designed to focus on hard examples rather than allowing easy examples to skew the detector. The result is a detector which is faster and more accurate than many two-stage detectors.

YoloV5 is another single-stage detector designed for speed and can be optimized end to end due to its single network [18] detection pipeline. It is more prone to localization errors than two-stage detectors but is better at avoiding false detections and importantly it learns very general representations of objects.

Efficientdet is a detector designed for efficiency. It includes a novel bi-directional feature pyramid network (FPN) [21] allowing for feature fusion. It also scales resolution, depth and width for each of the networks (backbone, features, prediction) concurrently. Importantly, it achieves a higher AP on COCO [15] than many other SOTA models despite having (in our experiments) over 90% fewer parameters.

Using Pytorch Lightning Flash [9] we configured the training pipeline (Fig. 5) to ingest a Hydra [26] based configuration object so that we could easily run different models using the same underlying code. We wrote a custom validation loop so that all models could be easily compared under the MAP metric. We also implemented cloud-based logging with weights and biases [3] to ensure data provenance and reproducibility.

Table 2. Model configurations

Variables				Constants	
Model	Backbone	Trained Params	Total Params	Learning Rate	0.001
Faster RCNN	resnet50_fpn	41087011	41309411	Optimizer	SGD
Retinanet	resnet50_fpn	31987784	32210184	Image Size (px)	896x896
Efficientdet	d0	3826868	3826868	Epochs	50
YoloV5	medium	20879400	20879400	Batch Size	2

```

from flash.image import ObjectDetector
from torchmetrics.detection.mean_ap import MeanAveragePrecision as MAP
class MultiDetectLitModel(ObjectDetector):
    def __init__(
        self, num_classes,
        backbone, head,
        image_size, learning_rate,
        optimizer
    ):
        super().__init__()
        self.val_map = MAP()
    def training_step(self, batch, batch_idx):
        (xb, yb), records = batch
        preds = self(xb, yb)
        loss = self.adapter.loss_fn(preds, yb)
        for k, v in preds.items():
            self.log(f"train/{k}", v)
        return loss
    def validation_step(self, batch, batch_idx):
        (xb, yb), records = batch
        preds = self.adapter.model(xb)
        preds = [pred.as_dict()["detection"] for pred in preds]
        targets = [record.as_dict()["detection"] for record in records]
        map = self.val_map(preds, targets)
        self.log("val/map_dict", map,
            on_step=False, on_epoch=True, prog_bar=False)

```

Fig. 5. Model implementation

4 Evaluation

We trained the models for 50 epochs each and their validation metrics were logged on each epoch (Fig. 6). We evaluated the models under four standard metrics for state-of-the-art object detection, MAP, MAP50, MAP75 and MAR. By keeping some constant configuration values as shown earlier we ensured a fair comparison between the models.

The maximum score for each of the metrics was calculated and the models were ranked based on their performance (Table 3). We found that each model had merits under the various metrics, with three out of four models having a best-in-metric result.

YoloV5 scored best in MAR, though all models were similar under the MAR metric with a standard deviation of 0.007. In SAR scenarios, object detection models should prioritize precision over recall. High precision is a priority in SAR operations due to the possibility of false positive detections impeding the SAR team’s efforts.

Under the MAP50 metric models once again performed similarly. It is easier for the models to be deemed correct when holding the threshold for detection at 50% and so separating the models in terms of performance under this metric was difficult. Retinanet scored best outperforming Efficientdet by 0.89%.

The metrics which proved most useful in separating the models were MAP (all IoU ranges) and MAP75 held at 75% IoU. These metrics had the largest spread of values between each of the models and precision was the metric we prioritized most due to its importance in SAR as mentioned earlier. Efficientdet was the best model under the MAP metric, out-performing Retinanet by 9.58%. Efficientdet was also best under the MAP75 metric with a score 14% higher than the second-best model Retinanet.

As mentioned previously high precision is important in SAR operations to best assist the first-response team, as such, based on our evaluations we recommend Efficientdet be used for its high precision on the “VisBuoy” dataset. Some other factors in favour of Efficientdet include its lower power usage (Fig. 7) during training and the second highest inference (Fig. 8) speed compared to other models.

Table 3. Model performance metrics

Model Scores					Model Rank		
Model	MAP	MAP50	MAP75	MAR	Metric	First	Second
Faster RCNN	0.5677	0.9856	0.5710	0.4503	MAP	Efficientdet	Retinanet
Retinanet	0.5908	0.9966	0.6512	0.4443	MAP50	Retinanet	Efficientdet
Efficientdet	0.6474	0.9878	0.7426	0.4479	MAP75	Efficientdet	Retinanet
YoloV5	0.5248	0.9129	0.5202	0.4624	MAR	Yolov5	Faster RCNN

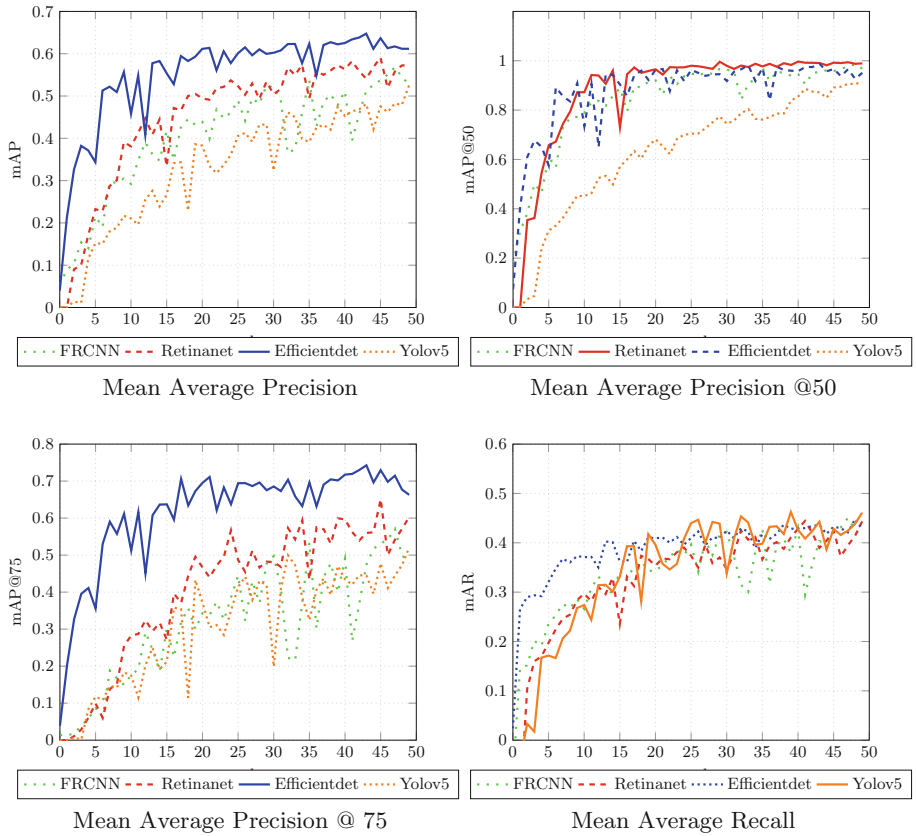


Fig. 6. Model comparison

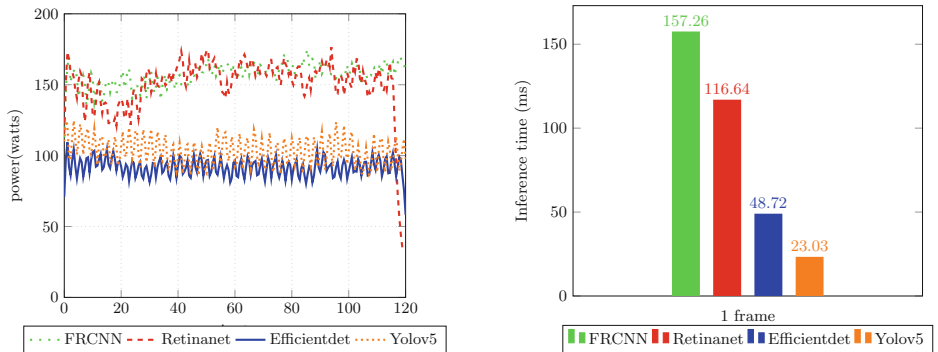


Fig. 7. Model power usage GPU

Fig. 8. Model inference speed

5 Conclusion

In this paper we compare the performance of four state-of-the-art object detection models on a data-set of danbuoy inflatable markers for water-based search and rescue scenarios. The data-set consisted of 1,279 images with 532 instances of danbuoys and 387 instances of boats.

Our analysis involved keeping some core hyper-parameters constant (learning rate, optimizer, image size, epochs and batch size) to allow for a fair comparison across all detectors. We rank the detectors based on their mean average precision and mean average recall in accordance with the standard object detection evaluation process. We rank each model in order of highest performance on our data-set as Efficientdet, Retinanet, YoloV5 and Faster RCNN.

As such, we recommend Efficientdet with a MAP75 score of 74% as the best model for detecting danbuoy inflatable markers from aerial imagery during SAR operations. Efficientdet has the added benefits of consuming less power while training and having the second fastest inference speed of all the models. We believe there are further improvements possible in future work for the Efficientdet model by exploring different combinations of the core hyper-parameter constants and varying the backbone.

UAV technology is already helpful in SAR efforts, providing a birds-eye view during operations. Extending this technology to add automated processing of the large amounts of data generated and providing precise location-based information to identify objects of interest in real-time is beneficial. Our research suggests Efficientdet as the best-in-class detection model to use for danbuoy inflatable marker detection in water-based SAR.

References

1. Acatay, O., Sommer, L., Schumann, A., Beyerer, J.: Comprehensive evaluation of deep learning based detection methods for vehicle detection in aerial imagery. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (2018). <https://doi.org/10.1109/avss.2018.8639127>
2. Akshatha, K.R., Karunakar, A.K., Shenoy, S.B., Pai, A.K., Nagaraj, N.H., Rohatgi, S.S.: Human detection in aerial thermal images using faster R-CNN and SSD algorithms. *Electronics* **11**, 1151 (2022). <https://doi.org/10.3390/electronics11071151>
3. Biewald, L.: Experiment tracking with weights and biases (2020). <https://www.wandb.com/>
4. Doherty, P., Rudol, P.: A UAV search and rescue scenario with human body detection and geolocalization. In: Orgun, M.A., Thornton, J. (eds.) *AI 2007. LNCS (LNAI)*, vol. 4830, pp. 1–13. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76928-6_1
5. Dousai, N.M.K., Lončarić, S.: Detection of humans in drone images for search and rescue operations. *APIT* (2021). <https://doi.org/10.1145/3449365.3449377>
6. Erdelj, M., Natalizio, E.: UAV-assisted disaster management: applications and open issues. Institute of Electrical and Electronics Engineers Inc. (2016). <https://doi.org/10.1109/ICCNC.2016.7440563>

7. Erdelj, M., Natalizio, E., Chowdhury, K.R., Akyildiz, I.F.: Help from the sky: leveraging UAVs for disaster management. *IEEE Pervasive Comput.* (2017). <https://doi.org/10.1109/mpv.2017.11>
8. Ezequiel, C.A.F., et al.: UAV aerial imaging applications for post-disaster assessment, environmental management and infrastructure development, pp. 274–283. *IEEE Computer Society* (2014). <https://doi.org/10.1109/ICUAS.2014.6842266>
9. Falcon: PyTorch lightning (2022). <https://github.com/PytorchLightning/pytorch-lightning>
10. Felice, M.D., Trotta, A., Bedogni, L., Chowdhury, K.R., Bononi, L.: Self-organizing aerial mesh networks for emergency communication, vol. 2014–June, pp. 1631–1636. *Institute of Electrical and Electronics Engineers Inc.* (2014). <https://doi.org/10.1109/PIMRC.2014.7136429>
11. Goodrich, M.A., et al.: Supporting wilderness search and rescue using a camera-equipped mini UAV. *J. Field Robot.* **25**, 89–110 (2008). <https://doi.org/10.1002/rob.20226>
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988 (2017). <https://doi.org/10.1109/ICCV.2017.322>
13. Kruijff, G.J.M., et al.: Rescue robots at earthquake-hit Mirandola, Italy: a field report (2012). <https://doi.org/10.1109/SSRR.2012.6523866>
14. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007 (2017). <https://doi.org/10.1109/ICCV.2017.324>
15. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014*. LNCS, vol. 8693, pp. 740–755. *Springer, Cham* (2014). https://doi.org/10.1007/978-3-319-10602-1_48
16. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library (2019). <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
17. Pi, Y., Nath, N.D., Behzadan, A.H.: Convolutional neural networks for object detection in aerial imagery for disaster response and recovery. *Adv. Eng. Inform.* **43**, 101009 (2020). <https://doi.org/10.1016/j.aei.2019.101009>
18. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection, vol. 2016–December, pp. 779–788. *IEEE Computer Society* (2016). <https://doi.org/10.1109/CVPR.2016.91>
19. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks, vol. 28. *Curran Associates, Inc.* (2015). <https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf>
20. Sambolek, S., Ivašić-Kos, M.: Automatic person detection in search and rescue operations using deep CNN detectors. *IEEE Access* (2021). <https://doi.org/10.1109/access.2021.3063681>
21. Tan, M., Pang, R., Le, Q.V.: EfficientDet: scalable and efficient object detection. pp. 10778–10787. *IEEE Computer Society* (2020). <https://doi.org/10.1109/CVPR42600.2020.01079>
22. Tkachenko, M., Malyuk, M., Holmanyuk, A., Liubimov, N.: Label studio: data labeling software (2020–2022). <https://github.com/heartexlabs/label-studio>
23. Tomar, S.: Converting video formats with FFmpeg. *Linux J.* **2006**, 10 (2006)

24. Ueyama, J., et al.: Exploiting the use of unmanned aerial vehicles to provide resilience in wireless sensor networks. *IEEE Commun. Mag.* **52**, 81–87 (2014). <https://doi.org/10.1109/MCOM.2014.6979956>
25. Waharte, S., Trigoni, N.: Supporting search and rescue operations with UAVs, pp. 142–147 (2010). <https://doi.org/10.1109/EST.2010.31>
26. Yadan, O.: Hydra - a framework for elegantly configuring complex applications (2019). <https://github.com/facebookresearch/hydra>
27. Zhu, P., et al.: Detection and tracking meet drones challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 7380–7399 (2021). <https://doi.org/10.1109/TPAMI.2021.3119563>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Cryptocurrency Volatility Index: An Efficient Way to Predict the Future CVI

An Pham Ngoc Nguyen^{1,2(✉)}, Martin Crane^{1,3}, and Marija Bezbradica^{1,3}

¹ School of Computing, Dublin City University, Dublin, Ireland
`ngocan.nguyenpham6@mail.dcu.ie`

² SFI Centre for Research Training in Artificial Intelligence, Dublin, Ireland

³ ADAPT Center for Digital Content Technology, Dublin, Ireland

Abstract. The Cryptocurrency Volatility Index (CVI index) has been introduced to estimate the 30-day future volatility of the cryptocurrency market. In this article, we introduce a new Deep Neural Network with an attention mechanism to forecast future values of this index. We then look at the stability and performance of our proposed model against the benchmark models widely used for time series prediction. The results show that our proposed model performs well when compared to popular methods such as traditional Long Short Term Memory, Temporal Convolution Network, and other statistical methods like Simple Moving Average, Random Forest and Support Vector Regression. Furthermore, we show that the well-known Simple Moving Average method, while it has its own advantages, has the weak spot when dealing with time series with large fluctuations.

Keywords: Cryptocurrencies · Volatility · CVI · LSTM · Attention mechanism

1 Introduction

The success of the cryptocurrency market can be seen through the constant increase in total market capitalization, going from 20 billion USD at the beginning of 2017 to over 3 trillion USD in 2021, and in the number of investors joining the community starting from 65 million in mid-2020 to over 300 million at the end of 2021. One reason for this expansion is due to the large gains that cryptocurrencies (especially Bitcoin) can bring to investors, thanks to dramatic fluctuations in prices [12]. Furthermore, unlike the stock market, the cryptocurrency market has much fewer restrictions, allowing investors to complete a transaction quickly and freely [38]. Unfortunately, the ease of trading in cryptocurrency makes it very vulnerable to external factors such as news of financial developments, the movement of other assets and even the statements from influencers [17, 29]. As a consequence, compared to traditional markets such as stocks, bonds and commodities, the volatility of the cryptocurrency market tends to be extremely high [39]. This can lead to large fluctuations, for instance, a study

conducted by Alexander et al. revealed that the losses of cryptocurrencies can reach 70% within one day [9]. Thus, the understanding of the volatility of the cryptocurrency market is essential to reduce the investment risk as well as to open opportunities to predict market's movements and gain profits.

Due to the demand to measure the volatility of the cryptocurrency market, the Cryptocurrency Volatility Index (CVI index) has been launched. This is defined as a measure of the 30-day future fluctuation degree of the price of the entire cryptocurrency market using the Black-Scholes option pricing model. In this way, an index that fluctuates between 0 and 200 is developed, such that 200 will indicate the maximum level of implied volatility in the market whilst a value of zero indicates the lowest volatility [10]. This index is intended to prevent investors from putting themselves at risk by modifying their trading strategy in line with different values of CVI. The higher the CVI value is, the greater the risks are but also the greater the potential return is.

The usefulness of the CVI index is the main motivation behind this work. Specifically, our objective is to answer the following research question:

- Is it possible to predict with accuracy the future value of the CVI index using neural networks?

We examine data from 10 cryptocurrencies which act as input features. Furthermore, we use the CVI index time series as labels for our prediction task. All time series span from 10/11/2017 to 12/01/2022. The details of this dataset are described in Sect. 3. Regarding the prediction model, we utilize Long Short Term Memory (LSTM), which is suitable for time series-related problems [16, 18], to extract these input features, followed by a Multilayer Perceptron (MLP) which allows the synthesis of the information extracted from the previous stage, the outputs of this stage being the predicted CVI values. Moreover, instead of using only the output from the last LSTM cell, which might lead to the losses of valuable information from earlier input data points, we use all outputs from the entire sequence of LSTM cells that are weighted according to their level of importance. We use a technique named *Attention mechanism* [41] for this weighing and name this novel neural network method AT-LSTM-MLP.

The remainder of this paper is organized as follows. Section 2 discusses related works. Section 3 introduces the dataset used in our experiments. Section 4 describes our AT-LSTM-MLP model and an outline of appropriate hyperparameters and specifications required to run experiments. Section 5 shows empirical results followed by an analysis learned from experiments. Section 6 gives conclusions for the article.

2 Related Works

Throughout the history of financial markets, there have been a number of prediction methods used to predict the future price and implied volatility of different assets such as stocks, bonds and cryptocurrencies.

Simple methods based on statistical learning frameworks have been found to show good performance in many studies, e.g. Simple Moving Average (SMA)

[2], Support Vector Regression (SVR) [11] and Random Forest (RF) [32]. The advantages of these statistical methods are that they are easy to implement, thus, the time complexity is significantly low and they tend to work well with different datasets. That is why these methods are often used as benchmark models to verify the efficiency of other methods even though there is a superiority from more recent methods [23,30].

Another commonly used method is GARCH, typically used to estimate the volatility of a time series such as stocks, bonds, market indices and recently, cryptocurrencies [7]. The first to use the GARCH model was Abdelhamid to estimate the volatility of different stocks circulating in the stock exchange of Casablanca at that time [13]. With time, there have been many variants of GARCH models proposed to optimize the prediction problem corresponding to a specific time series [4,20]. Until now, although the global financial market has changed considerably (i.e. more people invested, more investment options, etc.), the performance of GARCH-type models still seems to be good. The study [28] showed that the volatility of world currencies, namely the GBP, CAD, AUD, CHF and the JPY is effectively predicted by using a GARCH-type model called IGARCH. On the other hand, CGARCH and TGARCH models work well with major cryptocurrencies such as Bitcoin, Litecoin and Ripple. All the time series were observed from October 2015 to November 2019. For a longer period from 2010 to 2020, GARCH and a simpler corresponding model ARCH have also been successfully used to form variance equations for highly-capitalized cryptocurrencies [5].

Neural Networks were first introduced in 1944 by McCulloch and Pitts from the University of Chicago [25]. Since then, they have been applied to many different areas from healthcare services [15] to daily utility applications [42] and entertainment purposes [40]. In Finance, there is an increasing tendency to use RNNs to understand the operation of financial markets [1]. This is because they tend to work very well with time series data and are often integrated into prediction-related problems [18,31]. LSTM [14], a version of Recurrent Neural Networks, appears to be dominant because of its ability to recollect longer so that information in the architecture can travel deeper. In [27], Aditi et al. made a comparison between Polynomial Regression, RNN, LSTM and ARIMA [6] for Bitcoin price fluctuation prediction. To do the experiment, the authors used a dataset of a combination between daily close price, volume and social media-related information such as Google Trend index, tweets' emotions as well as the number of posts on Twitter containing the keyword "Bitcoin". Out of four methods, LSTM outperformed the others with an accuracy of approximately 83%. Elsewhere, the information about Bitcoin blockchain technology has also been exploited to predict the future price. Specifically, Suhwan et al. investigated the performance of different Deep Learning models for estimating Bitcoin prices one day ahead [19]. In this study, 30 features were collected with one of them being daily closing price of Bitcoin and the rest being about blockchain specifications such as the average block size, the relative measure of difficulty in finding a new block, the total number of blockchain wallets created, etc. The results showed that LSTM

slightly outperformed Multi-layer perceptrons and a pre-trained model ResNet. Another common Neural Network architecture is Convolution Neural Network (CNN), widely applied to analyze image data [35]. Recently, this type of Neural Network has been incorporated into financial research and has achieved certain successes. In particular, the authors in [3] used CNN as a part of their model to predict the future Bitcoin volatility and named it Temporal Convolution Network (TCN). Moreover, CNN can be incorporated into a LSTM model to improve the efficiency of the prediction. A method introduced in [24] for the prediction of Bitcoin, Ethereum and Ripple prices on the following hour combined three different Deep Learning models, including LSTM, Bidirectional LSTM and CNN, then used a statistical method such as Support Vector Regression or k-Nearest Neighbour to average the weighted output of each single model. It was confirmed that this practice helped improve the performance compared to individual models. In general, a large variety of methods for future volatility prediction exist in both traditional and cryptocurrency markets that use RNNs. All of these methods show an outperformance compared to GARCH-type models [16, 23].

3 Dataset

In order to answer our research question, we have chosen 77 digital currencies that meet the following three criteria:

- (i) Large market capitalization: The market capitalization is greater than 1 million USD.
- (ii) Long historical time series: The number of trading days is greater than 1000.
- (iii) Few missing values: The percentage of missing values is less than 10%.

Furthermore, we use the CVI Index, which we have introduced in the previous section acting as labels for our proposed prediction model. The dataset used in this work was collected from two different sources: for cryptocurrencies' historical data, we downloaded time series' closing prices on the open financial website [Yahoo! Finance](#); for CVI Index, the data was collected from the website [Investing.com](#). All time series are collected day by day.

However, the trading schedule of each cryptocurrency is different, resulting in inconsistency across the entire dataset. Due to the difference in trading dates, if we attempted to use all the historical time series, we might end up with an empty dataset. For this reason, we chose 10 out of 77 cryptocurrencies that are widely known and have the longest historical data, the ultimate input features comprise BTC, ETH, BCH, ADA, XRP, DOGE, LINK, LTC, XLM and ETC. Each of them has 1537 data points starting from 10/11/2017 until 12/1/2021. When it comes to our labels, CVI Index begins on 31/3/2019 and ends on the same day as the other time series, the total number of data points is 1019. Subsequently, after merging all available data together (10 cryptocurrency time series and CVI time series), we have a 1019 data point dataset that consists of 10 cryptocurrencies acting as input features and CVI index acting as labels.

Since the original input features were collected daily from closing prices, it is necessary to convert this type of data into daily volatility:

Step 1: Calculating daily return of each cryptocurrency by using daily closing prices: $y_t = \log\left(\frac{Close_t}{Close_{t-1}}\right)$. Where y_t , $Close_t$ are daily return, close price at timestamp t , respectively.

Step 2: Calculating daily volatility from daily return, as suggested by Fernando et al [33]: $\sigma_t = \sqrt{\frac{1}{5} \sum_{i=0}^4 y_{t-i}^2}$. Where σ_t is realized volatility at timestamp t .

4 Methodology

4.1 Attention Based-Deep Neural Network (AT-LSTM-MLP)

We create a novel Neural Network by combining 2 different types of Neural Network models, Long Short Term Memory [14] and Multilayer Perceptron (MLP) [22] and adding Attention Mechanism to weight the degree of importance for each LSTM cell. A diagram for this architecture is shown in Fig. 1.

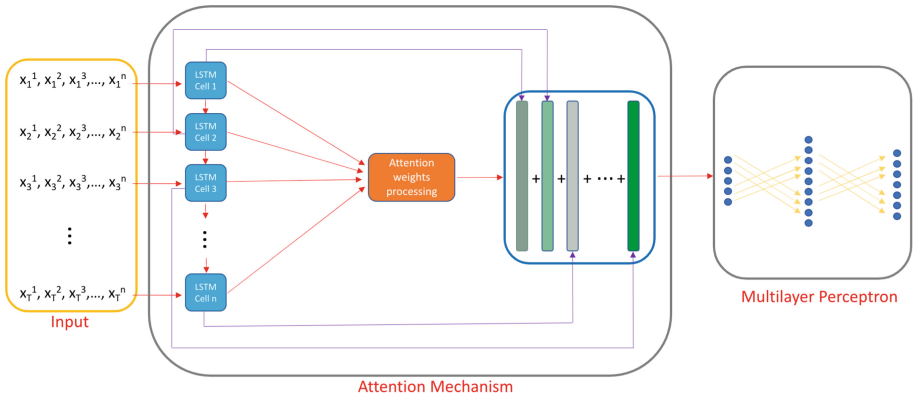


Fig. 1. AT-LSTM-MLP model. x_k refers to time series values at timestamp k , $k = (1, 2, \dots, T)$, x^l refers to time series values of l^{th} input feature, $l = (1, 2, \dots, n)$. The model comprises 2 parts: Long Short Term Memory with Attention Mechanism and Multilayer Perceptron. The output of the model are future predicted CVI indices.

Our proposed architecture (AT-LSTM-MLP) takes advantage of the outstanding characteristics not only of LSTM but also Attention Mechanism by using the weighted outputs of all LSTM cells from the sequence. These LSTM outputs will go to a Multilayer Perceptron where the future CVI index is predicted at the end. The process can be described as follows:

Firstly, we determine the shape of input data, each row is one timestamp while each column is one input feature. In our defined formula, we set T as the number of timestamps and n as the number of input features. We use the matrix form below as an input of our model:

$$\begin{pmatrix} x_1^1 & x_1^2 & x_1^3 & \dots & x_1^n \\ x_2^1 & x_2^2 & x_2^3 & \dots & x_2^n \\ x_3^1 & x_3^2 & x_3^3 & \dots & x_3^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_T^1 & x_T^2 & x_T^3 & \dots & x_T^n \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_T \end{pmatrix} \quad (1)$$

Secondly, we move this input forward to a sequence of LSTM cells where each single LSTM cell (f_t) takes one row of the input (x_t) sequentially with the number of LSTM cells equal to the number of rows. The output at this phase is a set of hidden states at each LSTM cell (h_1, h_2, \dots, h_T): $h_t = f_t(x_t, h_{t-1})$. Since LSTM takes the previous hidden state as an input argument, we set as default that the original hidden state (h_o) is zero, meaning that there no information from the time series revealed to the first LSTM cell. The hidden states ($h_1, h_2, h_3, \dots, h_T$) then go through the self-attention mechanism. The output at this stage, denoted as s , is obtained by the following mathematical equations:

$$u_k = \tanh(W \times h_k) \quad (2a)$$

$$a_k = \frac{\exp(\text{score}(u_k, u))}{\sum_{t=1}^T \exp(\text{score}(u_t, u))} \quad (2b)$$

$$s = \sum_{i=1}^T a_i h_i \quad (2c)$$

where W and u are training parameters, a_1, a_2, \dots, a_n are Softmax coefficients. The Eq. 2a can be interpreted as a representation of a Fully Connected layer once each hidden state is passed to this layer in order to be embedded in a new vector space. We define an alignment function named *score* to measure how important each embedded hidden state u_t is. Theoretically, *score* can be any function depending on the research questions. In this work, we choose a straightforward function $\text{score}(u_k, u) = u_k^T u$. This function will return a scalar value for each corresponding embedded hidden state. We use Softmax function to map the original values to a probability distribution where all the components add up to 1, the larger input components will correspond to larger probabilities. The last Eq. 2c is a weighted sum of all considered hidden states as well as the output of this attention mechanism.

Thirdly, the process continues moving to a Multilayer Perceptron with the predicted CVI index y_{CVI} obtained at the end: $y_{CVI} = MLP(s)$.

4.2 Training Parameters and Implementation of Our Proposed Model

Our aim is to predict one-day future CVI index using 10 input features, which are 10 cryptocurrency time series. We run our proposed model on Tesla K80 GPU

with memory size of 12 GB. We take the last 20% of the dataset for test set in order to evaluate the model's performance, while the remaining data is used for training. All trainable parameters we initialize randomly following normal distribution with mean of 0. We built the model and ran all mentioned methods using Pytorch [34]. The details for the model specifications are described in Table 1, the optimal value according to each specification is shown in bold. We note that Mean Absolute Error is chosen as the loss function for our model. This is due to the lowest errors in the test set which were calculated using this metric.

4.3 Training Parameters and Implementation of Benchmark Models

The performance of our new method is verified by comparing with the following five different techniques:

1) Simple Moving Average [37]

We run the Simple Moving Average method for various window sizes according to the [Autocorrelation](#) plot. From this, we noticed 30 spikes outside of the blue area and thus statistically significant. Since a small change in the window sizes could lead to a small difference among the results, we do not use all possible window sizes. Instead, we choose window sizes = 2, 3, 4, ..., 10, 15, 20, 25 and 30.

2) Support Vector Regression [26]

We use the Radial Basis Function kernel (RBF kernel) to find the optimal hyperplane for this regression problem because our data is clearly non-linear [36]. The regularization parameter is set to 1.0 to achieve a lower generalization error. To capture the shape of the data efficiently, we choose Gamma coefficient $\gamma = 0.1$, which is a parameter attached to the RBF kernel. All observations in the train set are put into SVR to estimate the most optimal solution for future prediction.

3) Random Forest [21]

We tested different possible values for the number of Decision Trees and found that 100 gives the best result. At each internal node of a Decision tree, we randomly choose three input features to consider when looking for the best split. The optimal feature will be chosen based on Variance Reduction technique. The minimum number of samples required to split an internal node is 2, the minimum number of samples at one leaf is 1. We use the average operation to synthesize the results of all Decision Trees. All observations in the train set are put into RFm to estimate the most optimal solution for future prediction.

4) Standard Long Short Term Memory [14]

This model comprises 20 LSTM cells with the size of hidden state being 1. We use the hidden state of the last LSTM cell as the output of the model. We train this model in 7000 epochs with a learning rate of 0.005. The other parameters used in this model are similar to our original model AT-LSTM-MLP.

5) Temporal Convolution Network [3]

From our original model, we replace the LSTM block with a TCN block. As each input of this model has 10 features and 20 timestamps, we set a kernel with a length of 5 and the number of channels of 10 to slide on the input. We choose $\text{dilation} = 2$. We train this model in 5000 epochs with a learning rate of 0,001. The other parameters used in this model are similar to AT-LSTM-MLP.

5 Empirical Results and Analysis

Table 1. Specifications used in AT-LSTM-MLP, the best results are shown in bold.

Specifications	Values/Methods
Loss function	Mean Absolute Error (MAE) Mean Squared Error (MSE) Root Mean Square Error (RMSE) Mean Absolute Percentage Error (MAPE) Symmetric Mean Absolute Percentage Error (SMAPE)
Optimizer	Stochastic Gradient Decent (SGD) Nesterov Accelerated Gradient (NAG) Adagrad AdaDelta Adam RMSPProp
#Time stamps	5, 10, 15, 20 , 25, 30, 35
LSTM's hidden state size	10, 20, 30, ... , 100 , 110, 120
#Fully Connected layers	1, 2, 3 , 4, 5
1st Fully Connected size	16, 32, 64 , 128, 256
2nd Fully Connected size	16, 32 , 64, 128, 256
3rd Fully Connected size	1
Batch size	16, 32 , 64
#Epochs	5000, 7000 , 10000
Learning rate	0.0001, 0.001, 0.005 , 0.1

We measure the error of the methods through three different metrics: MAE, RMSE, and SMAPE [8]. For methods when results vary at every single run, we run each model 10 times, then get the mean μ and standard deviation σ from all single results to give the final figure that is represented as $\mu \pm \sigma$. For static methods, such as Simple Moving Average and Support Vector Regression, we run them once and consider the result as the final result.

Table 2 shows the results of our proposed model and four other models used for comparison. AT-LSTM-MLP, which is our model, outperforms other methods

with RMSE, MAE, and SMAPE of 2.24 ± 0.17 , 1.62 ± 0.09 , 1.76 ± 0.10 , respectively. LSTM comes into second place with errors at more than double comparing to AT-LSTM-MLP. Whereas, the three remaining methods show poor results as the predicted values are too far from real values. These results give an answer to our research question that AT-LSTM-MLP can predict the future value of the CVI index well.

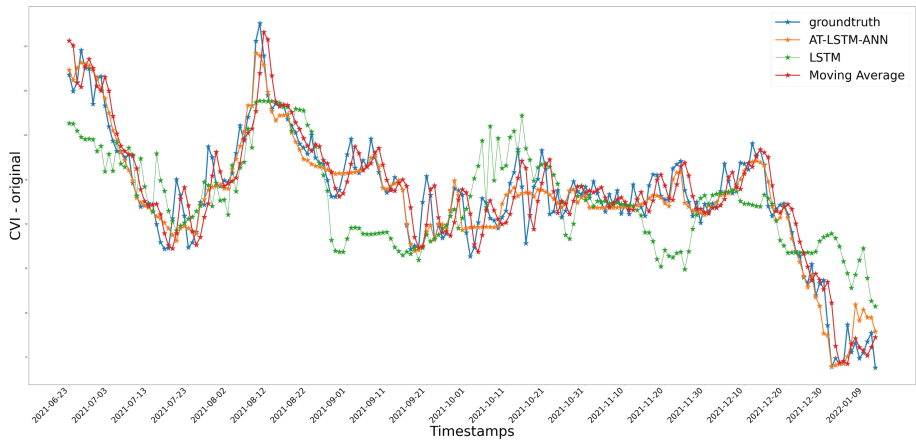


Fig. 2. Predicted values and real values, using our proposed method and some aforementioned benchmarks (only benchmark models with good performance are illustrated).

Table 2. Empirical results of 5 different methods measured in three metrics RMSE, MAE and SMAPE. The best results are shown in bold.

Models	RMSE	MAE	SMAPE
AT-LSTM-MLP	2.24 ± 0.17	1.62 ± 0.09	1.76 ± 0.10
LSTM	5.54 ± 0.85	4.11 ± 0.57	4.50 ± 0.64
SVR	12.69	10.42	11.84
RF	14.25 ± 0.17	12.20 ± 0.14	13.69 ± 0.36
TCN	9.20 ± 0.13	7.35 ± 0.11	8.10 ± 0.12

Table 3 show the results of SMA with different sizes of sliding window. In general, we can conclude that SMA works well with the data. However, the accuracy seems to fall off gradually as the sliding window size increases, the first sliding window sizes perform better than all other benchmarks with the exception of our method (AT-LSTM-MLP). SMA with sliding window size of 2 yields the best result of 2.02, 2.63, and 2.16 in MAE, RMSE, and SMAPE, respectively. This performance indicates the weakness of the SMA method. Particularly, SMA

Table 3. Empirical results of Simple Moving Average with different window sizes. The best results are shown in bold.

Window size	2	3	4	5	6	7	8	9	10	15	20	25	30
RMSE	2.63	2.89	3.04	3.13	3.19	3.31	3.46	3.63	3.83	4.59	5.20	5.68	6.13
MAE	2.02	2.27	2.42	2.49	2.54	2.64	2.77	2.92	3.07	3.57	3.92	4.29	4.69
SMAPE	2.16	2.43	2.60	2.67	2.72	2.82	2.95	3.12	3.28	3.82	4.22	4.62	5.02

only works well with stable data, i.e. when the difference between time stamps within a sliding time window is small.

An illustration of predicted and real values using AT-LSTM-MLP and benchmark models is shown in Fig. 2 (only methods with good performance are illustrated).

6 Conclusions

We have built a novel Deep Neural Network by combining Long Short Term Memory, Multilayer Perceptron and Attention mechanism, named AT-LSTM-MLP. Our goal is to predict the day-ahead CVI index using ten long-series cryptocurrencies with large market capitalization. By comparing our proposed method with five benchmark models in three different metrics, we show that AT-LSTM-MLP performs better than the others. Moreover, we also indicate the main weakness of SMA: it only works well with stable data, when the difference between time stamps within a sliding time window is small.

The results from this study contribute to literature on the cryptocurrency market with some useful tools and information that aim to helping the investors in making decisions in investment. We believe that our method can be applied to other prediction tasks that involve time series because of its good performance.

Acknowledgements. The authors Martin Crane, Marija Bezbradica wish to acknowledge the support, in part, from the Science Foundation Ireland under Grant Agreement No. 13/RC/2106.P2 at the ADAPT SFI Research Centre at DCU. ADAPT, the SFI Research Centre for AI-Driven Digital Content Technology, is funded by the Science Foundation Ireland through the SFI Research Centres Programme. The author An P. N. Nguyen wishes to acknowledge the support from Dublin City University’s Research Committee and research grants from Science Foundation Ireland Centre for Research Training in Artificial Intelligence under grant number 18/CRT/6223.

References

1. Saeed, M.: An introduction to recurrent neural networks and the math that powers them (2021)
2. Abu Bakar, N., Rosbi, S., Uzaki, K.: Forecasting cryptocurrency price movement using moving average method: a case study of Bitcoin cash. *J. Adv. Res.* **7**(12), 609–614 (2019)

3. Akbiyik, E., Erkul, M., Kaempfe, K., Vasiliauskaite, V., Antulov-Fantulin, N.: Ask “who”, not “what”: Bitcoin volatility forecasting with Twitter data. arXiv preprint [arXiv:2110.14317](https://arxiv.org/abs/2110.14317) (2021)
4. Ali, G.: EGARCH, GJR-GARCH, TGARCH, AVGARCH, NGARCH, IGARCH and APARCH models for pathogens at marine recreational sites. *J. Stat. Econ. Methods* **2**(3), 57–73 (2013)
5. Almansour, B., Alshaterand, M., Almansour, A.: Performance of ARCH and GARCH models in forecasting cryptocurrency market volatility. *Ind. Eng. Manag. Syst.* **20**(2), 130–139 (2021)
6. Ariyo, A., Adewumi, A., Ayo, C.: Stock price prediction using the ARIMA model. In: 2014 UKSim-AMSS 16th International Conference on Computational Modelling, Simulation, pp. 106–112. IEEE (2014)
7. Bhowmik, R., Wang, S.: Stock market volatility and return analysis: a systematic literature review. *Entropy* **22**(5) (2020)
8. Botchkarev, A.: Performance metrics (error measures) in machine learning regression, forecasting and prognostics: properties and typology. arXiv preprint [arXiv:1809.03006](https://arxiv.org/abs/1809.03006) (2018)
9. Brauneis, A., Mestel, R.: Price discovery of cryptocurrencies: Bitcoin and beyond. *Econ. Lett.* **165**, 58–61 (2018)
10. Briefing, C.: What is the crypto volatility index (CVI)? (2021)
11. Dash, R.K., Nguyen, T., Sharma, A., Cengiz, K., Sharma, A.: Fine-tuned support vector regression model for stock predictions. *Neural Comput. Appl.* (2021)
12. Dong, F., Xu, Z., Zhang, Y.: Bubbly Bitcoin. *Econ. Theory* 1–43 (2021)
13. El Bouhadi, A.: Conditional volatility of most active shares of Casablanca stock exchange. *Econometrica* **50**(1) (1982)
14. Graves, A.: Long short-term memory. In: Graves, A. (ed.) *Supervised Sequence Labelling with Recurrent Neural Networks*. SCI, vol. 385, pp. 37–45. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24797-2_4
15. Han, C., Rundo, L., Murao, K., Noguchi, T.: MADGAN: unsupervised medical anomaly detection GAN using multiple adjacent brain MRI slice reconstruction. *BMC Bioinform.* **22**(31) (2021)
16. Hu, Y., Ni, J., Wen, L.: A hybrid deep learning approach by integrating LSTM-ANN networks with GARCH model for copper price volatility prediction. *Physica A* **557** (2020)
17. Huynh, T.: When Elon Musk changes his tone, does Bitcoin adjust its tune? *Comput. Econ.* (2022)
18. Jay, P., Kalariya, V., Parmar, P., Tanwar, S., Kumar, N., Alazab, M.: Stochastic neural networks for cryptocurrency price prediction. *Blockchain-Enabled Trustworthy Syst.* **8** (2020)
19. Ji, S., Kim, J., Im, H.: A comparative study of Bitcoin price prediction using deep learning. *Mathematics* **7**(10), 898 (2019)
20. Katsiampa, P.: Volatility estimation for Bitcoin: a comparison of GARCH models. *Econ. Lett.* **158**, 3–6 (2017)
21. Khaidem, L., Saha, S., Dey, S.R.: Predicting the direction of stock market prices using random forest. *Appl. Math. Finan.* (2016)
22. Krogh, A.: What are artificial neural networks? *Nat. Biotechnol.* **26**(2), 195–197 (2008)
23. Liu, Y.: Novel volatility forecasting using deep learning-long short term memory recurrent neural networks. *Expert Syst. Appl.* **132**, 99–109 (2019)
24. Livieris, I., Pintelas, E., Stavroyiannis, S., Pintelas, P.: Ensemble deep learning models for forecasting cryptocurrency time-series. *Algorithms* **13**(5) (2020)

25. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biol.* **5**(4), 115–133 (1943)
26. Meesad, P., Rasel, R.I.: Predicting stock market price using support vector regression. In: 2013 International Conference on Informatics, Electronics and Vision (ICIEV), pp. 1–6 (2013)
27. Mittal, A., Dhiman, V., Singh, A., Prakash, C.: Short-term Bitcoin price fluctuation prediction using social media and web search data. In: 2019 Twelfth International Conference on Contemporary Computing (IC3), pp. 1–6. IEEE (2019)
28. Naimy, V., Haddad, O., Fernández-Avilés, G., El Khoury, R.: The predictive capacity of GARCH-type models in measuring the volatility of crypto and world currencies. *PLoS One* **16**(1) (2021)
29. Nguyen, A.P.N., Mai, T.T., Bezbradica, M., Crane, M.: The cryptocurrency market in transition before and after Covid-19: an opportunity for investors? *Entropy* **24**(9), 1317 (2022)
30. Nguyen, H.V., Naeem, A., Wichitaksorn, N., Pears, R.: A smart system for short-term price prediction using time series models. *Comput. Electr. Eng.* **76**, 339–352 (2019)
31. Nguyen-Pham, N.A., Nguyen, T.: An efficient hybrid mechanism with LSTM neural networks in application to stock price forecasting. In: Knowledge Innovation Through Intelligent Software Methodologies, Tools and Techniques: Proceedings of the 19th International Conference on New Trends in Intelligent Software Methodologies, Tools and Techniques (SoMeT_20), vol. 327, p. 447 (2020)
32. Park, J.S., Sung Cho, H., Sung Lee, J., Chung, K.I., Kim, J.M., Kim, D.J.: Forecasting daily stock trends using random forest optimization. In: 2019 International Conference on ICT Convergence (ICTC), pp. 1152–1155 (2019)
33. Pérez-Cruz, F., Afonso-Rodríguez, J.A., Giner, J.: Estimating GARCH models using support vector machines. *Quant. Finan.* **3**(3), 163 (2003)
34. PyTorch: PyTorch Tutorials (2022). <https://pytorch.org/tutorials/>
35. Quach, B.M., Dinh, V.C., Pham, N., Huynh, D., Nguyen, B.T.: Leaf recognition using convolutional neural networks based features. *Multimed. Tools Appl.* 1–25 (2022)
36. Ramedani, Z., Omid, M., Keyhani, A., Shamshirband, S., Khoshnevisan, B.: Potential of radial basis function based support vector regression for global solar radiation prediction. *Renew. Sust. Energ. Rev.* **39**, 1005–1011 (2014)
37. Raudys, A., Pabarskaite, Z.: Optimising the smoothness and accuracy of moving average for stock price data. *Technol. Econ. Dev. Econ.* **24**(3), 984–1003 (2018)
38. Reiff, N.: What are the advantages of paying with Bitcoin? (2021)
39. Rogojanu, A., Badea, L.: The issue of competing currencies. *Theor. Appl. Econ.* **590**(1), 103–114 (2014)
40. Silver, D., et al.: Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint [arXiv:1712.01815](https://arxiv.org/abs/1712.01815) (2017)
41. Tixier, A.: Notes on deep learning for NLP. Computer Science Department (DaSciM team), École Polytechnique, Palaiseau, France (2018)
42. Zhang, X., Zhou, Y., Wang, J., Lu, X.: Personal interest attention graph neural networks for session-based recommendation. *Entropy* **23**(11) (2021)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Exploring Abstractive vs. Extractive Summarisation Techniques for Sports News

Ahmed Jouda^(✉) 

University College Dublin, Dublin 4 D04 V1W8, Ireland
ahmedjudah00@gmail.com

Abstract. The high demand generated by the information age has led to recent breakthroughs in both extractive and abstractive text summarisation. This work explores the algorithms that were the product of these advances, focusing on the domain of sports news summarisation. By creating a new hybrid evaluation system that incorporates automatic evaluation metrics, such as ROUGE and BLEU scores, with human evaluation, we observe that abstractive techniques return the best results in the sports domain. This also generalises to the domain of political articles. However, here the metrics report lower scores across most algorithms. Another finding is that the algorithms considered perform independently of the dialect of English used.

Keywords: Summarisation · Performance · Extractive · Abstractive

1 Introduction

Automatic text summarisation is described as the reduction of a text to its important content [1]. There has been significant progress in the area in the last 50 years. However, this task still poses many challenges to the scientific community. Sports news summarisation is one application of such techniques, which focuses on generating rich and concise text summaries that capture the essence of sports-related articles. There are two key methods used in text summarisation, namely extraction and abstraction. The former strategy selects sentences and phrases from the original text as a summary, while the latter strategy can potentially involve the generation of new relevant content to use in a summary.

This paper looks at two main problems in the field of automatic text summarisation. Firstly, are abstractive or extractive algorithms more appropriate for summarisation in particular domains of interest? Secondly, how do we effectively evaluate the summaries generated by different algorithms? There exist a number of automatic evaluation metrics that are widely applied in this area [2,3]. However, these are sometimes not sufficient on their own. This paper proposes a new hybrid evaluation system that uses both automatic evaluation metrics alongside human evaluation.

Through investigating the above problems, this paper aims to achieve a number of goals. Firstly, finding the best algorithms and comparing their performance

in different article domains. This is done to examine if an algorithm's performance is domain-dependent. This will enable researchers to determine what summarisation algorithm to consider, depending on the domain of the corpus that they want to summarise. Secondly, exploring what effect different dialects might have on the performance of the summarisation algorithms. Finally, in terms of evaluation, we aim to develop a hybrid system to evaluate the performance of the algorithms when summarising sports news and to discover if automatic evaluation metrics correlate well with human judgements.

2 Background Research

Interest in the area of text summarisation started to increase in the 1950's due with increases in news production and consumption. The first research paper to address this task was published in 1958 at IBM and targeted the automatic creation of literature abstracts [4]. The paper set multiple ground rules that are now the fundamental steps in most summarisation tasks. These include reverting words to their stems, removing stop words, and having some form of a table of words with corresponding significance scores.

Extractive text summarisation techniques determine which words, phrases, and/or sentences from the original text are incorporated into a short summary [5]. In this case no new content is created by the algorithm. Instead, the entire summary is derived from the original author's words. One of the most popular extractive algorithms is TextRank [6]. Based on Google's PageRank algorithm, it is an unsupervised technique that extracts keywords and sentences from text documents to use as a summary. A graph is used where sentences are vertices and the weight of the edges depends on a given sentence similarity function. Extractive methods are mainly concerned with the content of the summary, rather than how it reads and flows.

Abstractive text summarisation creates and combines an abridgment by using vocabulary words. This is then put together to give the summary of the original document [5]. These algorithms generate a concise summary that captures the essence and primary ideas of the original document, but can also contain new phrases that were not in the original text. There is a great emphasis on the form of the summary and its grammatical structure. A popular example of an abstractive technique summarisation is BART [7]. This technique includes a pre-training approach – it is first trained by firstly corrupting text with an arbitrary noising function, and it then learns a model to reconstruct the original text.

In order to compare the performance of different summarisation algorithms, appropriate evaluation techniques are required. Over time, evaluation approaches have varied from manual inspection, comparison to a human-written summary, and more complex scoring measures, such as ROUGE and BLEU. It is also possible to construct new evaluation techniques as was done in [5], depending on the use case of the system. The authors in that paper proposed an evaluation technique called, PolyTope, which is an error-oriented fine-grained human evaluation method consisting of 8 elements or issue types. Work in this area is ongoing and choosing the right metric is often subjective and domain-dependent.

There has been very limited work in the specific area of sports news summarisation. All of the above extractive and abstractive techniques can be applied to sports news articles, as they are simply documents of unstructured text. However, it is interesting to see which algorithms perform better with sports articles and if the algorithm performance is affected by the domain of news. Initial work in [8] discusses the summarisation of live sports commentary from Chinese news outlets. The authors suggested that a new algorithm is needed to deal with the different writing styles based on different languages and cultures. The proposed algorithm, SportsSum, outperformed various baseline methods.

Based on a review of the literature, we are left with two main considerations. There are no current papers that discuss and compare the different algorithms in both extractive and abstractive summarisation in the domain of sports news. Secondly, there is currently no user interface that helps users to pick which summarisation algorithm to use, based on pre-defined text input examples. In this work we try to address both these limitations.

3 Data Considerations

The data set that was chosen for use in this paper is the benchmark CNN/DM news collection [9]. This corpus is labelled, which means the news articles are associated with highlights. These highlights are short human-written summaries that can be used to in the evaluation of automatically-generated summaries. A version of the data which was pre-processed into clean CSV files was used¹. For the purpose of this comparative study, the data was combined into a single representation, consisting of 311,971 articles in total.

Following that, these articles needed to be split into two groups: (i) US CNN articles; (ii) UK Daily Mail (DM) articles. This was done by filtering articles that start with (*CNN*). This excludes CNN articles from other countries and DM articles. DM articles were identified by finding articles that start with ‘*By . Daily Mail*’. This resulted in two sets, one with CNN articles containing 50,335 articles and the other with 149,884 Daily Mail articles.

In addition, the generalisation of summarisation algorithms across two different domains was explored. The set from each news outlet was filtered to find only sports and politics articles, both identified using manual sets of keywords. An article needed to contain two or more keywords to be categorised as part of that domain. This resulted in 661 CNN sports and 351 politics articles, while the Daily Mail (DM) provided 2,106 sports and 682 politics articles. These are treated as four separate sets for later experimentation.

4 Implementation

We conduct a comparative study to identify the best algorithms and compare their performance in different domains. We also explore the difference in performance depending on the English dialect used (British English vs American

¹ https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail?select=cnn_dailymail.

English). A human evaluation was also performed to support this comparison and determine whether automatic evaluation agrees with human judgement.

4.1 Comparative Study

This comparative experiment was implemented using Python 3, based upon the libraries *sumy* [10] and *transformers* [11]. The purpose of this study is to generate multiple summaries for every single article to compare the performance of each of the algorithms. We focus on four popular algorithms. The two extractive ones are LexRank and LSA, while the abstractive ones are BART and Pegasus. The abstractive algorithms require training. To robustly evaluate their performance, the two abstractive algorithms are run twice, once using a model pre-trained on the CNN/DM data set and once on another training set. The second data set allows their performance on completely unseen data to be evaluated. BART1 and Pegasus1 are trained on the CNN/DM data. On the other hand, BART2 is trained using the *BART-based* data, which is basic English language with no fine-tuning [12]. Pegasus2 is trained using the *wiki-how* data set, which is a basic English language article and instructions data set [13].

LexRank is an unsupervised graph-based approach for automatic text summarisation. The paper that introduces it views the extraction of sentences as trying to find the most central ones that give a sufficient amount of information [14]. The articles went through tokenising and preprocessing before being passed to the algorithm. The number of sentences required in the summary was set to 2, after analysing the human-generated summaries in the CNN/DM data set.

Latent Semantic Analysis (LSA) is based on linear algebra method which extracts hidden semantic structures of words and sentences (i.e. it extracts the features that cannot be directly mentioned). These features are essential to data, but are not original features of the data set. It is an unsupervised approach which is implemented in a similar way to LexRank.

BART is trained by firstly corrupting text using an arbitrary noising function, and then learning a model to reconstruct the original text. It is fined tuned to a standard sequence-to-sequence model [7]. In the two BART runs, the summarisers are first initialised using the models required. The body of each article is then passed into these summarisers as well as a minimum and a maximum length for the summary, these were set to 30 and 130 words respectively. This was done after testing the length of the summaries yielded. The final parameter passed relates to early stopping, which is used to avoid over-fitting as a way of regularising the summaries. In effect, this stops the beam search when a certain threshold of sentences is finished per batch.

Pegasus was developed by Google AI in 2020. It uses an encoder-decoder model for sequence-to-sequence learning [15]. Pegasus avoids the naive approach of training like an extractive summariser and makes full use of the large training corpus. It masks whole sentences and concatenate gap-sentences into a pseudo-summary which enhances it as an abstractive algorithm [15].

4.2 Web Application

A new user interface was developed where the key objectives are: 1) to enable users to compare the performance of summarisation algorithms; 2) to allow users to select the topic of the article to be summarised; 3) to give users an overview of the algorithms and the automatic evaluation techniques to help them to understand what the results mean; 4) to provide a tool to conduct user studies to compare the performance of the summarisation algorithms for the purpose of this project. The application was developed in Python, using the Flask library.

Each of these design goals has a corresponding subset of stakeholders. The web application will be used by individuals who would like to find an article with multiple summaries for various uses such as find out which summarisation algorithm to use in their own projects, users who are conducting deep research into the summarisation algorithms and would like to see their overall performance before digging deeper into them using the links provided as a starting point, and researchers who would like to conduct user studies.

With regards to researchers who might conduct user studies, in the web interface they can choose if they want articles from CNN or DM, and the topic of those articles. Once this is selected, the user is navigated to a page with five articles and six summaries per article corresponding to the algorithms chosen at random. The first design consideration here was to remove the automatic evaluation metrics to avoid any bias in the user study. Otherwise, participants in a study might tend to rate the summaries that have a high metric score better. Such user studies usually require a few participants to rate the same article, therefore it is a bad idea to use the entire data set of hundreds of articles as that makes it unlikely that two participants rate the same article. Instead, a small pool of 20 articles per subsection was created. To avoid the risk of users learning a bias towards certain algorithms, the summaries were labelled with letters *A* to *F* (instead of algorithm names) and the order of presentation was randomised.

5 Evaluation

Evaluating the usefulness of the summary of an article can be a very subjective and challenging task. To this day there is no single perfect measure [16]. We now evaluate how close an algorithm-generated summary is to a human one, as well as its coherence, relevance and content.

5.1 Evaluation Metrics

There are the two key metrics that have been used to automatically evaluate the summaries generated in this article.

The **ROUGE** measure has been used widely in many studies that evaluate text summarisation performance [3]. It is a recall-oriented evaluation that calculates the lexical overlap between the output generated by the system and some test summary [5]. These test summaries may create a bias as they act as a baseline. Two common variations of the ROUGE score have been used:

- **ROUGE-1** looks for overlap of single word uni-grams when comparing the algorithm’s summary with the human-written reference summary. This is considered as a baseline.
- **ROUGE-2** looks for the overlap of two-worded bi-grams when comparing the algorithm’s summary with the reference summary. This is a more strict measure than ROUGE-1 and will generally yield lower scores. The reason it was chosen is to assess the fluency of the summaries.

The other metric considered is the **BLEU** score (Bilingual Evaluation Understudy Score) and it measures the precision, which is how many words in the algorithm-generated summary also appear in the reference summary. A perfect match results in a score of 1.0. Note that BLEU also computes a brevity penalty [2]. The two variations selected:

- **BLEU-1:** This refers to the implementation that counts matching uni-grams where each uni-gram is a word and acts as a baseline.
- **BLEU-2:** Unigrams up to 4-grams are assigned different weights as follows (0.6, 0.25, 0.1, 0.05). This weight distribution was chosen to allow the score to reflect the overlapping of the summaries by giving the uni-gram the highest weight and also allowing it to account for overall fluency by giving descending weights to the following n-grams.

Basing the evaluation of a summary on a frequency-based metric can bias the score against abstractive techniques, since they change the words used to enhance the flow of the summary [17]. In addition, we are likely to want sports news summaries to be engaging, which might be difficult for an automated scoring system to assess. Hence, human evaluation will be used too.

5.2 Human Evaluation

Another type of intrinsic assessment is text quality evaluation. This can consider aspects of summaries ranging from the quality of grammar to the level of content redundancy. Automated techniques still struggle to measure such human-subjective features [5, 17]. After analysing the different attributes that have been considered in previous experiments, a list of attributes was compiled: Fluency, Relevance, Duplication, Inclusion, and Exclusion. The first two measures are best suited for a rating approach, while the latter three for an error-spotting approach. Thus the proposed score will be split into two parts: (i) a Coherence score which covers fluency and relevance; (ii) an Error score which covers the other attributes.

The web application described above used to conduct our human evaluation. A pilot experiment was first run to fine tune the evaluation process. This allowed us to identify an issue where the pilot participant started building a bias towards a certain letter choice (i.e., a particular algorithm). This was dealt with by randomising the algorithms’ order and the letter mapping. The participant complained about the length of some of the articles. There is a trade-off between the variety of article lengths and participants’ attention span. To counteract this,

Table 1. Explanation of the weights and how they contribute to the overall score.

Attribute	Contribution- overall score	Reasoning
Fluency	15%	This may create a bias towards abstractive algorithms
Relevance	35%	Summary must contain the key points
Duplication	15%	Not optimal, but overall meaning of the summary holds
Inclusion	15%	Adding unimportant sentences defies summarisation
Exclusion	20%	Excluding important sentences defies summarisation

only articles that have 225 words or less were displayed to the participants to maintain their attention span after analysing the data set. In the pilot study the participant was asked to not only identify the number of errors but also to comment on their syntactic type. This was not effective as the summaries are short and do not have much variety for such detail. Comments on errors types are not requested in the final experiment.

The full user experiment included 10 participants, each assigned 10 sports articles split between the CNN and DM article sets. For each article, users had to rate the six summaries. This included a rating out of 10 in the first two measures and identification of errors for the latter three. Participants included an equal amount of student men and women, with varying proficiencies of English.

Score Calculation. Each evaluation attribute contributes a different percentage towards the overall score. The reasons for the different relative weights are listed in Table 1.

Limitations. Only shorter articles from the CNN/DM data set were used, which limits the experiment somewhat, as it does not allow a large variation window in article length. In addition, a user’s performance could change over time, which causes their judgement to vary due to either a better understanding of the task or due to fatigue.

6 Results

The sports articles were passed to the six summarisation algorithms. The resulting summaries were passed to the four automatic evaluation measures. The results can be seen in Table 2. The human evaluation was conducted on 20 sports articles, 10 from CNN and 10 from DM. The experiment had 10 participants as explained in Sect. 5.2. The results of this experiment are reported in Table 3.

6.1 Analysis

A range of findings can be inferred from these results. To test the significance of these findings, a paired *t*-test is used. It is a parametric procedure hence it makes several assumptions, the main one being the data having approximately a normal

distribution. To check this, the *Shapiro-Wilk* test was used. The significance level used is 0.05 which indicates a 5% risk of concluding that a difference exists when there is no actual difference. Overall, human evaluation metrics are on average significantly higher than the automatic metrics. This can be due to human-linked biases. Automatic metrics that take into consideration 1-gram overlaps score higher than those that take into account more. This is because it is more likely for one word to match the reference summary than a sequence of words.

Extractive vs. Abstractive. From the results for automated measures, it is apparent that abstractive techniques outperform extractive in the domain of sports articles. However, in the human evaluation metrics, there is a slight advantage for the extractive techniques. This was an unexpected result. The reason was hypothesised to be the two abstractive models that were not pre-trained on the CNN/DM data set. They brought the average down and made a significant difference only in human evaluation due to the small range that the participants rated within. Further analysis needed to be done. A one-tailed paired t-test was conducted. There was a significant difference in the performance of extractive algorithms ($M = 0.364$, $SD = 0.304$) compared to abstractive algorithms ($M = 0.433$, $SD = 0.257$), $t(5) = -2.677$, $p = 0.022$. This means that abstractive algorithms have indeed performed better. The one-tailed test is more appropriate than two-tailed in this situation since the hypothesis we are interested in is determining if extractive algorithms perform worse. Another t-test was conducted on the data this time excluding abstractive techniques that were not trained on CNN/DM. The test showed an even more significant difference.

CNN vs. DM. The CNN and DM article sets are written in American and British English respectively. Overall, the writing dialect of sports articles, whether American or British, does not affect the performance of the algorithms.

Table 2. Summary of performance for automatic evaluation metrics.

Algorithm	ROUGE1	ROUGE2	BLEU1	BLEU2
Lexrank (CNN)	0.2501	0.0847	0.2130	0.0960
LSA (CNN)	0.2062	0.0652	0.1825	0.0738
BART1 (CNN)	0.4546	0.2274	0.3476	0.2099
BART2 (CNN)	0.3496	0.1495	0.2244	0.1217
Pegasus1 (CNN)	0.5642	0.3530	0.4291	0.3101
Pegasus2 (CNN)	0.2604	0.0955	0.1692	0.0780
Lexrank (DM)	0.2727	0.1029	0.2237	0.1087
LSA (DM)	0.2244	0.0796	0.1862	0.0826
BART1 (DM)	0.4404	0.2211	0.3291	0.2055
BART2 (DM)	0.3590	0.1580	0.2598	0.1447
Pegasus1 (DM)	0.4873	0.2732	0.3443	0.2332
Pegasus2 (DM)	0.2286	0.0822	0.1344	0.0639

Table 3. Summary of performance for human evaluation. (A High score indicates that the algorithm performed well)

Algorithm	Coherence	Error	Overall_Human
Lexrank (CNN)	0.7324	0.8320	0.7813
LSA (CNN)	0.7034	0.8196	0.7615
BART1 (CNN)	0.7922	0.8564	0.8261
BART2 (CNN)	0.7326	0.8140	0.7733
Pegasus1 (CNN)	0.7712	0.8624	0.8168
Pegasus2 (CNN)	0.6388	0.7658	0.7023
Lexrank (DM)	0.7682	0.8352	0.8017
LSA (DM)	0.7398	0.8430	0.7914
BART1 (DM)	0.7567	0.8602	0.8079
BART2 (DM)	0.7352	0.8164	0.7758
Pegasus1 (DM)	0.7760	0.8706	0.8233
Pegasus2 (DM)	0.6518	0.7756	0.7137

However, the CNN summaries seem to outperform DM summaries in the automatic metrics while the opposite is true for human metrics. This is likely due to a limitation in the human evaluation as the participants are based in Ireland which likely shares more similar language usage to the UK than to the US. The t-test failed to reject the null hypothesis, that is the mean performance on CNN and DM is not significantly different.

Best Performing Algorithms. The algorithm that scores the best in each evaluation measure has certain desirable features. Pegasus1 scored the best in six evaluation measures, it is the pre-trained on the CNN/DM data set. Pegasus1 is the most effective summarisation algorithm for sports articles, followed by BART1. The latter loses out in the automatic evaluation metrics. This may be due to BART1 rephrasing the summary while keeping the same meaning. This is a limitation of such metrics. LSA is the worst performing algorithm in terms of ROUGE1 and ROUGE2 scores. This signifies that LSA summaries have poor recall. However, the worst performer in the remaining evaluation measures is Pegasus2. It is the same algorithm as Pegasus1 but is pre-trained on the wiki-how data set. Pegasus2 demonstrated low precision and coherence, with many errors. In general, extractive algorithms performed badly alongside abstractive algorithms that were trained using inappropriate data sets.

Automatic vs Human Evaluation. Human evaluation results are similar across the six algorithms. Humans tend to not be extreme and therefore give scores with small variations. The other reason may be the lack of diversity in the data set used in the human evaluation. The metric evaluation scores are highly correlated with one another as they all measure overlap (see Table 4). However, the human and automatic evaluation metrics also have a strong pos-

itive correlation, suggesting a good level of agreement with human judgement. The human Error score seems to correlate a little higher with all four automated metrics compared with Coherence score. This may be due to the latter being more subjective. Similar results were obtained when using Spearman's rank correlation. From this, it is concluded that human and automatic evaluation measures correlate quite strongly, at least for the domains that we have studied.

Table 4. Correlation between measures.

	Rouge1	Rouge2	Bleu1	Bleu2	Coherence	Error	Avg_Human
Rouge1	1.0000	0.9901	0.9653	0.9821	0.6597	0.6718	0.6773
Rouge2	0.9901	1.0000	0.9625	0.9906	0.6298	0.6704	0.6583
Bleu1	0.9653	0.9625	1.0000	0.9886	0.7727	0.7974	0.7967
Bleu2	0.9821	0.9906	0.9886	1.0000	0.7005	0.7424	0.7303
Coherence	0.6597	0.6298	0.7727	0.7005	1.0000	0.9349	0.9899
Error	0.6718	0.6704	0.7974	0.7424	0.9349	1.0000	0.9756
Avg_Human	0.6773	0.6583	0.7967	0.7303	0.9899	0.9756	1.0000

We observe that the automatic and human evaluation scores have high correlations, but statistically different means. A two-tailed paired t-test showed that there is a significant difference in the mean of the automatic summarisation metrics ($M = 0.224$, $SD = 0.090$) compared to human summarisation metrics ($M = 0.781$, $SD = 0.038$), $t(11) = -27.126$, $p = 0.000$. On average, human metrics result in higher scores than automatic evaluation. The standard deviation of the measures tend to be similar. Taking this into account, in addition to the correlation results, it appears that the automatic metrics correlate to human judgement in a relatively accurate manner.

6.2 Domain Dependence

One of the goals of this work is to establish if summarisation algorithms perform differently depending on the topic of the text. To do this, the algorithms were also run on political articles from the CNN/DM data set, and then evaluated using the automatic evaluation metrics. In general, it seems that the algorithms performed worse on the politics articles compared to sports. The gap is larger with abstractive algorithms. A two-tailed paired t-test showed that there is a significant difference in the performance of the summarisation algorithms on sports articles ($M = 0.224$, $SD = 0.088$) compared to politics articles ($M = 0.189$, $SD = 0.072$), $t(5) = 4.359$, $p = 0.007$.

The correlation between each of the six summarisation algorithms' mean evaluation score for politics articles and sports articles was calculated. All six algorithms were highly correlated according to both Pearson's and Spearman's measures as seen in Table 5. This leads to the conclusion that a similar performance pattern for the summarisation algorithms exists in both domains.

The results discussed here lead to the conclusion that the order of algorithms does not depend on the domain, as the best and worst performing algorithms were the same for sports and politics articles. However, there is a significant difference in the actual performance of each algorithm. In general, the algorithms performed considerably better with sports articles according to the evaluation measures used. The abstractive algorithms that were trained on the CNN/DM data set showed the biggest gap in performance between the two domains. This could be due more sports than politics articles being available for training.

Table 5. Correlation between sports and politics articles.

Algorithm	Pearson	Spearman
Lexrank	0.998887	1.0
LSA	0.999665	1.0
BART1	0.993963	0.8
BART2	0.994381	1.0
Pegasus1	0.995575	1.0
Pegasus2	0.996562	1.0

7 Conclusions and Future Work

In the experiments conducted in this paper, the two abstractive algorithms performed better than the two extractive ones, not only with regards to coherence as expected, but also in terms of the automatic evaluation metrics. This was the case in both the sports and politics news domains. It is worth noting that, even if the classification of algorithms is domain-independent, their performance is domain-dependent. That being said, the performance of abstractive algorithms is dependent on the data used in their pre-training. The results of the evaluation metrics in this paper agree with the results from [5]. The only exception here is that, when the abstractive algorithms are pre-trained using an inappropriate training set, then they perform more poorly than their extractive counterparts.

The incorporation of human evaluation allowed us to further understand the effectiveness of the algorithms considered. It was concluded that, due to the high correlation between human evaluation metrics and automatic ones, automatic metrics such as ROUGE and BLEU can provide a good estimation of human judgement. It was also found that the algorithms performed similarly on CNN and DM article sets. However, human evaluation was closer on DM articles by a small margin, possibly due to a user bias in relation to British English. The web application developed as part of this work acts as the first user interface that allows the comparison of sample summaries generated by different summarisation algorithms across different subject domains. The application also enables human evaluation experiments to be conducted, providing a useful tool for researchers working in this area.

This study would benefit from expanding the number of algorithms evaluated, including a larger set of extractive and abstractive techniques. The human evaluation could be improved by adding more participants. In addition, a larger pool of articles would make the results more robust. More detailed aspects can be added to the human evaluation – e.g. by collecting the syntactic types and the severity level of the errors appearing in summaries. This paper was an introduction into testing the generalisation of summarisation algorithms. This could be expanded by looking at the performance of algorithms for more domains, such as travel articles, or by performing within-domain comparisons, such as sports news articles covering football versus basketball.

References

1. Saggion, H., Poibeau, T.: Multi-source, Multilingual Information Extraction and Summarization, pp. 3–13. Springer (2012). <https://doi.org/10.1007/978-3-642-28569-1>
2. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the ACL (2002)
3. Lin, C.-Y.: ROUGE: a package for automatic evaluation of summaries. In: Proceedings of ACL Workshop on Text Summarization Branches Out, vol. 10 (2004)
4. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. **2**, 159–165 (1958)
5. Huang, D., et al.: What have we achieved on text summarization? (2020)
6. Mihalcea, R., Tarau, P.: TextRank: bringing order into texts (2004)
7. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension (2020)
8. Huang, K.-H., Li, C., Chang, K.-W.: Generating sports news from live commentary: a Chinese dataset for sports game summarization (2020)
9. Hermann, K.M., et al.: Teaching machines to read and comprehend. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
10. Sumy. <https://pypi.org/project/sumy/>
11. Transformers. <https://huggingface.co/docs/transformers/index>
12. Facebook/Bart-base. <https://huggingface.co/facebook/bart-base#>
13. Google/Pegasus-Wikihow. <https://huggingface.co/google/>
14. Erkan, G., Radev, D.R.: LexRank: graph-based lexical centrality as salience in text summarization. CoRR [arXiv: 1109.2128](https://arxiv.org/abs/1109.2128) (2011)
15. Zhang, J., Zhao, Y., Saleh, M., Liu, P.J.: PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. CoRR (2019)
16. Lloret, E., Plaza, L., Aker, A.: The challenging task of summary evaluation: an overview (2018)
17. Grusky, M., Naaman, M., Artzi, Y.: Newsroom: a dataset of 1.3 million summaries with diverse extractive strategies. CoRR (2018)


Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Automatic Linking of Podcast Segments to Topically Related Webpages

Carla McKeon¹, Claudio Rocha¹, and Gareth J. F. Jones^{1,2} 

¹ School of Computing, Dublin City University, Dublin, Ireland
{carla.mckeon32,claudio.rocha2}@mail.dcu.ie, gareth.jones@dcu.ie

² ADAPT Centre, Dublin City University, Dublin, Ireland

Abstract. Podcasts are becoming an increasingly popular source of information. However, they often rely on the topical knowledge of the listener in order for them to be fully understood. We describe an investigation into methods to augment the contents of podcasts with related information from the Web. We seek to identify webpages related to segments within a podcast. NLP techniques are used to analyze audio podcast transcripts and link these to related content. We propose and examine 10 methods for automatically generating search queries from transcript segments, which are then used to search for related content on the web. The relevance of retrieved webpages to retrieved content is evaluated using crowdsourcing via Amazon Mechanical Turk. Extracting key phrases directly from the podcasts using YAKE was the most successful approach with more than 90% returned pages assessed as relevant, with precision at rank 1 and rank 3 above 0.9.

Keywords: Automatic content linking · Key phrase extraction · Podcast summarization · Automatic query construction

1 Introduction

Podcasts are an increasingly popular form of audio media providing information, topical comment and entertainment to ever growing numbers of people. A podcast episode may cover multiple topics or themes over the course of its content. The full meaning of the topics discussed may not though be apparent to the listener if they do not have a reasonable background in the issue under discussion. In this situation the listener may turn to a web search engine, such as *Google* to seek further information about the topic in order to better understand the podcast. In this paper we investigate the development of a method to automatically link segments of podcast content to related webpages. These links would then be available to podcast listeners removing the need for them to carry out their own search if they wish to find further information about the content.

In order to identify content related to a podcast, a suitable query must first be created from the words spoken in the podcast. In our study we explore the use of number of NLP techniques to construct an effective query [1]. To retrieve

webpages using these queries, we make use of the Google Search API. This returns a ranked list of webpages from the Google Search engine. To assess the relevance of the returned webpages to the segments of the podcast transcript used to construct the queries, we use crowdsourcing via Amazon Mechanical Turk (MTurk) [2, 3]. Each online assessor judges relevance of the retrieved webpage for the segment on a scale between fully relevant and not relevant. These judgements are then used to evaluate the effectiveness of each query generation technique using standard precision based evaluation metrics. For this investigation we make use of a large collection of podcasts with transcripts made available for research purposes by Spotify [4].

This paper is structured as follows. The next section reviews existing work in podcast search, keyphrase extraction and content linking. Section 3 describes the creation of the experimental dataset used in our study, Sect. 4 introduces our automated query generation methods, Sect. 6 outlines our evaluation processes and the metrics used to evaluate the effectiveness of our automated linking methods, Sect. 7 gives experimental results and analysis, and Sect. 8 concludes the paper and makes suggestions for further work.

2 Related Work

In this section, we review relevant work in podcast search and summarization, information retrieval using automated query generation, and the development of test collections for the evaluation of information retrieval.

2.1 Podcast Search and Summarization

Until recently there has been limited related research reported on the automated processing of podcasts for search and summarization. To encourage more work in this area Spotify released a large collection of podcasts with corresponding transcripts created using automatic speech recognition (ASR) [4]. This collection contains on the order of 100,000 podcasts, and formed the basis of benchmark podcast tasks at the Text Retrieval Conference (TREC)¹ in 2020 and 2021. These tasks evaluated methods for the effective search of podcasts in response to user search queries and summarization of the podcast transcripts [5, 6]. Recognising that podcasts are long and multi-topic, the search task focused on retrieval of relevant podcast segments using the ASR transcripts for a set of topical search queries. Participants were provided with 58 search queries describing user information needs. The relevance of segments to each query was manually judged by assessors at NIST. Segments identified as relevant within this task form the starting point for our investigation of linking segments to related webpages.

Summarization methods are concerned with creation of concise shortened documents preserve the most salient information [7, 8]. For our query generation processes, we begin by summarising podcasts segments to capture their key information and then extract queries containing this key information.

¹ <https://trecpodcasts.github.io/>.

2.2 Information Retrieval Through Key Phrase Extraction and Query Generation

The Retrieval from Conversational Dialogues (RCD) track at the FIRE 2020² conference explored the utility of information retrieval methods to retrieve more information about topics discussed in transcripts of interactive movie dialogues. The conversational structure of these movies is similar to that of many podcasts. Participants in the RCD task were required to retrieve a ranked list of potentially related documents from Wikipedia for a span of text [9]. The participation requirements for the RCD 2020 task [10] adopted a similar strategy to the one which we follow here. The main topics of discussion in movie dialogues needed to be identified using summarization methods. Key phrase extraction techniques were then used to create queries to search for related articles in Wikipedia. The first approach used different methods, mixing different techniques for summarization such as BERT and different key phrase extraction techniques such as YAKE and TextRank. Those models used a phrase frequency of 1–3 words. When comparing participant results, the method that obtained the best results was the one using a custom summary extraction method and the TextRank technique. For the second approach, only one key phrase extraction technique, TextRank, was tested, which was applied directly to the movie dialogue. After which the key topics extracted were again used as queries to retrieve relevant documents from the Wikipedia dataset.

There has been limited previous work on automated link creation. Probably the most closely related task to our work is the NTCIR-9 Crosslink task, which examined cross-lingual linking between Wikipedia documents [11].

2.3 Information Retrieval Test Collections

A standard test collection for the evaluation of information retrieval methods consists of a set of documents, a set of queries and the identities of the relevant documents from the collection for each query. The relevance of each document to the query must be judged manually since the contents of the query do not fully express the information need that it seeks to express. The relevance of each document to the query can either be judged in a binary manner as “relevant” or “non-relevant”, or using a graded scale, where documents can be assessed between highly, partially, marginally or not relevant. A number of studies have been carried out examining the design of effective and reliable test collections for information retrieval. A key result for our study is that at least 25–50 queries are required for results to be reliable [12]. For our investigation we thus sought to examine the effectiveness of link generation for at least 30 podcast fragments from which we generated queries for automated search.

² <https://rcd2020firedtask.github.io/RCD2020FIRETASK/>.

3 Experimental Dataset

In this section we introduce the Spotify podcast collection which forms the basis of our study, selection of the data from this collection used for our experimental investigation and preprocessing procedures applied to the selected data.

3.1 Podcast Dataset

As outlined earlier, we used the podcast dataset made available by Spotify for this investigation. This collection consists of 105,360 podcast episodes taken from 18,376 shows collected between January 2019 and March 2020. [4]. The podcast dataset can be requested from Spotify for research purposes³.

The collection includes recordings of the original podcasts with an ASR transcript created using a standard online Google transcription service. The podcasts are all in English and vary in length, being on average 30 min long, with the shortest running for 10.5 s and the longest for over 5 h.

For its use in the TREC Podcasts tasks [5, 6], the podcast collection was supplemented with 58 Topic statements intended to be representative of the sort of queries that a listener might issue to a podcast search engine. Since podcasts are long and listening to them to identify content relevant to such a query would be time consuming, the TREC search tasks focused on identifying relevant segments created from the podcast transcripts. Segments were created by dividing the transcripts 120 s pieces with a 50% overlap to ensure the presence of segments where topical content is contained with single segments, rather than being split between adjacent segments.

3.2 Content Selection

The TREC Podcast task defined highly relevant segments as being ideal entry points into the podcast for a listener, and as being fully on topic.

For our investigation we wished to focus on a set of segments which have significant topical interest for a defined topic. We thus decided to begin by selecting the segments from the TREC Podcast task assigned a relevance score of 4 for one of the search topics. This resulted in a set of 55 segments associated with 18 of the topics.

Since errors in the transcripts may impact on content analysis processes that will be used for the query construction stage, we further filtered these segments to remove those containing obvious ASR transcription errors. This filtering resulted in segments associated with all 18 topics remaining. Mindful of the cost of relevance assessment using crowdsourcing and the minimum number of queries required for reliable experimentation [12], we randomly selected 30 segments of the remaining segments for use in our investigation.

In an operational setting all segments in the podcast archive would be linked to related web content during the indexing setup of the podcast search engine in

³ <https://podcastsdataset.byspotify.com>.

Table 1. Methods to extract queries

Method	Summ.	Keyword E.	Keyword G.
M1	BERT	YAKE	–
M2	BERT	Text Rank	–
M3	BERT	–	KeyBART
M4	BERT	–	T5-s.-OpenKP
M5	T5-P.Sum.	YAKE	–
M6	T5-P.Sum.	Text Rank	–
M7	T5-P.Sum.	–	KeyBART
M8	T5-P.Sum.	–	T5-S.-OpenKP
M9	–	YAKE	–
M10	–	Text Rank	–

advance of listeners entering search queries. This would of course include query construction with errorful podcast transcripts. Examining the impact of these transcription errors on the effectiveness of the linking process will be the focus on further experimental studies.

3.3 Data Preprocessing

In order for the transcripts to be used consistently in the construction of queries, they were preprocessed into a standard format and standard NLP analysis applied. Preprocssing involved tokenising the raw text into words. and converting each word to lower case to ensure that the system avoids interpreting the same word (e.g. “Book” and “book”) as two different words. Then converting the word into its base form using lemmatization, and combining this with part-of-speech (POS) tagging to give context to each token and to bring together different word forms (e.g. meeting - meet (core-word extraction), was - be (tense conversion to present), mice - mouse (plural to singular)).

4 Query Generation

In order to identify potentially relevant or interesting web content to link to podcast segments, a query representing the key content of the segment must be created for submission to a search engine to seek this content. The core element of our investigation is the proposal and evaluation of a number of methods for query generation from podcast transcripts.

Our methods can be grouped into three approaches: summarization followed by key phrase extraction, summarization followed by key phrase generation and key phrase extraction directly from the podcast segment. Overall 10 different query generation methods were examined, the details of these methods are summarized in Table 1. These methods are described in outline in the following subsections.

4.1 Summarization Methods

Since podcasts often contain colloquial, conversational, and noisy commercials and sponsorship segments, we examined a preprocessing step of summarization to reduce the transcript to its core content [13]. We investigated whether generating queries from segment summaries improves likelihood of relevance of the retrieved webpages. In this paper we examine two different summarization methods:

BERT. This is a pre-trained BERT model developed by Google’s AI Language team and described in [14]. BERT has achieved groundbreaking results on various NLP tasks. In this paper, we use BERT for extractive summarization using the PyTorch transformers library (HuggingFace) to perform extractive summarization using the following process [15]: sentence embedding, running a clustering Algorithm (K-means), and locating the sentences near the cluster’s centroid.

T5. The T5-podcast summarizer model is the result of fine-tuning t5-base [16] on the Spotify podcasts dataset [4]. This model is based on Google’s T5, a text-to-text framework, pre-trained on the C4 Dataset (Colossal Clean Crawled Corpus). The main concept of this model is that the input and output will always be strings, while the output of the BERT model is a class label or span of the input.

4.2 Key Phrase Extraction Methods

The main objective of key phrase extraction in query generation is to capture the main topic discussed in the podcast, including the key events reported, the entities involved in these events and, their outcome, impact and significance. The following constraints [17] were considered while fine-tuning the parameters in the key phrase extraction techniques:

- A key phrase can be a single word or a sequence of up to four consecutive words as they appear in the podcast
- A minimum of one and a maximum of three key phrases were selected
- A key phrase has to be either a noun phrase, verb, proper name or adjective

Extensive research surveys of key phrase extraction methods and comparison of their relative performance are provided in [18,19]. We implemented both an unsupervised statistical and a graph based approach as outlined below.

YAKE! Yet Another Keyword Extraction (YAKE) [20] is a statistical method which exploits frequency and positional information of each candidate key phrase (word n-grams not containing punctuation marks, nor starting or ending with a stop word).

Terms are scored by gathering all of the feature weights into a unique score as shown in Eq. 1.

$$Score(t) = \frac{T_{Rel} * T_{Position}}{T_{Rel} + \frac{TFreq_{Norm} + T_{Sentence}}{T_{Rel}}} \quad (1)$$

where T_{Rel} is a score given to the relatedness of the term in the context of the document, which downgrades terms that co-occur with unique terms in a given window size. Dividing both $TFreq_{Norm} + T_{Sentence}$ by T_{Rel} gives a higher value to terms that appear more frequently and in many sentences. The level of importance of a term is higher as long as it is relevant, i.e. has a low T_{Rel} and achieves a high score on $TFreq_{Norm} + T_{Sentence}$ score. $T_{Rel} * T_{Position}$ takes the position of the term in the sentence into account multiplying it by the term's relevance to context score.

The score for a candidate key phrase $k = t_1 t_2 \dots t_n$ is then computed as shown in Eq. 2.

$$Score(k) = \frac{\prod_{i=1}^{\infty} Score(t_i)}{frequency(k) * (1 + \prod_{i=1}^{\infty} Score(t_i))} \quad (2)$$

We use the LIAAD⁴ version of YAKE with parameters tuned as follows: setting a window size of 2, which is used to capture the context of the phrases, and threshold to 0.8 which helps to eliminate redundant queries using the concept of Levenshtein distance. The top three key phrases were extracted as representing the core content of the segment for construction of the query.

TextRank. TextRank [21] is a graph based extraction method where documents are modeled with weighting co-occurrence networks using a co-occurrence window of variable sizes (2–10). $TR(V_i)$ represents the TextRank score of the point V_i calculated as shown in Eq. 3.

$$TR(V_i) = (1 - d) + d \sum_{V_j \in In(V_i)} \frac{W_{ji}}{\sum_{V_k \in Out(V_j)} W_{jk}} TR(V_j) \quad (3)$$

Similar to the concept of PageRank [22], d is the probability of the phrase occurring at random in the document, and is between 0 and 1.

4.3 Key Phrase Generation

Key phrase generation differs from key phrase extraction. The former are models trained to learn the mapping between a pair of texts and generate new key phrase text, while the latter extracts the most relevant words from a given input.

BART [24] & T5 [16] are two of the most successful generation transformers developed to date. They can be fine tuned for text-to-text-generation problems such as our task of key phrase generation.

⁴ <https://github.com/LIAAD/yake>.

KeyBART. A generative model for text generation that reproduces key phrases with connection to the input in the CatSeq format. The internal architecture of BART is built on a transformer encoder-decoder (seq2seq) model with a bidirectional encoder, similar to BERT and an autoregressive decoder. The pre-training of BART takes two stages: 1) corrupting text with an arbitrary noising function, and 2) learning a model to reconstruct the original text.

T5-small-OpenK. This model is a key phrase generation technique based on the T5-small model and fine-tuned using the dataset OpenKP. This generation transformer model is tuned as text to text to generate keywords, with the limitation of only working on documents using the English Language. This model has approximately 220 million parameters, which is approximately twice the number of parameters as BERT. The model was pre-trained on a different mixture of supervised and unsupervised tasks.

5 Webpage Retrieval

Once the query has been constructed for a podcast segment, the next stage is to use it to search for relevant webpages. For this task we make use of the Google Custom Search Engine API [23]. This is a restful API that retrieves results of a search query in a JSON object, with three types of data: search results, metadata containing information about the requested search and metadata containing information about Programmable Search Engine. Every search retrieves a maximum of 10 results per query. We extracted the search ranking, URL link and title of the page obtained for each query.

It should be noted that multiple calls to the Google API are not guaranteed to return the same documents. However, since the calls are made in quick succession, the features of the search engine are unlikely to change between calls.

6 Evaluating of Our Information Retrieval System

In this section we describe our method for assessing relevance of retrieved items, and the evaluation metrics used in our investigation.

6.1 Relevance Evaluation

To assess the relevance of webpages we used crowdsourcing with Amazon Mechanical Turk (MTurk)⁵. This service provides a platform providing access to online human workers how can complete assigned tasks referred to as Human Intelligence Tasks (HITS). Online workers receive payment for successfully completed HITS. For our assessment of linking using our query generation methods, we form a pool of retrieved items for relevance assessment for each podcast segment in our test dataset. The pool was formed by selecting the top 3 ranked

⁵ <https://mturk.com/>.

documents retrieved for each query, merging these into a pool of unique document entries, and then requesting MTurk workers to assess the relevance of each retrieved document in the pool. Relevance of each item was assessed in terms of its relation to the segment used to create the queries which retrieved the item. The worker was shown the transcript and the contents of each retrieved item in the pool one after the other. Workers were required to perform the assessment on a graded scale (Excellent(3), Good (2), Fair (1), Bad (0)).

Since workers may not carry out the tasks correctly, we implemented a number of quality control measures. We estimate that each HIT should take 3–5 min, therefore any HITs completed in under 45s were rejected without payment, and the HIT re-submitted to MTurk. Similarly, we rejected workers who returned an identical answer every time. Additionally, in order to help ensure that we recruited reliable workers, we required that workers had an approval rating of at least 80% for their previous completed HITs on MTurk.

Since relevance judgements are subjective, we gathered more than one judgement of each podcast-segment webpage pair to improve reliability of the relevance scores. To do this, we assigned each HIT to three AMT workers, meaning that 3 rounds of assessments are recorded for each HIT. We took the average of the three judgements as the agreed relevance level for each assessed document.

6.2 Evaluation Metrics

Information retrieval is typically evaluated based on the top k documents returned for a query. We use three metrics to evaluate the effectiveness of our content linking retrieval method [25,26]:

P@k: Precision at K. Quantifies how many items in the top-k results are relevant, i.e. out of the returned list how many were relevant. TP is True Positives and FP is False Positives:

$$Precision@k = \frac{TP@k}{(TP@k) + (FP@k)} \quad (4)$$

MAP: Mean Average Precision. Calculates the average precision across multiple queries, with Q being the number of queries and $AP(q)$ the average precision for query q :

$$MAP = \frac{1}{Q} \sum_{q=1}^Q AP(q) \quad (5)$$

MRR: Mean Reciprocal Rank. This is a measure of the rank position of the highest ranked relevant document, i.e. how far down the list we have to go to find a relevant document, with Q being the total number of queries, and $rank_i$ the rank of the first relevant result:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (6)$$

Table 2. Results - relaxed relevance

Method	P@1	P@3	MAP	MRR	nDCG
M1	0.767	0.644	0.811	0.800	0.967
M2	0.767	0.744	0.811	0.797	0.948
M3	0.733	0.744	0.822	0.822	0.922
M4	0.700	0.700	0.789	0.789	0.930
M5	0.800	0.789	0.856	0.847	0.923
M6	0.7000	0.711	0.767	0.769	0.950
M7	0.800	0.711	0.844	0.840	0.950
M8	0.833	0.822	0.900	0.886	0.942
M9	0.933	0.911	0.967	0.953	0.953
M10	0.800	0.844	0.900	0.892	0.917

Table 3. Results - stricter relevance

Method	P@1	P@3	MAP	MRR	nDCG
M1	0.367	0.333	0.417	0.422	0.967
M2	0.367	0.378	0.422	0.400	0.948
M3	0.467	0.467	0.561	0.533	0.922
M4	0.333	0.411	0.467	0.469	0.930
M5	0.367	0.478	0.533	0.542	0.923
M6	0.367	0.378	0.472	0.458	0.9500
M7	0.333	0.356	0.428	0.422	0.950
M8	0.367	0.378	0.472	0.458	0.9500
M9	0.700	0.678	0.778	0.761	0.953
M10	0.467	0.556	0.611	0.603	0.917

nDCG: Discounted Cumulative Gain. This is a sum of relevance scores for the top- n documents, normalized by penalizing each score by its position, i.e. it gives a weight to relevant documents in order of ranking, with $rel(n)$ being an indicator function which is 1 when the item at rank K is relevant

$$nDCG@k = \sum_{i=1}^n \frac{rel_i}{\log_2(i+1)} \quad (7)$$

7 Experimental Results

In this section, we present results using two different relevance levels. In the first relaxed setting, webpages assessed with relevance levels 1, 2 and 3 are all counted as relevant. In the second stricter setting, only documents assessed with relevance levels 3 & 4 are counted as relevant.

Setting 1. From Table 2, we see that for M9 at P@1 around 93% of the first retrieved webpages are relevant. M9 also obtained the best results for MAP and MRR showing that the relevant items are also found at higher positions. In relation to nDCG, the values are similar across all the methods, revealing that all have a similar pattern for the ideal order on the relevant retrieved webpages. On the other hand, M4 and M6 obtain the lowest results with a P@1 of 0.7, meaning that only 70% of the first retrieved webpages are relevant, with a value of MAP below 0.8.

Setting 2. Table 3 shows results for Setting 2, we can see that M9 is again the best performing method. The values across the different metrics are lower than Table 2, as would expected, but M9 is still able to perform relatively well compared with the other methodologies, being the only method to achieve values above 0.6 for P@1, P@3, MAP and MRR. On the other hand, M4 and M7 have the lowest results, with only around 33% of the first retrieved webpages being

relevant. It is worth noting that when analysing some queries, it is possible to verify that some techniques using summarization do not capture key ideas in the summary, and consequently cannot generate accurate queries, while the best performing method was applied directly to the raw podcast segment.

8 Conclusions and Further Work

This paper described an investigation into the automated linking of related webpages to the transcripts of 2 min segments of podcasts. We examined 10 methods of creating queries from the podcast segment transcripts, and evaluated their effectiveness for retrieving related webpages using the Google API.

In general terms, we demonstrated that it is possible to link relevant content to different podcast segments, and that applying a key phrase extraction technique directly on the raw segment obtained better results than summarizing the already short 2 min segment. For both relevance settings reported, the best performing method was able to retrieve more than 70% of relevant content in the first ranked webpage.

There are various directions for potential further work. An important area for expansion is to increase the depth of the pool of retrieved documents assessed from the top 3, to the top 10 document or potentially more. Also rather than relying on fixed 2 min length segments, automatic segmentation methods could be applied to form topically related segments, which may form the basis of better queries since they provide full coverage of the topical region in the podcast.

Acknowledgement. The contribution of Gareth Jones is partially supported by Science Foundation Ireland as part of the ADAPT Centre (Grant 13/RC/2106) at Dublin City University.

References

1. Siddiqi, S., Sharan, A.: Keyword and keyphrase extraction techniques: a literature review. *Int. J. Comput. Appl.* **109**(2), 18–23 (2015)
2. Alonso, O., Rose, D.E., Stewart, B.: Crowdsourcing for relevance evaluation. In: *ACM SigIR Forum*, vol. 42, no. 2, pp. 9–15 (2008)
3. Paolacci, G., Chandler, J., Ipeirotis, P.G.: Running experiments on amazon mechanical turk. *Judgm. Decis. Mak.* **5**(5), 411–419 (2010)
4. Clifton, A., et al.: 100,000 podcasts: a spoken English document corpus. In: *Proceedings of the 28th International Conference on Computational Linguistics* (2020)
5. Jones, R., et al.: TREC 2020 podcasts track overview. In: *Proceedings of TREC 2020*, NIST, Online (2020)
6. Karlgren, J., et al.: TREC 2021 podcasts track overview. In: *Proceedings of TREC 2021*, NIST, Online (2021)
7. Nenkova, A., McKeown, K.: Automatic summarization. *Found. Trends Inf. Retr.* **5**(2–3), 103–233 (2011)
8. Maynez, J., Narayan, S., Bohnet, B., McDonald, R.: On faithfulness and factuality in abstractive summarization. *arXiv preprint [arXiv:2005.00661](https://arxiv.org/abs/2005.00661)* (2020)

9. Ganguly, D., Pal, D., Verma, M., Sen, P.: Overview of RCD-2020, the FIRE-2020 track on retrieval from conversational dialogues. In: *Proceedings of FIRE 2020, Online* (2020)
10. Kaushik, A., Ramachandra, V.B., Jones, G.J.F.: DCU at the FIRE 2020 retrieval from conversational dialogues (RCD) task. In: *FIRE*, pp. 788–805 (2020)
11. Tang, L.-X., Geva, S., Trotman, A., Xu, Y., Itakura, K.Y.: Overview of the NTCIR-9 crosslink task: cross-lingual link discovery. In: *Proceedings of the NTCIR-9 Workshop* (2011)
12. Buckley, C., Voorhees, E.M.: Evaluating evaluation measure stability. *ACM SIGIR Forum* **51**(2), 235–242 (2017)
13. El-Kassas, W.S., Salama, C.R., Rafea, A.A., Mohamed, H.K.: Automatic text summarization: a comprehensive survey. *Expert Syst. Appl.* **165**, 113679 (2021)
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)* (2018)
15. Miller, D.: Leveraging BERT for extractive text summarization on lectures. *arXiv preprint [arXiv:1906.04165](https://arxiv.org/abs/1906.04165)* (2019)
16. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**(140), 1–67 (2020)
17. Piskorski, J., Stefanovitch, N., Jacquet, G., Podavini, A.: Exploring linguistically-lightweight keyword extraction techniques for indexing news articles in a multi-lingual set-up. In: *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pp. 35–44 (2021)
18. Papagiannopoulou, E., Tsoumakas, G.: A review of keyphrase extraction. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **10**(2), 1339 (2020)
19. Hasan, K.S., Ng, V.: Automatic keyphrase extraction: a survey of the state of the art. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1262–1273 (2014)
20. Campos, R., et al.: YAKE! Keyword extraction from single documents using multiple local features. *Inf. Sci.* **509**, 257–289 (2020)
21. Mihalcea, R., Tarau, P.: TextRank: bringing order into text. In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404–411 (2004)
22. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.* **30**(1–7), 107–117 (1998)
23. Allauddin, M., Azam, F.: Service crawling using Google custom search API. *Int. J. Comput. Appl.* **34**(7), 2011 (2011)
24. Lewis, M., et al.: Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint [arXiv:1910.13461](https://arxiv.org/abs/1910.13461)* (2019)
25. Sakai, T., Kando, N.: On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Inf. Retr.* **11**(5), 447–470 (2008)
26. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst. (TOIS)* **20**(4), 422–446 (2002)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Knowledge Representation, Reasoning, Optimisation and Intelligent Applications



A Large Neighborhood Search Approach for the Data Centre Machine Reassignment Problem

Filipe Souza^{1,2(✉)}, Diarmuid Grimes^{2,3}, and Barry O’Sullivan^{1,2}

¹ Insight SFI Research Centre for Data Analytics, University College Cork,
Cork, Ireland

`f.desouza@cs.ucc.ie`

² SFI Centre for Research Training in Artificial Intelligence, Cork, Ireland

³ Munster Technological University, Cork, Ireland

<http://www.ucc.ie/>, <http://www.crt-ai.cs.ucc.ie>, <http://www.mtu.ie/>

Abstract. One of the main challenges in data centre operations involves optimally reassigning running processes to servers in a dynamic setting such that operational performance is improved. In 2012, Google proposed the Machine Reassignment Problem in collaboration with the ROADEF/Euro challenge. A number of complex instances were generated for evaluating the submissions. This work focuses on new approaches to solve this problem.

In particular, we propose a Large Neighbourhood Search approach with a novel, domain-specific heuristic for neighborhood selection. This heuristic uses the unbalanced resource usage on the machines to select the most promising processes in each iteration. Furthermore, we compare two search strategies to optimise the sub-problems. The first one is based on the concept of Limited Discrepancy Search, albeit tailored to large scale problems; and the second approach involves the standard combination of constraint programming with random restart strategies.

An empirical evaluation on the widely studied instances from ROADEF 2012 demonstrates the effectiveness of our approach against the state-of-the-art, with new upper bounds found for three instances.

Keywords: LNS · Neighbourhood selection · Machine reassignment problem · Limited discrepancy search

1 Introduction

There has been a significant increase in data centers over the past two decades. Today there are nearly 3000 in the US alone and over 70 in Ireland¹. With the

¹ <https://www.statista.com/topics/6165/data-centers>.

Supported by SFI Centre for Research Training in Artificial Intelligence under Grant No. 18/CRT/6223 and SFI under Grant No. 12/RC/2289-P2, co-funded under the European Regional Development Fund.

© The Author(s) 2023

L. Longo and R. O’Reilly (Eds.): AICS 2022, CCIS 1662, pp. 397–408, 2023.

https://doi.org/10.1007/978-3-031-26438-2_31

increase in streaming services, and the default data acquirement of most websites, etc. this is only expected to increase. In the research community, much research has naturally focused on reducing the environmental impact of data centers, with less focus on improving operational performance within data centers.

In 2012 Google proposed a challenge with this latter issue of operation performance in mind, via the ROADEF/Euro Challenge 2012 (*Roadef12*). The general goal is the optimisation of a data centre environment for virtualisation and service configuration. In particular, they proposed the Machine Reassignment Problem (MRP), which aims to reallocate a set of processes to a set of machines in order to minimise a multi-objective function subject to a number of constraints. During the competition a range of optimisation approaches were proposed and they are summarised by Afsar et al. in [9]. Due to its complexity and specificity, the MRP has been the focus of many works in the literature in the decade since, as discussed by Canales et al. [2].

In this paper we present a Large Neighbourhood Search (LNS) with some specific components to solve the MRP. The main contributions of this work are: (i) a novel domain specific neighbourhood selection heuristic; and (ii) a novel search strategy to optimise each LNS sub-problem. With regard to the search strategy, it relies on the known issue that a heuristic decision is more likely to be wrong in the beginning of the search because it has less information than deeper in search.

Bringing this to the MRP problem, when we assign the first process to the best machine based on the heuristic, perhaps this machine is the best machine at this moment because other processes are not assigned yet. While the last process to be assigned is more likely to be assigned correctly because the heuristic has the information of all processes already assigned. As the backtracking algorithm uses depth first search, it spends a long search time relying on the first decision taken by the heuristic. To overcome this issue, we investigate two search strategies that do not spend a large amount of time investigating alternatives values for heuristics decisions that are more likely to be correct.

2 Related Work

Large neighborhood search was first proposed by Shaw in 1998 [13] as a means of applying constraint programming (CP) techniques to large vehicle routing problems. In its basic form, an initial solution is generated and then refined in successive iterations. Each iteration involves firstly selecting a subset of variables (the neighborhood), whose assignment is relaxed while all other variables have their assignment fixed to the value in the current solution. The neighborhood of unassigned variables can then be solved using a systematic approach, like CP or MIP, to find the optimal solution to the neighborhood given the assignment of the non-neighborhood variables.

2.1 Machine Reassignment Problem

The Machine Reassignment Problem has received considerable attention in the literature, in particular the problem as defined by Google in Roadef12. Given

an assignment of processes to machines, the problem involves reassigning the processes to minimise a multi-objective cost function related to the migration of processes being reassigned. While some recent works (e.g [11]) have tackled the problem as a multiobjective optimisation problem, we consider the problem in its classical format as defined by Google. Here, the cost function is converted to a single objective function using a weighted sum of costs.

The costs are associated with: the resource load above safety capacity on machines including transient resource usage, where a process uses resources on both the original machine and the machine of its reassignment; the balance of resource usage on machines; and costs associated with migrating processes of services between machine pairs. This problem is further subject to constraints such as capacity of machines, relationships between process subsets and machine subsets, etc.

A recent study [2] shows that most current state-of-the-art approaches apply some variation of local search techniques to address the MRP. They also observed the superiority of the approaches that adapt the search strategy to the characteristics of each instance.

One of the most effective local search approaches to address the MRP came from Gavranovic and Buljubasic in [3]. An important component of the technique was a noising method to help avoid local optima. When the algorithm got stuck in a local minima, the weight of one of the objectives in the multiobjective cost function was changed. The search can then escape the local optima as it was specific to the previous objective function. When a new local optimal is reached, the approach returns to the original objective function to escape. This approach was the winner of Roadef12.

A number of LNS approaches for the MRP have been proposed, which are of particular relevance to this work. Indeed the second place entry was a CP-based LNS [8]. Mehta et al. investigated both a CP-based LNS and a MIP-based LNS for the problem, and found the CP-based LNS approach significantly more effective, particularly on the large-scale instances. This LNS approach does however have a number of parameters that are highly sensitive to the problem instance characteristics.

An improved method was proposed in subsequent work [7] to counter the issue of parameter sensitivity, using a non-model based portfolio approach (ISAC [6]) to tune the LNS parameters to clusters of similar instances. More recently another LNS approach was proposed by Brandt et al. [1] where four domain specific neighbourhood selection heuristics were evaluated, and only small sub-problems (less than 10 processes) were considered. However, this did not achieve the same level of performance as that of Mehta et al. which is the state of the art LNS approach.

In terms of overall state-of-the-art, Turkey [14] recently proposed two bi-level hyper-heuristic approaches, the first involving local search and the second involving ant colony optimization. The results presented demonstrated that this method was able to outperform most approaches on the Roadef12 instances, achieving a number of new upper bounds for instances.

3 Problem Definition

The Machine Reassignment Problem aims to reallocate a set of processes to a set of machines in order to minimise a multi-objective function subject to a number of constraints. We describe the problem as follows (due to space limitations, the reader is referred to the original problem specification² for more details).

We have a set of *machines* $m \in M$ that can be assigned processes. Each machine has strict capacity restrictions $C_{m,r}$ on its different resources ($r \in R$), e.g. CPU, Load, Disk. Machines also have associated safety capacities $SC_{m,r}$ per resource that can be exceeded but incur a penalty in so doing. Machines are further grouped into *locations* ($l \in L$), which are disjoint sets of machines. Similarly machines are grouped into *neighborhoods* ($n \in N$). Locations and neighborhoods handle different requirements that will be described subsequently.

Running on the machines are a set of *processes* ($p \in P$). Each process has associated resource requirements ($RE_{p,r}$). Processes are grouped into *services* ($s \in S$). Two processes of the same service cannot be assigned to the same machine. Processes of the same services have to be spread across machines in a minimum number of locations ($spreadMin_s$). Furthermore, services can have dependencies, if service s_1 depends on service s_2 then processes of s_1 must be assigned to machines in the *neighborhood* of machines handling processes of s_2 .

Unbalanced resource usage is penalised according to balance rules $b \in B$. b is characterized by $(b_{r1}, b_{r2}, b_{target})$, the two related resources and the acceptable imbalance between them. A solution A is an assignment of $\forall p \in P$ to a machine $m \in M$. We will formally represent the assignment by $MC_p = m$ for each process p , while the original machine assignment is denoted by MO_p .

The MRP multiobjective function to minimise involves five different costs. The first cost is the load cost, the usage of machine resources above the safety capacity. The second cost is the balance cost, i.e. the imbalance of resources usage in each machine.

The final three costs are related to the cost of migrating a process from its original machine MO_p to a new machine MC_p . The first of these is the process move cost, where each process has an associated fixed cost to deter moving it. The second migration cost is the service move cost, which aims to penalise solutions that don't have moves balanced across services. The final cost is the machine move cost, which has a penalty depending on the pair of machines involved in a move.

The MRP problem can thus be modeled as:

$$\begin{aligned}
 - LC &= \sum_{r \in R} weight_{LC_r} * (\sum_{m \in M} \max(0, U_{m,r} - SC_{m,r})) \\
 - BC &= \sum_{b \in B} weight_{BC_b} * (\sum_{m \in M} \max(0, b_{target} * (C_{m,b_{r1}} - U_{m,b_{r1}}) - \\
 &\quad (C_{m,b_{r2}} - U_{m,b_{r2}}))) \\
 - PrMC &= weight_{PMC} * (\sum_{p \in P \wedge MO_p \neq MC_p} PMC_p) \\
 - SMC &= weight_{SMC} * \max_{s \in S} \sum_{p \in s \wedge MO_p \neq MC_p} 1 \\
 - MaMC &= weight_{MMC} * \sum_{p \in P} MMC_{MO_p, MC_p}
 \end{aligned}$$

² https://www.roadef.org/challenge/2012/files/problem_definition.v1.pdf.

$$\text{minimize } \sum LC + BC + PrMC + MaMC + SMC \quad (1)$$

$$\text{subject to: } MC_p \in M \quad \forall p \in P \quad (2)$$

$$U_{m,r} + TU_{m,r} \leq C_{m,r} \quad \forall m \in M, \forall r \in R \quad (3)$$

$$MC_p \neq MC_j \quad \forall p, j \in s, \forall s \in S, p \neq j \quad (4)$$

$$\sum_{l \in L} \min(1, \sum_{p \in S \wedge MC_p \in l} 1) \geq spreadMin_s \quad \forall s \in S \quad (5)$$

$$\min(1, \sum_{p1 \in s1 \wedge MC_{p1} \in n} 1) \leq \min(1, \sum_{p2 \in s2 \wedge MC_{p2} \in n} 1) \quad \forall n \in N, s1 \text{ depends } s2 \quad (6)$$

Constraint 2 enforces that each process is assigned to a machine. Constraint 3 ensures that the machine resources are not overloaded, where resource usage of resource r on machine m is given by $U_{m,r} = \sum_{p \in P \wedge MC_p = m} re_{p,r}$ and transient resource usage is defined by $TU_{m,r} = \sum_{p \in P \wedge MC_p \neq m \wedge MO_p = m} re_{p,r}$. Constraint 4 establishes that processes from the same service cannot be assigned for the same machine. Constraint 5 defines that the set of processes from a service have to be assigned to machines in a minimum number of different locations. Constraint 6 assures that if service s_1 has a dependency of service s_2 , each process in s_1 has to be assigned for machines in the same neighbourhood of those process of s_2 .

4 Proposed Algorithm

We implemented a Large Neighbourhood Search (LNS) for the MRP and compared two approaches for optimising the sub-problems within a CP solver. The first method is a random restart strategy (*RRS*) where search is restarted based on a failure threshold. We added a stochastic component to the variable/value ordering heuristics by choosing randomly amongst the top x choices of the heuristic. Therefore each restart is likely to explore a different part of the search space. This approach runs a backtracking search for a number of times, in each of them the failure threshold is increased based on *maxFails*. The termination criteria is a maximum failure threshold.

The second approach is a variation of Limited Discrepancy Search (LDS) [4], which we refer to as Restricted Domain Search (*RDS*). Note in our case we do not have the objective of *proving* optimality in each neighbourhood, our objective is to find the best possible solution for the neighbourhood in a small execution time. Thus, the idea behind this approach is to equally investigate every variable in the sub-problem.

Algorithm 1 describes the proposed RDS approach. It is a recursive function that, in each call, selects one process and investigates the D best machines to

assign the selected process to. When all processes are assigned, it checks whether the current solution is better than the best solution so far. If so, it updates the best solution with the current solution. Note this is based on a similar logic to LDS but rather than return to the top of the search tree at each increase of deviation, RDS performs its deviations in depth first search. The reason for this difference with LDS is the cost of propagation of assignments at the top of the search tree for problems of this nature.

Algorithm 1: *RestrictedDomainSearch()*

```

if qttUnassignedProcesses == 0 then
    if oldObjectiveCost > currentObjectiveCost then
        | solutionSubProblem  $\leftarrow$  saveSolution() ;
    end
else
    process  $\leftarrow$  selectAndRemoveProcess() ;
    UnassignedProcesses -- ;
    updateDomain(process) ;
    if domain(process) == 0 then
        | Failures ++ ;
    else
        discrepancy  $\leftarrow$  0 ;
        while domain(process) > 0 & checkTime() & discrepancy <=
            MaxDiscrepancy & qttFailures <= MaxFailures do
            machine  $\leftarrow$  selectAndRemoveMachine(process) ;
            if isConsistent(process, machine) then
                | assignProcessToMachine(process, machine) ;
                | propagateConstraints(machine) ;
                | RestrictedDomainSearch() ;
                | discrepancy ++ ;
                | unassignProcess(process, machine) ;
            else
                | Failures ++ ;
            end
        end
    end
    UnassignedProcesses ++ ;
end

```

4.1 Adaptive Neighbourhood Size

An important parameter of Large Neighbourhood Search is the size of each neighbourhood. A very large neighbourhood requires too much time to be optimised which results in a poor investigation of other neighbourhoods. On the other hand too small a neighbourhood can result in getting stuck in local minima.

There are many approaches in the literature that address this problem by implementing an adaptive large neighborhood search, e.g. [5, 10, 12]. In this work a simple adaptive neighbourhood size method is used primarily for escaping local minima. A relatively small initial neighbourhood size is used until search reaches a local minimum, whereupon the neighbourhood size is increased until it leaves the local minimum, and the original neighbourhood size is then restored.

4.2 Neighbourhood Selection

To define which variables should be relaxed on each iteration, our LNS approach focuses on the unbalanced usage of resources in each machine. The idea is that if a machine has an unbalanced usage, where the proportion of capacity used by one resource differs greatly from other resources, it is likely a better solution can be achieved by reassigning processes of this machine.

However, this heuristic does not consider all components of the multi-objective function. Therefore, we alternated with a heuristic based on the maximum machine cost. Algorithms 2 and 3 show the process to create the sub-problems.

Algorithm 2: Create subProblem Unbalanced Machine

```

numMachine  $\leftarrow$  (random()%(subProblemSize/2)) + 1 ;
machine  $\leftarrow$  getUnbalancedMachine() ;
while unassignedProcessQtt < numProcesses do
    if (numProcesses  $\geq$  (subProblemSize/numMachine)) then
        numProcesses  $\leftarrow$  0 ;
        machine  $\leftarrow$  getUnbalancedMachine() ;
    end
    if isHeuristicUsed then
        | process  $\leftarrow$  getMaxCostProcess(machine) ;
    else
        | process  $\leftarrow$  randomProcess(machine) ;
    end
    unassignProcess(process) ;
    addToSubProblem(process) ;
end

```

4.3 Variable and Value Ordering Heuristic

To select the variable we used the well known Fail First heuristic, that simply orders the variable based on minimising domain size. This heuristic is robust and widely used in CP solvers. Furthermore, this heuristic also indirectly incorporates the knowledge of “Big Processes First” highlighted by the winner of Roade12 [3], larger processes are harder to assign to machines due to the capacity constraint, therefore the domain of possible machines for these processes is smaller.

For the value ordering heuristic we implemented an approximation of the minimum cost of assigning the process to a machine. This heuristic has a high

accuracy at the bottom of the search tree but loses accuracy as we approach the root. To alleviate this issue towards the root, ties were broken based on the number of remaining unassigned processes that could be placed on the machine.

Algorithm 3: getUnbalancedMachine()

```

if machineIndicesSize == 0 or solutionWasImproved then
    solutionWasImproved  $\leftarrow$  FALSE ;
    machineIndicesSize = machineIndices.size() ;
    if useFirstSort then
        useFirstSort  $\leftarrow$  FALSE ;
        machineIndices  $\leftarrow$  machinesSortedByResourcesAvaliable() ;
    else
        useFirstSort  $\leftarrow$  TRUE ;
        machineIndices  $\leftarrow$  machinesSortedByMachineCost() ;
    end
end
if isHeuristicUsed then
    isHeuristicUsed  $\leftarrow$  FALSE ;
    machineIndicesSize  $\leftarrow$  machineIndicesSize - 1 ;
    machine  $\leftarrow$  machineIndices[machineIndicesSize] ;
else
    isHeuristicUsed  $\leftarrow$  TRUE ;
    machine  $\leftarrow$  randomMachine() ;
end

```

4.4 Early Search Noise Strategy

The double use of transient resources when migrating a process is an important aspect that must be considered in this problem. On some instances it can result in local optimal solutions that are very difficult to escape, as many machines will be overloaded with double use of transient resources. To avoid a premature convergence for some of those intermediate solutions, we added an extra component to the objective function to discourage process moves with a high level of transient resource usage. The weight of this component is reduced each time the algorithm reaches a fixed threshold of iterations without improvement, and set to 0 on the last 20% of the search runtime.

5 Evaluation

The experiments were run on a Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-70-generic) with 16 Core and 32 GB. All runs had a runtime cutoff of 5 min per instance. Furthermore, as the proposed approach has stochastic components, the presented results are the average of 5 runs with different seeds. Table 1 presents the parameter configurations that were used to run the experiments.

Table 1. Configurations parameters for the benchmark experiment.

Parameter	Value
Runtime	300 s
Initial neighbourhood size	10 processes
Threshold of non-improvement	50 iterations
Failure threshold	400
Limit of deviation	2

The experiments used the three sets of instances from Roade12, where each set has 10 instances. The ‘A’ instance set is composed of smaller instances with a maximum of 1,000 processes and 100 Machines. The other two sets of instances are more complex and larger, with up to 50,000 processes and 5,000 machines.

5.1 Results

We first investigated the neighborhood search method, comparing Restricted Domain Search (RDS) with Random Restarted Search (RRS) on a range of neighbourhoods across the 30 instances. We observed that on a total of 271,219 neighbourhoods where one of these approaches managed to find a better (and improving) solution, RDS found better solutions in 54% of the neighbourhoods while RRS found better solutions in 46% of the neighbourhoods.

Furthermore, as we can see in Fig. 1, RDS is consistently faster than RRS across all Roade12 instances when the same neighborhoods were explored. This behaviour can be explained by the fact that every time the search is restarted in RRS, the algorithm has to assign many variables and propagate constraints before the search starts to have complete solutions or failures, which considerably increases the run-time when compared with RDS. We finally tested each search method independently (5 runs, 5 mins per run per instance) and found that RRS, while performing relatively well, never found better solutions on an instance when compared to RDS. These results demonstrate the quality of our novel RDS approach.

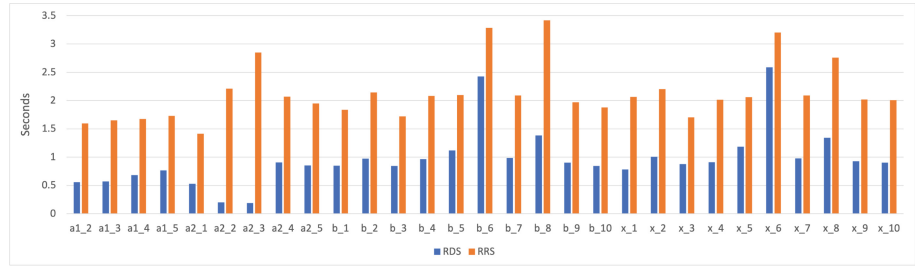


Fig. 1. Comparing the average Run-time of RDS and RRS in the same neighbourhoods.

Table 2. Average cost results and % gap to best known solution for RDS-LNS compared with state of the art. Best known solution taken from [2,14]. All approaches had a 5 min cutoff per run. CP-LNS and RDS-LNS run on same machine, results are average of 5 runs. NLS and Ant-HH results taken from paper, former is average of 100 runs, latter is average of 31 runs. Bold indicates best amongst the four comparison approaches according to the given metric.

Instances	BK Cost	Cost				% Gap = 100* (Cost - BK)/BK			
		Ant-HH _[14]	NLS _[3]	CP-LNS _[8]	RDS-LNS	Ant-HH	NLS	CPLNS	RDS
a1.1	44306501	44306501	44306501	44306501	44306501	0.0	0.0	0.0	0.0
a1.2	777532177	777532179	778142261	778654913	777680794	0.0	0.0	0.0	0.0
a1.3	583005715	583005715	583006320	583005829	583005836	0.0	0.0	0.0	0.0
a1.4	244875200	244875200	259815285	254185892	249753518	0.0	6.0	4.0	2.0
a1.5	727578306	727578306	727578311	727578311	727578311	0.0	0.0	0.0	0.0
a2.1	161	165	333	201	159	2.0	107.0	25.0	-1.0
a2.2	720671511	720671511	740140535	803912789	765776193	0.0	3.0	12.0	6.0
a2.3	1182260491	1190713410	1210207120	1304726522	1245045531	1.0	2.0	10.0	5.0
a2.4	1680368560	1680368560	1680629156	1683592281	1683447864	0.0	0.0	0.0	0.0
a2.5	307150821	307150821	317804454	339471433	362894481	0.0	3.0	11.0	18.0
b.1	3291069365	3291069365	3343410128	3339134760	3352472777	0.0	2.0	1.0	2.0
b.2	1010949451	1015482891	1015561513	1024629389	1028734683	0.0	0.0	1.0	2.0
b.3	156519816	156691279	157737166	157512118	159219336	0.0	1.0	1.0	2.0
b.4	4677792536	4677792536	4677981438	4677833576	4677858641	0.0	0.0	0.0	0.0
b.5	922944510	922944510	923905512	923721068	930511569	0.0	0.0	0.0	1.0
b.6	9525851389	9525851389	9525934654	9525870196	9525853048	0.0	0.0	0.0	0.0
b.7	14834456020	14834456193	14835328102	14853146933	14878044937	0.0	0.0	0.0	0.0
b.8	1214291129	1214291141	1214453127	1214589052	1214526543	0.0	0.0	0.0	0.0
b.9	15885437252	15885437252	15885693227	15885768231	15885669118	0.0	0.0	0.0	0.0
b.10	18048187105	18048187105	18048711483	18069108773	18130155888	0.0	0.0	0.0	0.0
x.1	3030246091	3044411001	3065081130	3106902032	3102713793	0.0	1.0	3.0	2.0
x.2	1002379317	1002379317	1003356104	1015807185	1010454229	0.0	0.0	1.0	1.0
x.3	69970	75155	341508	811958	136036	7.0	388.0	1060.0	94.0
x.4	4721586142	4721586142	4721856521	4721635533	4721697878	0.0	0.0	0.0	0.0
x.5	54132	57974	160418	100449	53899	7.0	196.0	86.0	-0.0
x.6	9546936159	9546936159	9546972261	9546952069	9546938475	0.0	0.0	0.0	0.0
x.7	14252476500	14252476500	14253212517	14397486190	14349370657	0.0	0.0	1.0	1.0
x.8	29193	32014	147269	65953	30678	10.0	404.0	126.0	5.0
x.9	16125531142	16125531142	16125760293	16125916363	16125823486	0.0	0.0	0.0	0.0
x.10	17815045320	17815981156	17815072367	17839540583	17903620248	0.0	0.0	0.0	0.0

The results of RDS-LNS is given, along with comparison results from other approaches in the literature, in Table 2. In particular we present results for the best LNS approach from the literature (CP-LNS [8]), the winner of Roadef12 (NLS [3]), and the current state of the art (Ant-HH [14]). We also provide the reference best known solution (*BK_Cost*) for each instance.

For the CP-LNS approach, we ran their code with the same experimental setup on the same machine as our experiments. The results for the other two methods were taken from their respective references. Both used a 5 min runtime cutoff but NLS is an average of 100 runs, while Ant-HH is an average of 31 runs.

The results for each method on each instance are given in terms of average objective value across runs, and in terms of gap to the best known solution. Looking at the gap, the first point to note is that RDS-LNS was within 2% of the best known solution in its average performance on 25/30 instances, and less than 1% for 17 of these.

Comparing our approach with the other methods in the table, we first consider the other LNS approach. This is most directly comparable both because it's a CP-based LNS approach, and because experiments were performed under identical conditions. RDS-LNS found better results than CP-LNS in 17/30 instances, and had identical averages for two others. Of the remaining 11 instances that CP-LNS had better results, RDS-LNS was within 1% for all but one (instance a2.5). Results were also impressive in comparison Roadef12 winner NLS [3], our approach outperformed their approach on 13/30 instances, while for the majority of the other 17 instances the RDS-LNS results are within 2%. Finally, comparing against the current SOA, Ant-HH [14], RDS-LNS only managed to perform better on 3/30 instances. However, RDS-LNS was within 1% for all but nine instances, and within 10% for all but two of those nine.

Even more importantly, despite the amount of research that has been dedicated to these instances in the past decade, upon further analysis of our results we found that RDS-LNS improved on the best known solution [2, 14] for three instances (a2.1, x.5 and x.8). Indeed, as can be seen in Table reftable:Result, the *average* performance alone was better than the best known. Based on the best solution among our 5 runs, the new upper bounds found were 153, 45922, and 28564 respectively (equating to roughly 5%, 15%, and 2% reductions in cost compared to the current best known cost). This is even more noteworthy given that some of the other approaches in the literature had a 30 min cutoff [1], or had many more runs, e.g. 31 [14] and 100 [3] and did not find such best solutions for these three instances.

6 Conclusion

In this paper, we proposed a new Large Neighbourhood Search approach for the Machine Reassignment Problem with a novel domain specific neighbourhood operator, and novel search strategy for the subproblems. Our empirical evaluation demonstrated the quality of the approach on a well studied problem set, resulting in improvement on best known solution for three of the thirty instances. Comparison of our results against other state-of-the-art demonstrated the effectiveness of RDS-LNS. Further analysis of the sub-problem optimisation strategies showed the superiority of the proposed Restricted Domain Search when compared with a standard random restart strategy.

References

1. Brandt, F., Speck, J., Völker, M.: Constraint-based large neighborhood search for machine reassignment. *Ann. Oper. Res.* **242**(1), 63–91 (2016)

2. Canales, D., Rojas-Morales, N., Riff, M.C.: A survey and a classification of recent approaches to solve the google machine reassignment problem. *IEEE Access* **8**, 88815–88829 (2020)
3. Gavranović, H., Buljubašić, M.: An efficient local search with noising strategy for google machine reassignment problem. *Ann. Oper. Res.* **242**(1), 19–31 (2016)
4. Harvey, W.D., Ginsberg, M.L.: Limited discrepancy search. In: *IJCAI*, no. 1, pp. 607–615 (1995)
5. He, K., Tole, K., Ni, F., Yuan, Y., Liao, L.: Adaptive large neighborhood search for circle bin packing problem. *arXiv preprint arXiv:2001.07709* (2020)
6. Kadioglu, S., Malitsky, Y., Sellmann, M., Tierney, K.: ISAC–instance-specific algorithm configuration. In: *ECAI 2010*, pp. 751–756. IOS Press (2010)
7. Malitsky, Y., Mehta, D., O’Sullivan, B., Simonis, H.: Tuning parameters of large neighborhood search for the machine reassignment problem. In: Gomes, C., Sellmann, M. (eds.) *CPAIOR 2013. LNCS*, vol. 7874, pp. 176–192. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38171-3_12
8. Mehta, D., O’Sullivan, B., Simonis, H.: Comparing solution methods for the machine reassignment problem. In: Milano, M. (ed.) *CP 2012. LNCS*, pp. 782–797. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33558-7_56
9. Murat Afsar, H., Artigues, C., Bourreau, E., Kedad-Sidhoum, S.: Machine reassignment problem: the roadef/euro challenge 2012 (2016)
10. Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* **40**(4), 455–472 (2006)
11. Saber, T., Gandibleux, X., O’Neill, M., Murphy, L., Ventresque, A.: A comparative study of multi-objective machine reassignment algorithms for data centres. *J. Heuristics* **26**(1), 119–150 (2020)
12. Sacramento, D., Pisinger, D., Ropke, S.: An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transp. Res. Part C: Emerg. Technol.* **102**, 289–315 (2019)
13. Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: Maher, M., Puget, J.-F. (eds.) *CP 1998. LNCS*, vol. 1520, pp. 417–431. Springer, Heidelberg (1998). https://doi.org/10.1007/3-540-49481-2_30
14. Turkey, A.: Bi-level hyper-heuristic approaches for combinatorial optimisation problems. Ph.D. thesis, RMIT University (2019)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Bayesian Optimization with Multi-objective Acquisition Function for Bilevel Problems

Vedat Dogan¹(✉)  and Steven Prestwich² 

¹ Confirm Centre for Smart Manufacturing, School of Computer Science
and Information Technology, University College Cork, Cork, Ireland
`vedat.dogan@cs.ucc.ie`

² Insight Centre for Data Analytics, School of Computer Science and Information
Technology, University College Cork, Cork, Ireland
`s.prestwich@cs.ucc.ie`

Abstract. A bilevel optimization problem consists of an upper-level and a lower-level optimization problem connected to each other hierarchically. Efficient methods exist for special cases, but in general solving these problems is difficult. Bayesian optimization methods are an interesting approach that speed up search using an acquisition function, and this paper proposes a modified Bayesian approach. It treats the upper-level problem as an expensive black-box function, and uses multiple acquisition functions in a multi-objective manner by exploring the Pareto-front. Experiments on popular bilevel benchmark problems show the advantage of the method.

Keywords: Bayesian optimization · Bilevel optimization problems · Multi-objective acquisition · Multi-objective optimization

1 Introduction

Bilevel optimization deals with optimization problems including additional optimization problem within the constraints. Two decision-makers attempt to find his/her optimal solution on these hierarchical nested systems. The upper-level problem is the first problem, and the decision-maker is called the leader. The lower-level problem forms a constraint in the leader problem, and this decision-maker is called the follower. The leader knows the follower's objective and constraints, but the follower may have no knowledge about the leader. The decision-makers objectives are often in conflict though they may also be cooperative. During the optimization process the leader takes his/her action first. The follower takes that decision as a parameter and tries to find the best reaction. However,

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 16/RC/3918.

© The Author(s) 2023

L. Longo and R. O'Reilly (Eds.): AICS 2022, CCIS 1662, pp. 409–422, 2023.

https://doi.org/10.1007/978-3-031-26438-2_32

the follower’s reaction affects the leader’s decisions because the leader makes choices in the knowledge of how the follower will react.

Bilevel optimization problems occur in many practical applications including transportation, management, environmental economics, engineering and design. [43]. They also occur in machine learning: signal processing, meta-learning, hyperparameter optimization, reinforcement learning and neural architecture search can be modelled as bilevel optimization [21]. However, a lack of efficient solution methods has prevented the uptake of bilevel optimization.

The aim of this paper is to propose a new approach based on Bayesian optimization (BO) using multiple acquisition functions (MACBP) to improve efficiency (defined in terms of function evaluations). BO is a surrogate-based method for solving black-box functions that are expensive to evaluate [16], making it a useful approach to solving bilevel problems. An example is the BOBP algorithm [23]. BOBP used one lower confidence bound (LCB) acquisition function and obtains one decision point at a time. We propose using more than one acquisition function to improve the optimization process by making a wiser choice of acquisition points. Multiple acquisition functions have been used in BO, for example in the MACE algorithm for optimizing analog circuit design [33]. However, to the best of our knowledge no work has been done on this area for solving bilevel optimization problems.

Our contributions are twofold:

- We use multiple acquisition functions, as no single acquisition function is appropriate for every problem [18]. We solve the resulting multi-objective optimization problem with evolutionary techniques, and select new points on the Pareto-front solution set.
- We show empirically how using multiple acquisition functions affects optimization performance.

The rest of the paper is organised as follows. Background is provided in Sect. 2. The preliminaries for general bilevel optimization problems and BO are given in Sect. 3. The proposed method and algorithm details are explained in Sect. 4. In Sect. 5 the experimental setup is described. Finally, Sect. 6 concludes the paper and proposes future work.

2 Background

Bilevel optimization problems are described in two areas. In game theory von Stackelberg [50] proposed descriptive models of decision behaviour and built game-theoretic equilibria. In mathematical programming problems containing nested lower-level optimization problem as a constraint of upper-level optimization problem [8]. The hierarchical structure of bilevel problems might cause difficulties such as non-convexity and having no relation between instances. It is known to be strongly NP-hard [17].

A considerable number of exact approaches have been applied to bilevel problems. *Karush-Kuhn-Tucker* conditions [3] can be used to reformulate a bilevel

problem to a single-level problem. Penalty functions compute the stationary points and local optima. Vertex enumeration has been used with a version of the Simplex method [6]. Gradient information for the follower problem can be extracted for use by the leader objective function. In terms of integer and mixed integer bilevel problems, reformulation [14], branch-and-bound [4] and parametric programming approaches have been applied to solve bilevel problems [27].

Because of the inefficiency of exact methods in complex bilevel problems, several kinds of meta-heuristics have been applied to bilevel problems in the literature. Four existing categories have been published in [53]: the nested sequential approach [25], the single-level transformation approach, the multi-objective approach [41] and the co-evolutionary approach [31]. An algorithm based on a human evolutionary model for non-linear bilevel problems [34], and the Bilevel Evolutionary Algorithm based on Quadratic approximations (BLEAQ) have been proposed [45]. This is another work which attempts to try to reduce the number of follower optimizations. The algorithm approximates the inducible region through the feasible region of the bilevel problem. In [40] they consider single optimization problem at both levels. They propose the Sequential Averaging Method (SAM) algorithm. In different recent works [32, 42] they used a truncated back-propagation approach to approximate the (stochastic) gradient of the upper-level problem. Basically, they use a dynamical system to model an optimization algorithm that solves the lower-level problem, and replaces the lower-level optimal solution. In another work [19] they developed a two-timescale stochastic approximation algorithm (TTSA) for solving a bilevel problem assuming the follower problem is unconstrained and strongly convex and the leader is a smooth objective function.

Many practical problems can be modelled and solved as Stackelberg games in the field of economics [46, 47] including principal agency problems and policy decisions. Hierarchical decision-making processes in management [2, 51] and in engineering and optimal structure design are other practical examples [24, 48]. Network design and the toll setting problem are the most popular applications in the field of transportation [9, 11, 35]. Finding optimal chemical equilibria, planning the preposition of defensive missile interceptors to counter an attacking threat, and interdicting nuclear weapons are other applications [10]. Inverse optimal control problems are modelled as bilevel optimization problems in nature [22, 37, 52]. There are many applications in robotics, computer vision, communication theory etc. In the machine learning community, bilevel optimization received significant attention recently and became an important framework in applications. Some interesting topics are meta-learning [5, 15, 39], hyperparameter optimization [13, 42], reinforcement learning [19, 26] and signal processing [29].

3 Preliminaries

The description of the MACBP algorithm will be divided into three parts. Firstly, we explain bilevel programming problems and their structure. Secondly, we discuss Bayesian optimization (BO) and Gaussian processes (GP). Finally, we propose the MACBP algorithm for solving bilevel optimization problems.

3.1 Bilevel Optimization Problems

For the upper-level objective function $F : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ and lower-level objective function $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$, bilevel optimization problem can be defined as

$$\begin{aligned} \min_{\mathbf{x}_u \in X_u, \mathbf{x}_l \in X_l} \quad & F(\mathbf{x}_u, \mathbf{x}_l) \\ \text{s.t. } \quad & \mathbf{x}_l \in \underset{\mathbf{x}_l}{\operatorname{argmin}} \{f(\mathbf{x}_u, \mathbf{x}_l) : g_j(\mathbf{x}_u, \mathbf{x}_l) \leq 0, j = 1, 2, \dots, J\} \\ & G_k(\mathbf{x}_u, \mathbf{x}_l) \leq 0, k = 1, 2, \dots, K \end{aligned} \quad (1)$$

where $\mathbf{x}_u \in X_U, \mathbf{x}_l \in X_L$ are upper-level and lower-level decision variables and decision spaces, G_k, g_j are constraints.

Because the lower-level decision maker depends on the upper-level variables, for every decision x_u , there is a follower-optimal decision x_l^* . In bilevel optimization, the decision set $\mathbf{x}^* = (\mathbf{x}_u^*, \mathbf{x}_l^*)$ is a feasible member for the upper-level *only if* it satisfies all the upper-level constraints and vector \mathbf{x}_x^* is an optimal solution to the lower-level problem with upper-level decision as *parameter*.

3.2 Bayesian Optimization and Gaussian Process

BO is a method to optimize expensive-to-evaluate black-box functions. The probabilistic surrogate model and acquisition functions is important for BO. Predictions and uncertainties are provided by the surrogate model. It uses commonly GP [49] as a surrogate model, to obtain a posterior distribution $\mathbb{P}(\mathbf{f}|D)$ over the objective function \mathbf{f} given the observed data $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$. An acquisition function uses the posterior distribution to explore the search space. So the surrogate model is assisted by an *acquisition function* to choose the next candidate or a set of candidates $X_{\text{cand}} = \{\mathbf{x}_i\}_{i=1}^q$. Though the objective function is expensive to evaluate, the surrogate-based acquisition function is not, so it can be optimized much more easier than the true function to yield X_{cand} .

Let us assume that we have a set of collection points $\{x_1, \dots, x_n\} \in \mathbb{R}^d$ and an objective function values of these points $\{f(x_1), \dots, f(x_n)\}$. After we observe n points, the mean vector is obtained by evaluating a *mean function* μ_0 at each point x_i and the covariance matrix by evaluating a *covariance function* or *kernel* Σ_0 at each pair of x_i, x_j . The resulting prior distribution on $\{f(x_1), \dots, f(x_n)\}$ is defined by

$$f(x_{1:n}) \sim N(\mu_0(x_{1:n}), \Sigma_0(x_{1:n}, x_{1:n})) \quad (2)$$

Let us suppose we wish to find a value of $f(X_{\text{cand}})$ at some new candidate point X_{cand} . For this purpose, the prior over $\{f(x_{1:n}), f(X_{\text{cand}})\}$ is given by (2). Then we can compute the distribution of $f(X_{\text{cand}})$ given the observations

$$f(X_{\text{cand}})|f(x_{1:n}) \sim N(\mu_0(X_{\text{cand}}), \sigma_0^2(X_{\text{cand}})) \quad (3)$$

$$\mu_0(X_{\text{cand}}) = \Sigma_0(X_{\text{cand}}, x_{1:n})\Sigma_0(x_{1:n}, x_{1:n})^{-1}(f(x_{1:n}) - \mu_0(x_{1:n})) + \mu_0(X_{\text{cand}}) \quad (4)$$

$$\sigma_n^2(X_{cand}) = \Sigma_0(X_{cand}, X_{cand}) - \Sigma_0(X_{cand}, x_{1:n})(\Sigma_0(x_{1:n}, x_{1:n})^{-1} \Sigma_0(x_{1:n}, X_{cand})) \quad (5)$$

The distribution is called the *posterior probability distribution* in Bayesian statistics. So it is very important during the Bayesian optimization and Gaussian process to choose the next point to evaluate.

Acquisition functions are used to guide the search to a promising next point during the likelihood optimization, and it balances exploration and exploitation. Several acquisition functions have been developed over the years, such as probability of improvement (PI), expected improvement (EI) and upper confidence bound (UCB).

Probability of Improvement. The PI acquisition function tries to measure the probability that an arbitrary x exceeds the current best. Given the minimum objective function value τ in the data set, the formulation is as follows [30]:

$$PI(x) = \Phi(\lambda) \quad (6)$$

where $\Phi(\lambda)$ is the cumulative distribution function of standard normal distribution and $\lambda = (\tau - \mu(x))/(\sigma(x))$.

Expected Improvement. We can expect that the observation x will not only reach the current best, but also reach the current best value at the highest magnitude. The corresponding formulation can be expressed as [36]:

$$EI(x) = \sigma(x)(\lambda\Phi(\lambda) + \phi(\lambda)) \quad (7)$$

where $\phi(\cdot)$ is probability density function of standard normal distribution and $\lambda = (\tau - \mu(x))/(\sigma(x))$.

Upper Confidence Bound. This is not an improvement-based strategies like EI and PI. It tries to guide the search from an optimistic perspective. The formulation is:

$$UCB(x) = \mu(x) + \beta\sigma(x) \quad (8)$$

where β is a parameter represents exploration-exploitation trade-off. We fix $\beta = 0.1$.

Algorithm 1. The MACBP Algorithm for Upper-level Optimization**Inputs:** $\mathbf{F}_u(\mathbf{x}_u, \mathbf{x}_l) : \mathbf{x}_u \in \mathbb{X}_u, \mathbf{x}_l \in \mathbb{X}_l$,Total epoch N ,Initial decision data set $D = (\mathbf{x}_{u_i}, F_u(\mathbf{x}_{u_i}, \mathbf{x}_{l_i}^*))_{i=1:n}$ with size of n ,

- 1: \mathbf{x}_l^* : Find the best lower-level decisions and set as upper-level parameters,
- 2: Initialize *Gaussian model* with Observations $\{\mathbf{x}_u, F_u(\mathbf{x}_u, \mathbf{x}_l^*)\}$
- 3: **for** $i = 0 : N$ **do**
- 4: Construct the acquisition functions according to the Eqs. (8), (7) and (6)
- 5: Find Pareto-front of multi-objective acquisition problem using the NSGA-II algorithm
- 6: For the chosen upper-level decision \mathbf{x}_u , find optimal \mathbf{x}_l^* by using SLSQP
- 7: Calculate fitness scores \mathbf{F}_u^* and \mathbf{f}_u^*
- 8: Update the data set $D = (\mathbf{x}_{u_i}, \mathbf{F}_u(\mathbf{x}_{u_i}, \mathbf{x}_{l_i}^*))_{i=1:q}$
- 9: Update Multi-objective Gaussian Process Model with new observations
- 10: **end for**
- 11: **Return** Best F_u^* and corresponding optimum variables $\mathbf{x}_u^*, \mathbf{x}_l^*$

4 Proposed Method

Bilevel problems have two levels of optimization tasks, such that the lower-level problem is a constraint of the upper-level problem. In general bilevel problems, the follower depends on the leader decisions x_u . The leader has no control over the follower decision x_l . For every leader decision there is an optimal follower decision, which can be called the reaction. Because the follower problem is a parametric optimization problem that depends on the leader decision x_u , it is very time-consuming to adopt a nested strategy approach which sequentially solves both levels. In the continuous domain, the computational cost is very high. During the optimization process, it is important to choose wisely the next leader decision x_u according to make the process faster. For this purpose, we will present the proposed algorithm, we call MACBP, for solving bilevel problems by BO via multiple acquisition functions.

Problem Statement. Let us assume that we have a expensive black-box function that takes leader decisions in leader decision space $x_u \in X_u$ and follower decisions coming from the follower decision maker $x_l \in X_l$ as input. The function returns a scalar fitness score:

$$F(x_u, x_l) : X_u \times X_l \rightarrow \mathbb{R} \quad (9)$$

Given a budget of N , the leader makes a decision and the follower makes its decisions accordingly. The leader can observe this information during the optimization process, and how follower decision maker reacting to leader decisions in every iteration and chooses the next leader decision to optimize the fitness score.

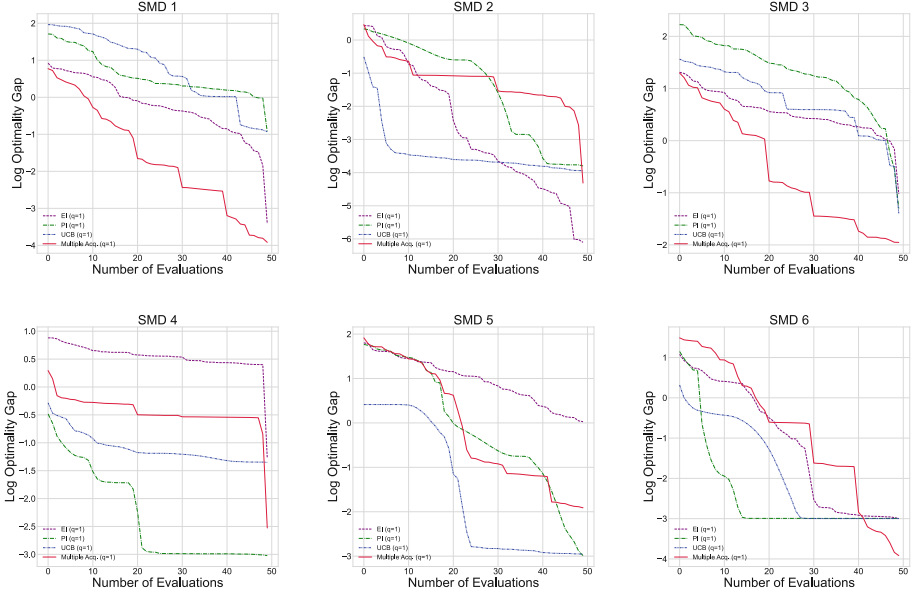


Fig. 1. The log optimality gap for the leader’s objective in SMD benchmark problems.

Algorithm Description. First we discuss fitting the decision data to the Gaussian process model. After observing n decision data $\{(x_u^i, y^i)\}_{i=1}^n$ where $y_i = F(x_u^i, x_l^i)$, we fit the data set to the Gaussian process model. After we have the data set let $\hat{X}^n = ((x_u)^1, \dots, (x_u)^n)$ and $Y^n = (y^1, \dots, y^n)$, then we define the Gaussian process by a prior mean $\mu(x_u)$ and prior covariance function $k((x_u), (x_u'))$. After observing n data points, let $K = k(\hat{X}^n, \hat{X}^n) \in \mathbb{R}^{n \times n}$. So the posterior mean and covariance is given by:

$$\mu(x_u)^n = \mu(x_u) + k(x_u, \hat{X}^n)(K + \sigma_0^2 I)^{-1}(Y^n - \mu(\hat{X}^n)) \quad (10)$$

$$k^n(x_u, x_u')^n = k^n(x_u, x_u') - k^n(x_u, \hat{X}^n)(K + \sigma_0^2 I)^{-1}k(\hat{X}^n, x_u') \quad (11)$$

After fitting the data to the model, we choose the next leader decision. After we find the optimal reaction (x_u^{n+1}, x_l^{n+1}) and the fitness score of leader function $F(x_u^{n+1}, x_l^{n+1}) = y^{n+1}$, we update the Gaussian process model with new decision data (x_u^{n+1}, y^{n+1}) . We shared the details of the MACBP algorithm on Algorithm 1 for upper-level optimization.

4.1 Multi-objective Optimization

There are multiple objectives to optimize when we consider the multi-objective optimization problems. It is formulated as

$$\underset{\mathbf{x} \in X}{\text{minimize}} \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_d(\mathbf{x})) \quad (12)$$

Table 1. The summary of SMD benchmark problems

	Dimension		Search domain		Optimal value	
	UL	LL	UL	LL	UL	LL
SMD1	2	2	$[-5, 10] \times [-5, 10]$	$[-5, 10] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$	0.0	0.0
SMD2	2	2	$[-5, 10] \times [-5, 1]$	$[-5, 10] \times (0, e]$	0.0	0.0
SMD3	2	2	$[-5, 10] \times [-5, 10]$	$[-5, 10] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$	0.0	0.0
SMD4	2	2	$[-5, 10] \times [-1, 1]$	$[-5, 10] \times (0, e]$	0.0	0.0
SMD5	2	2	$[-5, 10] \times [-5, 10]$	$[-5, 10] \times [-5, 10]$	0.0	0.0
SMD6	2	2	$[-5, 10] \times [-5, 10]$	$[-5, 10] \times [-5, 10]$	0.0	0.0

for a vector-valued function $\mathbf{f}(\mathbf{x}) : \mathbb{R} \rightarrow \mathbb{R}^d$ and $X \in \mathbb{R}$. So it is hard and commonly impossible to find a single optimum solution as there may be conflicts between the objectives. Therefore the main goal for these problems is to approximate the Pareto-front. Let us say that $\mathbf{f}(\mathbf{x})$ *dominates* another solution $\mathbf{f}(\mathbf{x}')$ if $\mathbf{f}^{(i)}(x) \succ \mathbf{f}^{(i)}(x')$ for all $i = 1, 2, \dots, M$ and there exists $i' \in \{1, 2, \dots, M\}$ such that $f^{i'}(x) \succ f^{i'}(x')$. So we can express the *Pareto-optimal* by $P^* = \{\mathbf{f}(\mathbf{x}) \text{ s.t. } \nexists \mathbf{x}' \in X : \mathbf{f}(\mathbf{x}') \succ \mathbf{f}(\mathbf{x})\}$ and $X^* = \{\mathbf{x} \in \mathbf{X} \text{ s.t. } \mathbf{f}(\mathbf{x}) \in P^*\}$. A solution set is Pareto-optimal if it is not dominated by any other point and it dominates at least one point. The Pareto-set the set of all Pareto-optimal points, and a set of Pareto-optimal points is called a Pareto-front. There are many multi-objective optimization algorithms such as non-dominated sorting based genetic algorithm (NSGA-II) [12], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [55] and multi-objective optimization based on differential evolution (DEMO) [54].

4.2 Multi-objective Acquisition Function in Bayesian Optimization

Different acquisition functions have different characteristics according to their structure and point selection strategy. Improvement based strategies rely on the best selection so far at each iteration. For example the PI function value decreases when difference between mean function the best objective value so far below zero, $\mu(x) - F^*(x) < 0$. The EI function value at sampled points would always be worse than the EI values at pending decision points. Uncertainty-based acquisition functions, for instance UCB, increase as $\sigma(x)$ increases.

According to the different selection strategies explained above, we use the multi-objective optimization method NSGA-II in this work, to find the best trade-off between acquisition functions. Then we select the next point of the leader's decision during the bilevel optimization process from the best trade-off between acquisition functions. This is called the Pareto-front of acquisition functions. So in every iteration the multi-objective optimization problem constructed is:

$$\underset{\mathbf{x} \in X}{\text{minimize}} \left\{ -UCB(x), -PI(x), -EI(x) \right\} \quad (13)$$

After we find the Pareto-front from the multi-objective optimization Problem 13 we make the random selection from Pareto-optimal decision set.

Table 2. Upper-level function evaluations for the proposed MACBP algorithm and other known algorithms for SMD1-SMD6

	Upper-Level Function Evaluations					
	MACBP	CGA-BS [1]	BLMA [20]	NBLE [20]	BIDE [20]	BLEAQ [45]
SMD1	9.70×10^1	1.01×10^4	1.19×10^3	1.52×10^3	6.00×10^3	1.19×10^3
SMD2	19.90×10^1	5.00×10^4	1.20×10^3	1.56×10^3	6.00×10^3	1.20×10^3
SMD3	10.30×10^1	1.00×10^4	1.29×10^3	1.56×10^3	6.00×10^3	1.29×10^3
SMD4	26.90×10^1	1.25×10^5	1.31×10^3	1.53×10^3	6.00×10^3	1.31×10^3
SMD5	17.00×10^1	1.00×10^5	2.06×10^3	3.40×10^3	6.00×10^3	2.06×10^3
SMD6	14.70×10^1	1.37×10^5	4.08×10^3	4.06×10^3	6.00×10^3	4.08×10^3

Table 3. Upper-level accuracy for the proposed MACBP algorithm and other known algorithms for SMD1-SMD6.

	Upper-Level Accuracy					
	MACBP	CGA-BS [1]	BLMA [20]	NBLE [20]	BIDE [20]	BLEAQ [45]
SMD1	2.51×10^{-6}	0	1.00×10^{-6}	5.03×10^{-6}	3.41×10^{-6}	1.00×10^{-6}
SMD2	1.33×10^{-6}	2.22×10^{-6}	1.00×10^{-6}	3.17×10^{-6}	1.29×10^{-6}	5.44×10^{-6}
SMD3	3.82×10^{-6}	0	1.00×10^{-6}	1.37×10^{-6}	4.10×10^{-6}	7.55×10^{-6}
SMD4	6.27×10^{-7}	3.41×10^{-11}	1.00×10^{-6}	9.29×10^{-6}	2.30×10^{-6}	1.00×10^{-6}
SMD5	7.44×10^{-7}	1.13×10^{-9}	1.00×10^{-6}	1.00×10^{-6}	1.58×10^{-6}	1.00×10^{-6}
SMD6	1.09×10^{-7}	9.34×10^{-11}	1.00×10^{-6}	1.00×10^{-6}	3.47×10^{-6}	1.00×10^{-6}

5 Experiments

We evaluate the MACBP algorithm using two experiments. First, we run the experiments by choosing a single point at each iteration for the setting of $N_{iter} = 50$. We set the number of initial random sampling to $N_{init} = 20$. Then, we compare the results with those for the three single acquisition functions EI, PI and UCB performances. Second, we run the experiment with stopping criteria of $d < 10^{-5}$ where d represents the difference between the results and the optimum value of functions. We compare the performance of our proposed method in terms of function evaluations in Table 2 and in terms of accuracy in Table 3. We run the algorithm in sequential mode and the Matern52 kernel is used for GP for both experiments. The parameters for acquisition functions are declared in Sect. 3.2. For the first experiment, the experiments are repeated 31 times to average the random fluctuations and the optimality gap in the log scale presented in Fig. 1.

The optimization is completed in a single core of 1.4 Ghz Quad Core i5, 8 Gb 2133 MHz LPDDR3 RAM. Bayesian optimization is implemented in the Python language and uses BoTorch [38], the SLSQP algorithm [28] is used for lower-level optimization, and the NSGA-II algorithm for multi-objective optimization by using PyMOO [7] library.

5.1 SMD Problems

We evaluated the MACBP algorithm on six standard benchmark problems proposed in [44]. It is called SMD test problems and the problems are unconstrained and high-dimensional with controllable complexities. They are scalable in terms of the number of decision variables. Each problem in the benchmark represents a different difficulty level in terms of convergence, the complexity of interaction and lower-level multi-modality as declared in [44]. Table 1 provides details on the problems. For all functions we used 2D decision variables. The total function evaluations for the leader's objective can be calculated by $N_{iter} + N_{init}$.

5.2 Results

Although bilevel optimization problems deal with the leader's and follower's optimization problems, we shall consider only the leader's performance as it is the only one we model as an expensive black-box function. The optimality gap plots between true optimal points and approximated points in 50 iterations *in log scale* are given in Fig. 1. As can be seen in Fig. 1, the proposed algorithm for bilevel optimization is competitive with the sequential Bayesian method at upper-level optimization with the UCB, EI and PI acquisition functions. We fixed the iteration number for the first experiment to see how using multiple acquisitions effect the performance when we compare it with single acquisition functions. As we can see in Fig. 1, the multi-objective acquisition approach gave better results than EI, PI and UCB alone for SMD1, SMD3 and SMD6. We can see that at the end of optimization by reaching the closer point to the optimal value for these problems. The proposed algorithm gave better performance than UCB and PI for the SMD2 problem but EI reached closer point to the optimal at the end. PI reached the best point at the end of iterations for SMD 4 and they are so close as it is the second best one.

In the second experiment, we can see from Table 3 the MACBP algorithm reached better results for SMD4, SMD5 and SMD7 than compared algorithms. For SMD1, we get closer to the optimal solution than NBLE and BIDE algorithms. We reached the better results for SMD2 when we compare the results with NBLE and BLEAQ algorithms. Comparing with BIDE and BLEAQ, the proposed algorithm get better results for SMD3. In terms of function evaluations, our MACBP algorithm decreased significantly the function evaluations as we can see at the Table 2 when we compare the other state-of-art algorithm in the literature.

6 Conclusion

In this paper, we proposed the MACBP algorithm, a Bayesian approach via multi-objective acquisition functions for bilevel optimization problems. We approached the leader's objective as an expensive black-box function. We used multiple acquisition functions during the bilevel optimization process, and made

our selection from a Pareto-front solution set in each iteration. We selected six popular SMD benchmark problems for the experiments. We compared our experimental results with a classic sequential setting of Bayesian optimization with each acquisition function performance individually. We also compare our results in terms of required function evaluations at the upper-level. It is shown that the proposed MACBP algorithm is competitive with existing well-known algorithms compared in the paper for solving bilevel optimization problems.

Acknowledgement. This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 16/RC/3918 which is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

1. Abo-Elnaga, Y., Nasr, S.: Modified evolutionary algorithm and chaotic search for bilevel programming problems. *Symmetry* **12** (2020). <https://doi.org/10.3390/SYM12050767>
2. Bard, J.F.: Coordination of a multidivisional organization through two levels of management. *Omega* **11**(5), 457–468 (1983)
3. Bard, J.F., Falk, J.E.: An explicit solution to the multi-level programming problem. *Comput. Oper. Res.* **9**(1), 77–100 (1982). [https://doi.org/10.1016/0305-0548\(82\)90007-7](https://doi.org/10.1016/0305-0548(82)90007-7)
4. Bard, J.F., Moore, J.T.: A branch and bound algorithm for the bilevel programming problem. *SIAM J. Sci. Stat. Comput.* **11**(2), 281–292 (1990). <https://doi.org/10.1137/0911017>
5. Bertinetto, L., Henriques, J.F., Torr, P., Vedaldi, A.: Meta-learning with differentiable closed-form solvers. In: International Conference on Learning Representations (2019). <https://openreview.net/forum?id=HyxnZh0ct7>
6. Bialas, W., Karwan, M.: On two-level optimization. *IEEE Trans. Autom. Control* **27**(1), 211–214 (1982). <https://doi.org/10.1109/TAC.1982.1102880>
7. Blank, J., Deb, K.: Pymoo: multi-objective optimization in python. *CoRR* abs/2002.04504 (2020). <https://arxiv.org/abs/2002.04504>
8. Bracken, J., McGill, J.T.: Mathematical programs with optimization problems in the constraints. *Oper. Res.* **21**(1), 37–44 (1973). <https://www.jstor.org/stable/169087>
9. Brotcorne, L., Labbé, M., Marcotte, P., Savard, G.: A bilevel model for toll optimization on a multicommodity transportation network. *Transp. Sci.* **35**, 345–358 (2001). <https://doi.org/10.1287/trsc.35.4.345.10433>
10. Brown, G., Carlyle, M., Diehl, D., Kline, J., Wood, R.: A two-sided optimization for theater ballistic missile defense. *Oper. Res.* **53**, 745–763 (2005). <https://doi.org/10.1287/opre.1050.0231>
11. Constantin, I., Florian, M.: Optimizing frequencies in a transit network: a nonlinear bi-level programming approach. *Int. Trans. Oper. Res.* **2**(2), 149–164 (1995)
12. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002). <https://doi.org/10.1109/4235.996017>

13. Feurer, M., Hutter, F.: Hyperparameter optimization. In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds.) *Automated Machine Learning*. TSSCML, pp. 3–33. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05318-5_1
14. Fontaine, P., Minner, S.: Benders decomposition for discrete-continuous linear bilevel problems with application to traffic network design. *Transp. Res. Part B: Methodol.* **70**(C), 163–172 (2014). <https://doi.org/10.1016/J.TRB.2014.09.007>. <https://ideas.repec.org/a/eee/transb/v70y2014icp163-172.html>
15. Franceschi, L., Frascioni, P., Salzo, S., Grazi, R., Pontil, M.: Bilevel programming for hyperparameter optimization and meta-learning. In: Dy, J., Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning*. *Proceedings of Machine Learning Research*, vol. 80, pp. 1568–1577. PMLR (2018). <https://proceedings.mlr.press/v80/franceschi18a.html>
16. Frazier, P.: A tutorial on Bayesian optimization. *arXiv abs/1807.02811* (2018)
17. Hansen, P., Jaumard, B., Savard, G.: New branch-and-bound rules for bilevel linear programming. *SIAM J. Sci. Stat. Comput.* **13**, 273 (1992). <https://doi.org/10.1137/0913069>
18. Hoffman, M., Brochu, E., de Freitas, N.: Portfolio allocation for Bayesian optimization. In: *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2011, Arlington, Virginia, USA*, pp. 327–336. AUAI Press (2011)
19. Hong, M., Wai, H.T., Wang, Z., Yang, Z.: A two-timescale framework for bilevel optimization: complexity analysis and application to actor-critic. *arXiv abs/2007.05170* (2020)
20. Islam, M.M., Singh, H.K., Ray, T., Sinha, A.: An enhanced memetic algorithm for single-objective bilevel optimization problems. *Evol. Comput.* **25**, 607–642 (2017). <https://doi.org/10.1162/EVCOa00198>
21. Ji, K., Yang, J., Liang, Y.: Bilevel optimization: convergence analysis and enhanced design. In: Meila, M., Zhang, T. (eds.) *Proceedings of the 38th International Conference on Machine Learning*. *Proceedings of Machine Learning Research*, vol. 139, pp. 4882–4892. PMLR (2021). <https://proceedings.mlr.press/v139/ji21c.html>
22. Johnson, M., Aghasadeghi, N., Bretl, T.: Inverse optimal control for deterministic continuous-time nonlinear systems. In: *52nd IEEE Conference on Decision and Control*, pp. 2906–2913 (2013). <https://doi.org/10.1109/CDC.2013.6760325>
23. Kieffer, E., Danoy, G., Bouvry, P., Nagih, A.: Bayesian optimization approach of general bi-level problems. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2017*, pp. 1614–1621. Association for Computing Machinery, New York (2017). <https://doi.org/10.1145/3067695.3082537>
24. Kirjner-Neto, C., Polak, E., Kiureghian, A.D.: An outer approximation approach to reliability-based optimal design of structures. *J. Optim. Theory Appl.* **98**(1), 1–16 (1998)
25. Koh, A.: Solving transportation bi-level programs with differential evolution. In: *2007 IEEE Congress on Evolutionary Computation*, pp. 2243–2250 (2007). <https://doi.org/10.1109/CEC.2007.4424750>
26. Konda, V., Tsitsiklis, J.: Actor-critic algorithms. In: Solla, S., Leen, T., Müller, K. (eds.) *Advances in Neural Information Processing Systems*, vol. 12. MIT Press (1999). <https://proceedings.neurips.cc/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf>
27. Koppe, M., Queyranne, M., Ryan, C.T.: Parametric integer programming algorithm for bilevel mixed integer programs. *J. Optim. Theory Appl.* **146**(1), 137–150 (2010). <https://doi.org/10.1007/S10957-010-9668-3>

28. Kraft, D.: A software package for sequential quadratic programming. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht, Wiss. Berichtswesen d. DFVLR (1988). <https://books.google.ie/books?id=4rKaGwAACAAJ>
29. Kunapuli, G., Bennett, K., Hu, J., Pang, J.S.: Classification model selection via bilevel programming. *Optim. Methods Softw.* **23**(4), 475–489 (2008)
30. Kushner, H.J.: A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *J. Basic Eng.* **86**, 97–106 (1963)
31. Legillon, F., Liefvooghe, A., Talbi, E.G.: Cobra: a cooperative coevolutionary algorithm for bi-level optimization. In: 2012 IEEE Congress on Evolutionary Computation, pp. 1–8 (2012). <https://doi.org/10.1109/CEC.2012.6256620>
32. Likhoshervostov, V., Song, X., Choromanski, K., Davis, J., Weller, A.: UFO-BLO: unbiased first-order bilevel optimization. arXiv abs/2006.03631 (2020)
33. Lyu, W., Yang, F., Yan, C., Zhou, D., Zeng, X.: Batch Bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In: Dy, J., Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 80, pp. 3306–3314. PMLR (2018). <https://proceedings.mlr.press/v80/lyu18a.html>
34. Ma, L., Wang, G.: A solving algorithm for nonlinear bilevel programming problems based on human evolutionary model. *Algorithms* **13**(10) (2020). <https://www.mdpi.com/1999-4893/13/10/260>
35. Migdalas, A.: Bilevel programming in traffic planning: models, methods and challenge. *J. Glob. Optim.* **7**, 381–405 (1995). <https://doi.org/10.1007/BF01099649>
36. Močkus, J.: On Bayesian methods for seeking the extremum. In: Marchuk, G.I. (ed.) *Optimization Techniques 1974: Optimization Techniques IFIP Technical Conference Novosibirsk. LNCS*, vol. 27, pp. 400–404. Springer, Heidelberg (1975). https://doi.org/10.1007/3-540-07165-2_55
37. Mombaur, K., Truong, A., Laumond, J.P.: From human to humanoid locomotion—an inverse optimal control approach. *Auton. Robots* **28**, 369–383 (2010). <https://doi.org/10.1007/s10514-009-9170-7>
38. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. CoRR abs/1912.01703 (2019). <https://arxiv.org/abs/1912.01703>
39. Rajeswaran, A., Finn, C., Kakade, S.M., Levine, S.: Meta-learning with implicit gradients. CoRR abs/1909.04630 (2019). <https://arxiv.org/abs/1909.04630>
40. Sabach, S., Shtern, S.: A first order method for solving convex bi-level optimization problems (2017). <https://doi.org/10.48550/ARXIV.1702.03999>
41. Sahin, K., Ciric, A.R.: A dual temperature simulated annealing approach for solving bilevel programming problems. *Comput. Chem. Eng.* **23**, 11–25 (1998)
42. Shaban, A., Cheng, C.A., Hatch, N., Boots, B.: Truncated back-propagation for bilevel optimization. CoRR abs/1810.10667 (2018). <https://arxiv.org/abs/1810.10667>
43. Sinha, A., Malo, P., Deb, K.: A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Trans. Evol. Comput.* (2017). <https://doi.org/10.1109/TEVC.2017.2712906>
44. Sinha, A., Malo, P., Deb, K.: Unconstrained scalable test problems for single-objective bilevel optimization. In: 2012 IEEE Congress on Evolutionary Computation, pp. 1–8 (2012). <https://doi.org/10.1109/CEC.2012.6256557>
45. Sinha, A., Malo, P., Deb, K.: An improved bilevel evolutionary algorithm based on quadratic approximations. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 1870–1877 (2014). <https://doi.org/10.1109/CEC.2014.6900391>

46. Sinha, A., Malo, P., Frantsev, A., Deb, K.: Multi-objective stackelberg game between a regulating authority and a mining company: a case study in environmental economics. In: 2013 IEEE Congress on Evolutionary Computation, pp. 478–485 (2013). <https://doi.org/10.1109/CEC.2013.6557607>
47. Sinha, A., Malo, P., Frantsev, A., Deb, K.: Finding optimal strategies in a multi-period multi-leader-follower stackelberg game using an evolutionary algorithm. *Comput. Oper. Res.* **41**, 374–385 (2014)
48. Smith, W.R., Missen, R.W.: Chemical reaction equilibrium analysis: theory and algorithms. In: *Chemical Reaction Equilibrium Analysis: Theory and Algorithms* (1982)
49. Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.W.: Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Trans. Inf. Theory* **58**(5), 3250–3265 (2012). <https://doi.org/10.1109/tit.2011.2182033>
50. von Stackelberg, H.: *The Theory of the Market Economy*. William Hodge (1952). <https://books.google.ie/books?id=fjIAtQEACAAJ>
51. Sun, H., Gao, Z., Wu, J.: A bi-level programming model and solution algorithm for the location of logistics distribution centers. *Appl. Math. Model.* **32**(4), 610–616 (2008)
52. Suryan, V., Sinha, A., Malo, P., Deb, K.: Handling inverse optimal control problems using evolutionary bilevel optimization. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 1893–1900 (2016). <https://doi.org/10.1109/CEC.2016.7744019>
53. Talbi, E.G.: A taxonomy of metaheuristics for bi-level optimization. In: Talbi, E.G. (ed.) *Metaheuristics for Bi-level Optimization*. SCI, vol. 482, pp. 1–39. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37838-6_1
54. Tutar, T., Filipic, B.: Demo: differential evolution for multiobjective optimization. In: *Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization*, Guanajuato, Mexico, pp. 520–533 (2005). https://doi.org/10.1007/978-3-540-31880-4_36
55. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007). <https://doi.org/10.1109/TEVC.2007.892759>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Explaining the Effects of Preprocessing on Constraint Satisfaction Search

Richard J. Wallace^(✉) 

Insight Centre for Data Analytics and Department of Computer Science,
University College Cork, Cork, Ireland
richard.wallace@insight-centre.org

Abstract. Preprocessing constraint satisfaction problems is a much studied method for improving the performance of subsequent solution search. The traditional explanation for its beneficial effects is “problem reduction”, where possible values that cannot take part in a solution are discarded, leaving fewer possibilities to explore during search. Here, we show that this is not the only or even the main factor when dynamic variable ordering heuristics are used. Multiple lines of evidence indicate that under these conditions domain reductions effected by preprocessing serve to inform the heuristic as to which variables should be chosen for instantiation before others. It is suggested that an information transmission model is needed to account for such effects, and it is argued that an extension of this approach can incorporate simple domain reduction effects as well.

Keywords: Constraint satisfaction · Preprocessing algorithm · Arc consistency · Neighbourhood singleton arc consistency

1 Introduction

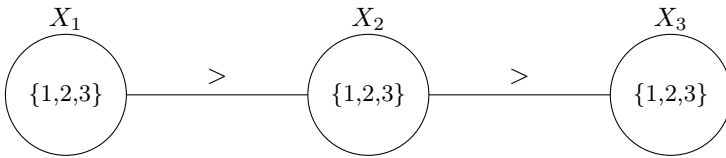
The study of constraint satisfaction problems (CSPs) has reached a point where there are deep formal analyses as well as a plethora of effective techniques for solving problems of this type. At the same time, there are many aspects of constraint solving that remain obscure to varying degrees. In part, this is because the combinatorial complexity of these problems gives them features that are hard to comprehend with the standard formal machinery that we have. Instead, we must often fall back on statistical analysis to even discern certain relationships.

But it can also happen that we simply fail to see the gaps in our explanations. An important example of such an oversight is the topic of the present paper.

In this case, the area of inquiry pertains to what are called local consistency techniques. It not too much to say that these methods form the core of constraint programming and serve to set it off from other approaches to the general problem of combinatorial optimization. Although they can be used in various contexts, a significant one, especially for more stringent forms of consistency, is the processing of a problem in order to simplify it prior to the actual search for a solution.

Local consistency techniques are polynomial-time algorithms that test for limited forms of consistency within portions of a problem [1]. The best-known example, which is the one to be considered here, is called “arc consistency”. This is described more carefully in the following section; here we can simply say that the algorithm tests whether a given value or label for a given variable in the problem is supported by each and every constraint that the variable is directly affected by. The thinking here is that if a value is not arc consistent, then it cannot be part of any complete set of labelings that satisfy all the constraints in the problem, aka a solution. Therefore, when we search for a solution, we don’t have to consider that value at all.

For example, consider the simple CSP depicted below, where there are three variables, X_1 , X_2 , and X_3 , represented by the circles (graph nodes) in the diagram below. Each variable can take one of three possible values (shown inside the circles). However, there are constraints between variables X_1 and X_2 and between X_2 and X_3 , indicated by the lines (graph edges) joining the circles, which specify conditions that must be met. Constraints, in turn, are associated with relations, indicated here by relational operators above the lines. In this case, the relations are that X_1 must be greater than X_2 , and the latter must be greater than X_3 .



Now, given these constraints, there are values in the domains of some variables that cannot appear in any solution. This can be shown by checking values in the adjacent domain to see if anything can go with (i.e. support) a given value. For example, value 1 in the domain of X_1 is not greater than any value in the domain of X_2 ; hence, it cannot be a part of any solution, so it can be removed from the problem without any solution being lost. By using this strategy repeatedly, we can reduce the number of values that we have to consider when we search for a solution. In this case we can reduce each domain to a single value, which gives us the unique solution to this problem immediately.

In most accounts of constraint solving, it is assumed that this is the (sole) reason that preprocessing is useful. That is, it reduces the number of alternatives that one must examine during search. Another way of putting this is that preprocessing reduces the search space.

So in a well-known textbook on the subject [9], we find this statement:

“Problem reduction techniques transform CSPs to equivalent but hopefully easier problems by reducing the size of the domains and constraints in the problems”. (p. 33)

And later on:

“The following are possible gains from problem reduction when combined with searching: (1) Reducing the search space. Since the size of the search space is measured by the grand product of all the domain sizes in the problem, problem reduction can help to reduce the search space by reducing the domain sizes”. (pp. 79–80)

A similar account of the aims of local consistency preprocessing is found in another important reference work [4]:

“In general, inference, as it is applied to constraints, narrows the search space of possible partial solutions by creating equivalent, yet more explicit networks”. (p. 52)

In the present paper we will argue that this is not the full story. In fact, for problems of any difficulty (in contrast to the example above), this is not the chief benefit of preprocessing. For strategies that are sensitive to the current state of the problem, and in particular the state achieved by preprocessing, the latter can impart critical information that allows the search process to operate much more efficiently than it otherwise could.

In the remainder of the paper, we will explicate the last statement and demonstrate it experimentally. Section 2 provides necessary background in the form of basic concepts and definitions. Section 3 describes the algorithms used in this paper. Section 4 describes the experimental environment used in this work, including types of CSPs used. Sections 5 and 6 present the basic argument of the paper together with supporting evidence. Section 7 discusses a related finding that bears on the present thesis. Section 8 considers how the communication process by which preprocessing affects subsequent search might be modelled. Section 9 summarizes work to date and indicates what still must be done to better understand this critical aspect of constraint solving.

2 Basic Concepts

Here, we review some of the basic concepts in the field that form the background to the present work.

A constraint satisfaction problem (CSP) involves assigning values to a set of variables subject to restrictions on the way that values can go together. More formally, a CSP can be defined as a tuple, (X, D, C) where:

X is a set of *variables*, X_1, \dots, X_n ,

D is a set of *domains*, D_i , where each D_i is a set of possible *values* for variable X_i

C is a set of *constraints*. Each C_i belonging to C consists of a relation R_i and a particular subset of the variables in X , called the *scope* of the constraint. R_i is based on the Cartesian product of the values of the domains of the variables in the scope. (In the introductory example, one constraint has scope $\{X_1, X_2\}$,

and its relation is based on the Cartesian product of domains D_1 and D_2 i.e. the pairs in $\{(1, 1), (1, 2) \dots (3, 2), (3, 3)\}$ where the first element is greater than the second).

A *solution* to a CSP is an assignment or mapping from variables to values, $A = \{(X_1, a), (X_2, b), \dots, (X_k, x)\}$, that includes all variables ($k = n$) and does not violate any constraint in C .

CSPs have an important monotonicity property in that inconsistency with respect to even one constraint implies inconsistency with respect to the entire problem. This has given rise to methods for filtering out values that cannot participate in a solution, based on local inconsistencies, i.e. inconsistencies with respect to subsets of constraints. By doing this, these algorithms establish well-defined forms of local consistency in a problem.

The most widely used methods establish *arc consistency*. In problems with binary constraints, arc consistency (AC) refers to the property that for every value a in the domain of variable X_i and for every constraint C_{ij} involving X_i there is at least one value b in the domain of X_j such that (a, b) satisfies that constraint. For non-binary constraints, this requirement is extended to include all the other variables in the constraint, i.e. for every value a , there must be one tuple of values in the relation associated with the constraint that includes a assigned to X_i .

The present paper also makes use of certain forms of *singleton arc consistency*, or SAC [2, 3, 10]. This is a form of AC in which the just-mentioned value a , for example, is considered the sole representative of the domain of X_i . If AC cannot be established for the entire problem under this condition, then there can be no solution with this value, so a can be discarded. If this condition can be established for all values in problem P , then the problem is singleton arc consistent. (Obviously, SAC implies AC, but not vice versa).

A closely related form of consistency is called *neighbourhood singleton arc consistency*, or NSAC [11]. NSAC algorithms establish SAC with respect to the neighbourhood of the variable whose domain is a singleton as opposed to the entire problem.

Definition 1. The *neighbourhood* of a variable X_i is the set $X_N \subseteq X$ of all variables in all constraints whose scope includes X_i , excluding X_i itself. Variables belonging to X_N are called the neighbours of X_i .

If for each value $a \in D_i$, where i is in $\{1 \dots n\}$, singleton arc consistency can be established in the subgraph based on that variable and its neighbours, then the problem is neighbourhood singleton arc consistent.

3 Arc Consistency and (N)SAC Algorithms

Pseudocode for a standard arc consistency algorithm is shown in Fig. 1. Basically, the algorithm considers each value in each domain in turn, and for each adjacent constraint, it determines whether that value has a supporting value in the domain

```

Procedure AC-3
1   Q ← X
2   OK ← true
3   While OK and not empty-Q
4       Select and remove  $X_i$  from Q
5       Foreach constraint  $C_{ij}$  that includes  $X_i$ 
6           Changeflag ← false
7           Foreach value  $a \in \text{domain of } X_i$ 
8               If there is no value  $b \in \text{domain of } X_j$  that supports  $a$ 
9                   Remove  $a$  from domain of  $X_i$ 
10                  Set Changeflag to true
11          If domain of  $X_i$  is empty
12              OK ← false
13          If OK and Changeflag is true
14              Update Q to include all variables adjacent to  $X$  except  $X_j$ 

```

Fig. 1. Pseudocode for AC-3, a standard AC algorithm, for binary CSPs.

of the adjacent variable (i.e. the other variable in the constraint). It continues to do this until all values have been checked as many times as they need to be, which may involve re-checking values if some of their neighbouring domains have changed. If a problem is not arc consistent, then eventually some domain of values will become empty. (This is called a domain “wipeout”).

The algorithm shown is of the type called “AC-3” [6]. All AC-3 style algorithms use a single queue (although the queue may take different forms) and queue updating procedures as shown in Fig. 1. The version shown here is for problems with binary constraints, but the AC-3 algorithm can be extended to k -ary constraints as well; in this case, the test is: for constraint C and for a given value v in the domain of X_i , is there is an k -tuple in C in which v is assigned to X_i ?

SAC and NSAC use AC as a building block. The basic idea is to reduce the domain of a given variable X_i to a single value v , and then determine whether the problem or a subproblem that includes X_i is consistent under these conditions. If it isn’t consistent, then v can be removed from the problem without losing any solutions, since if it can’t support AC, it can’t support any fully consistent solution to the problem.

Figure 2 shows a state-of-the-art NSAC algorithm [12]. This algorithm uses an AC-3 style queue similar to the arc consistency algorithm, except that at each step, the algorithm attempts to establish AC for the neighbourhood subproblem. In addition, if a value ‘fails’, i.e. at a given step, wipeout occurs, then after removing that value, simple AC is performed for the entire problem (line 10 in the figure). In addition, whenever a value is removed the queue must be updated (lines 12 and 14 in the figure). (The corresponding SAC algorithm is very similar, except that on line 7 AC is established for the entire problem and on line 14 the queue is updated to include all variables in the problem).

```

Procedure NSACQI
1   Q ← X
2   AC-OK ← AC(P)
3   While AC-OK and not empty-Q
4       Select and remove  $X_i$  from Q
5       Foreach  $v_j \in \text{dom}(X_i)$ 
6            $\text{dom}'(X_i) \leftarrow \{v_j\}$ 
7           If AC( $X_i + \text{neighbours}(X_i)$ ) leads to wipeout
8                $\text{dom}(X_i) \leftarrow \text{dom}(X_i) \setminus v_j$ 
9           If AC(P) leads to wipeout
10              AC-OK ← false
11          Elseif domains have changed during AC
12              Update Q to include all neighbours of variables with changed domains
13      If domain of  $X_i$  has changed
14          Update Q to include all neighbours of  $X_i$ 

```

Fig. 2. Pseudocode for NSACQ with interleaved AC. P is always the full problem in its current state.

The search algorithm used in this paper is called maintained arc consistency (MAC) [7]. In this version, after preprocessing, MAC performs a backtrack-style search in which a partial solution is extended by choosing a variable and then a value to assign to that variable that is consistent with all the assignments already chosen. Following each new assignment, MAC also establishes AC in the subproblem formed by the variables not yet given an assignment. If this AC fails, then the last value assigned is retracted; if there are no more values to test for the last variable chosen, then search backs up and tries another value for the next-to-last variable chosen. This process continues until either a complete set of consistent assignments is found (aka a solution) or there are no more values to test at the highest level, which means there are no solutions.

These algorithms will be used to analyze the effects of preprocessing on search. The following empirical studies will attempt to separate problem reduction from other effects that occur when some domain values are eliminated.

4 Experimental Methods

All algorithms were implemented in Common Lisp. Experiments were run using Macintosh Common Lisp (MCL) version 5.1 on an iMAC (MAC OS X version 10.2.8) with a Power PC 800 MHz CPU. Search was carried out using a form of MAC called MAC-3 using different *variable ordering heuristics* for selecting the next variable to assign a value to during search. After a variable was chosen, values in its domain were tested in lexical order (i.e. 1, 2, 3 ...).

One variable ordering heuristic used in these experiments, called minimum domain over forward degree (min d/fd), chooses the variable with the lowest ratio of domain size (number of domain values) to forward degree. The latter is defined as the number of constraints that a variable has with variables that have not yet been assigned a value (hence, the adjective “forward”). The chief

characteristic of this heuristic is that it is *dynamic*, i.e. it takes account of the problem representation at each stage of search. Therefore, the domain size used is the current size of the domain, which will have been reduced during search to those values consistent with prior assignments. Similarly, the value of the forward degree will depend on which variables sharing a constraint with a given variable have not yet been assigned a value.

The other variable ordering heuristic that was used is the maximum static degree heuristic. This chooses variables on the basis of their degree in the constraint graph (i.e. the number of constraints a variable is involved in), choosing the unassigned variable with the highest degree.

For purposes of demonstration, the following types of problems were utilized. “Geometric problems” are generated by selecting points at random within the unit square to represent variables, and adding constraints between those whose Euclidean distance is less than some criterion, called the ‘distance’ [5]. In the present work, in addition, if there is more than one connected component, separate components are connected via pairs of variables (one in each component) having the smallest Euclidean distance between their points in the unit square, in order to make a graph with a single connected component. The present sets of problems had binary constraints, although the scheme can be extended to produce constraints with any number of variables.

Four different problem sizes were tested (here, size = number of variables): 120, 80, 40, and 25. Problems in these sets had identical domain sizes of 20, 15, 12 and 15 values, respectively. To limit variation in graph density (proportion of possible edges), only problems within a small range around a “target” were accepted; for example, for 120-variable problems, the target was 540 constraints (± 3), giving a graph density of 0.076.

With geometric problems, constraint relations are generated according to a random scheme for selecting supports. In the problems used in these experiments, domain values had either of two levels of support within each constraint. As a result, values were generally well supported (low tightness), but occasionally they were not (high tightness). For 120-variable problems, a constraint tightness value of 0.3 was selected with a probability of 0.8; otherwise, a constraint tightness of 0.7 was chosen. For other problem sizes, the latter tightness value was 0.75. In the remainder of the paper, these problems will be referred to as “geovarsat” problems.

The second set of problems were a kind of benchmark problem known as Radio Link Frequency Allocation Problems (RLFAPs). These are also binary problems, and have two kinds of distance (difference) constraint, meaning that the difference between two assignments must satisfy a numerical relation. The two forms of distance constraint are $|v(X_i) - v(X_j)| = k$ and $|v(X_i) - v(X_j)| > k$ where X_i and X_j are variables, v is an operator that assigns a value to a variable, and k is some integer value. These problems have constraint graphs of low density and domain sizes of about 40. There is also considerable structure to the pattern of constraints and the relation of the k values to the values in the domains. For the present experiments, in order to obtain a larger sample of problems with the

same basic characteristics, the following method was used. A single benchmark problem was used (known as `rlfap-graph3`), having 200 variables. From this, a set of problems was generated by taking the benchmark problem and randomly choosing ten percent of the $|v(X_i) - v(X_j)| > k$ constraints and altering them by either incrementing or decrementing the value of k by 10. (The decision to increment or decrement was decided randomly, each decision occurring with equal probability).

For all problem sets, a large number of problems were generated that were filtered for solutions (in a generate-and-test fashion). From each set, the 25 hardest problems were chosen for use in the present experiments.

In these experiments, search was discontinued if a cutoff was reached, which was one million search nodes for geovarsat problems and one hundred thousand nodes for RLFAPs; in these cases the search node number was recorded as having the cutoff value.

5 Effects of Preprocessing on Geovarsat Problems

5.1 Results

Results for geovarsat problems of all sizes using the dynamic min d/fd heuristic are shown in Table 1. In this experiment, three different levels of consistency were established prior to search: AC, NSAC, and SAC. Successively higher levels of consistency were associated with successively larger numbers of deleted values, and, on average, successively smaller search trees. For these problems, NSAC deleted about five to ten times as many values as AC, while the difference between NSAC and SAC was sometimes almost double and sometimes just a few percentage points.

Table 2 gives results with min d/fd and max degree for the three smaller problem sizes. Note, first of all that differences in search with max static degree are generally more modest than with min d/fd. Thus for min d/fd, the mean reduction in search nodes after NSAC preprocessing, compared to AC, was 30%, 63%, and 45% for the 80-, 40- and 25-variable problems, respectively, while the corresponding values for max static degree were 23%, 46%, and 11%.

Moreover, when one looks at the data for individual problems, one finds that for d/fd there are occasional cases where following application of a higher level of consistency, search is actually somewhat worse (more search nodes). For the 80-variable problems there were two such cases when NSAC was compared to AC, and eight cases when SAC was compared to NSAC, and there were occasional cases for the two smaller problem sizes as well. In contrast, there were no such cases when the max static degree heuristic was used.

We can go further and consider the relation between the domain reduction after NSAC and AC and the search reduction following these two preprocessing algorithms. This was done by taking the ratio, $\frac{AC - NSAC}{AC}$, where AC and NSAC stand for search nodes following AC and NSAC, respectively. A similar

Table 1. Preprocessing and Search With Different Forms of Local Consistency (Geometric Problems, Varying Satisfiability)

problem size	AC		NSAC		SAC	
	Delete	Nodes	Delete	Nodes	Delete	Nodes
120	40	>376,457	229	2915	391	818
80	14	49,075	66	34,500	81	11,921
40	12	335	126	125	211	87
25	4	187	21	102	25	96

Notes. Means of 25 problems. min domain/forward degree variable ordering heuristic. “size” is number of variables. “delete” is number of values deleted. “nodes” is search nodes. “>” indicates cutoff reached on some searches.

Table 2. Search with Different Forms of Local Consistency And Different Variable Ordering Heuristics (Geovarsat Problems)

Heuristic	AC nodes	NSAC nodes	SAC nodes
80-variable probs			
Min d/fd	49,075	34,500	11,921
Max degree	86,379	66,501	58,440
40-variable probs			
Min d/fd	335	125	87
Max degree	334	181	84
25-variable probs			
Min d/fd	187	102	96
max degree	196	174	169

ratio, $\frac{\text{NSAC} - \text{AC}}{\text{NSAC}}$ was used for total deletions after NSAC and AC. Pearson Product Moment Correlations were derived after transforming these ratios using the arcsin function [8]. The results are shown in Table 3.

Table 3. Correlations between Proportional Values of Search and Domain Reduction Following AC and NSAC (Geovarsat Problems)

Probs	d/fd vs dels	mxdeg vs dels	d/fd vs mxdeg
80 vars	0.40*	0.68 ⁺	0.20
40 vars	0.45*	0.83 ⁺	0.63 ⁺
25 vars	0.07	0.67 ⁺	0.26

Notes. * $p < .05$, ⁺ $p < .01$, two-tailed.

For all three problem sets, correlations between search reduction and domain reduction were always higher for the max degree heuristic, sometime appreciably so. For one problem set, the correlation for search with d/fd and domain reduction was not even statistically significant. (In fact, it was close to zero). It is also interesting that in two cases the correlation for search reduction with the two heuristics was non-significant.

If one looks at individual problems, the difference is even clearer. To demonstrate this, five problems were chosen at random from the 80-variable set using digit pairs from a random number table; their search tree sizes are shown in Table 4 for each heuristic after preprocessing with either AC or NSAC.

Table 4. Search with different heuristics following AC and NSAC (individual geovarsat problems)

prob#	d/fd		mx dg	
	AC	NSAC	AC	NSAC
11	4577	331	2881	1938
23	1358	1868	1955	1897
54	4446	187	519	374
93	17,213	15,549	835,133	649,133
96	4955	711	8924	8903

Notes. Individual problems chosen at random from 80-variable problem set.

Here, we see that for each problem, search with max degree shows a modest reduction following NSAC in comparison with AC. In contrast, search with d/fd can show modest improvement, drastic improvement, or even worse performance when NSAC is used rather than AC.

Because of space restrictions, results with RLFAPs will not be discussed in detail. However, the same pattern of differences was found, both for the mean results and the correlational analysis.

5.2 Discussion

These results show that when search involves dynamic ordering heuristics, the effects of preprocessing involve other factors than domain reduction, although that of course plays a part. Both order-of-magnitude improvements and, even more decisively, deterioration in performance are inexplicable on this basis. (Note that neither of these are found with a non-dynamic heuristic like max degree). Moreover, the correlational analysis yields one case where there is almost a zero correlation between search with d/fd and the amount of domain reduction; yet NSAC still leads to improvement in search on average in this case (Tables 1 and 2).

Since there is a close relation between the dynamic character of the heuristic and these ‘anomalous’ effects, an obvious hypothesis is that domain reductions are serving to guide subsequent search in making good heuristic choices of the next variable to assign a value. The following section gives further evidence that this explanation is the correct one.

6 Further Evidence Using Restricted NSAC Testing

During the course of work on the restricted application of NSAC during preprocessing [13], an effect was found that bears on the present issue. In this study, NSAC was performed according to a scheme in which if a randomly generated number between 0 and 1 was less than some criterion, then NSAC was performed; otherwise, the value was skipped over. Otherwise, the procedure was the same as that shown in Fig. 2.

Figure 3 shows search results for one RLFAP problem over 50 separate runs with each of several criterial values (shown under the abscissa). These are frequency distributions of the number of nodes required (i.e. the search effort) for each run for this problem using the d/fd heuristic. (With this problem, using the d/fd heuristic, search after AC preprocessing stopped upon reaching the one hundred thousand node cutoff, while following full NSAC a solution was found after 2205 nodes).

Note that, with a very low probability of performing NSAC, most runs are very bad (in the 10^4 or 10^5 ranges, 10^5 being the cutoff). Then, for slightly higher probabilities, most runs fall in the 10^2 range. In fact, for $p = 0.025$ or 0.05 , most runs required fewer than 211 nodes (200 being a perfect, retraction-free run). This is much better than after full NSAC. For still higher probabilities, there is a shift so that there are increasingly more frequent runs requiring 10^3 nodes.

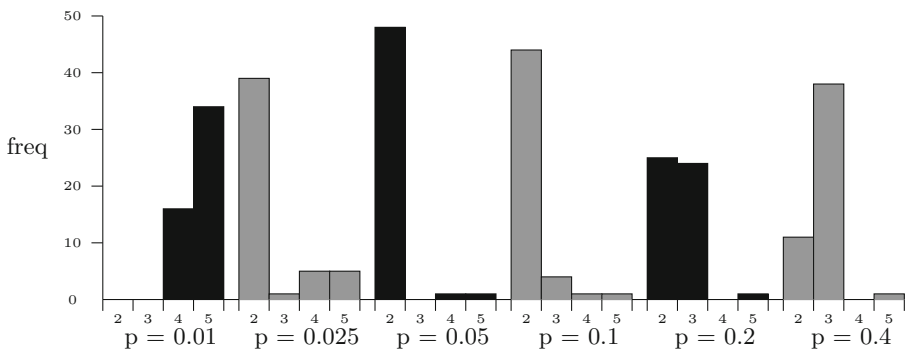


Fig. 3. Frequency histograms for six probabilities of performing NSAC on any given value. RLFAP problem # 1. Log scale on abscissa, expressed as characteristics of the logarithm; e.g. 2 is for total search nodes between 100 and 999, inclusive. Total range is from 100 to over 100 thousand. Successive distributions coloured black or gray to distinguish them.

Again, we see effects on search performance that bear no relation to the degree of problem reduction, which increases monotonically as p increases. Although the full explanation has not yet been worked out, the most reasonable (general) hypothesis is that for this problem, a small proportion of NSAC-based deletions guides the dynamic search heuristic very effectively on the majority of runs.

7 Related Work

Early in the 1990s, there were reports of occasional anomalies in search after preprocessing. In one case this concerned the forward checking algorithm, where backtrack search is interleaved with a more limited form of consistency checking than the full arc consistency that occurs with MAC. When forward checking is used with the minimum domain heuristic (another dynamic variable ordering heuristic), AC prior to search can sometimes result in much worse performance than forward checking alone [7]. Perhaps because this effect was not observed with MAC, this observation has been disregarded for the most part. However, it seems to be another case of domain reduction during preprocessing affecting the choices made by a dynamic variable ordering heuristic.

8 Modelling Preprocessing Effects on Search

The basic finding here is that domain reductions caused by preprocessing affect heuristics for choosing variables made during search when the latter take domain size into consideration. Most often this results in major improvements, but it can also be detrimental. What this means in effect is that preprocessing allows information to be communicated to the search process that, in turn, allows the latter to make better selections from among a set of alternatives.

If there is an information channel, then it would be helpful to characterize the amount of information transmitted, or at least the maximum number of alternatives that can be encoded. Since in this case we are considering the effects of domain reduction, then the maximum number of distinguishable alternatives is $O(d_{max})$, where d_{max} is the maximum domain size in the original problem. (Note that the alternatives we are considering here are different domain sizes).

However, there are two aspects to this. Most simply, the information from preprocessing allows the heuristic to distinguish among choices (variables). In this respect, there is no noise, i.e. no confusion between alternatives. But in the present situation, choices have implications, or further effects, and what we really want are choices that reduce search effort. If we consider an optimal search order as one that results in the smallest possible search tree among all possible orderings, then the information may allow search to better approximate that optimal ordering. In this respect, the information channel is ‘noisy’, in that a given ‘signal’ can fail to improve search efficiency or even lead to greater search effort rather than less.

If we consider not only variable but also value selection, we may be able to incorporate domain reduction into the same framework, since preprocessing is informing the search process that certain values should not be tried. In this case, the alternative signals could be considered the possible variable-value pairs, rather than the possible variables.

9 Summary and Conclusions

In this paper we have demonstrated that the classical view of preprocessing in the context of constraint satisfaction search is inadequate, and we have sketched out a more general model of the relation between preprocessing and subsequent search that appears to encompass all of the effects of the former on the latter. This more general model may, in turn, allow us to design and deploy preprocessing algorithms more intelligently, in order to avoid the pitfalls that can arise from insufficient information or even misinformation in the form of misleading signals to decision making processes that occur during search.

Although differences observed in these experiments may have been partly the result of selecting harder problems, this does not affect the basic conclusions regarding the effects of preprocessing on solution search. In fact, it is likely that differences related to preprocessing will be greater when problems are more difficult simply because of a ceiling effect. Hence, using harder problems to demonstrate preprocessing effects is probably a good research strategy. However, since improvement in search due to dynamic variable ordering heuristics is a general phenomenon, the differences observed here should occur regardless of the overall difficulty of the problems.

It seems quite likely that the information transmission conception outlined in this paper has wider application than constraint satisfaction search. Presumably it applies whenever there are heuristic decisions to be made by an algorithmic process and where there is the possibility of altering the problem in order to make the search for a solution more efficient. However, such possible extensions remain for future work.

References

1. Bessière, C.: Constraint propagation. In: *Handbook of Constraint Programming*, chap. 3, pp. 29–83. Elsevier (2006)
2. Bessière, C., Cardon, S., Debruyne, R., Lecoutre, C.: Efficient algorithms for singleton arc consistency. *Constraints* **16**, 25–53 (2011)
3. Debruyne, R., Bessière, C.: Some practicable filtering techniques for the constraint satisfaction problem. In: *Fifteenth International Joint Conference on Artificial Intelligence - IJCAI 1997*, vol. 1, pp. 412–417. Morgan Kaufmann (1997)
4. Dechter, R.: *Constraint Processing*. Morgan Kaufmann, Burlington (2003)
5. Johnson, D.S., Aragon, C.R., McGeoch, L.A., Shevron, C.: Optimization by simulated annealing: an experimental evaluation. Part II. Graph coloring and number partitioning. *Oper. Res.* **39**, 378–406 (1991)

6. Mackworth, A.: Consistency in networks of relations. *Artif. Intell.* **8**(1), 99–118 (1977)
7. Sabin, D., Freuder, E.: Contradicting conventional wisdom in constraint satisfaction. In: *Proceedings of the Eleventh European Conference on Artificial Intelligence-ECAI 1994*, pp. 125–129. Wiley (1994)
8. Snedecor, G.W., Cochran, W.G.: *Statistical Methods*, 7th edn. Iowa State University, Ames (1980)
9. Tsang, E.: *Foundations of Constraint Satisfaction*. Academic Press (1993)
10. Wallace, R.J.: Light-weight versus heavy-weight algorithms for SAC and neighbourhood SAC. In: Russell, I., Eberle, W. (eds.) *Twenty-Eighth International Florida Artificial Intelligence Research Society Conference - FLAIRS-28*, pp. 91–96. AAAI Press (2015)
11. Wallace, R.J.: SAC and neighbourhood SAC. *AI Commun.* **28**, 345–364 (2015)
12. Wallace, Richard J.: Interleaving levels of consistency enforcement for singleton arc consistency in CSPs, with a new best (N)SAC algorithm. In: Baldoni, Matteo, Bandini, Stefania (eds.) *AIxIA 2020. LNCS (LNAI)*, vol. 12414, pp. 301–317. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77091-4_19
13. Wallace, R.J.: Experimental analysis of restricted forms of SAC based reasoning for constraint satisfaction problems: preliminary results. In: Maratea, M., Vallati, M. (eds.) *Twenty-ninth RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion - RCRA 2022* (2022)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Variable-Relationship Guided LNS for the Car Sequencing Problem

Filipe Souza^{1,2(✉)}, Diarmuid Grimes^{2,3}, and Barry O’Sullivan^{1,2}

¹ Insight SFI Research Centre for Data Analytics, University College Cork,
Cork, Ireland

f.desouza@cs.ucc.ie

² SFI Centre for Research Training in Artificial Intelligence, Cork, Ireland

³ Munster Technological University, Cork, Ireland

<http://www.ucc.ie/>, <http://www.crt-ai.cs.ucc.ie>, <http://www.mtu.ie/>

Abstract. Large Neighbourhood Search (LNS) is a powerful technique that applies the “divide and conquer” principle to boost the performance of solvers on large scale Combinatorial Optimization Problems. In this paper we consider one of the main hindrances to the LNS popularity, namely the requirement of an expert to define a problem specific neighbourhood. We present an approach that learns from problem structure and search performance in order to generate neighbourhoods that can match the performance of domain specific heuristics developed by an expert. Furthermore, we present a new objective function for the optimization version of the Car Sequencing Problem, that better distinguishes solution quality.

Empirical results on public instances demonstrate the effectiveness of our approach against both a domain specific heuristic and state-of-the-art generic approaches.

Keywords: LNS · Neighbourhood selection · Car sequencing problem

1 Introduction

Large Neighbourhood Search (LNS) [14] is a powerful technique to tackle Combinatorial Optimisations Problems, but its main drawback remains on the necessity of an expert to refine the algorithm components for the specific behaviour of each problem. One of the most crucial components is the neighbourhood selection approach, which is highly sensitive to the characteristics of the given problem. Thus, an important open research question concerns developing generic neighborhood selection heuristics that can be efficient in a broad range of problems. Even though it is hard to imagine that a generic approach can overcome a domain specific heuristic designed accurately by an expert, generic approaches

Supported by SFI Centre for Research Training in Artificial Intelligence under Grant No. 18/CRT/6223 and SFI under Grant No. 12/RC/2289-P2, co-funded under the European Regional Development Fund.

© The Author(s) 2023

L. Longo and R. O’Reilly (Eds.): AICS 2022, CCIS 1662, pp. 437–449, 2023.

https://doi.org/10.1007/978-3-031-26438-2_34

have an essential role to popularize LNS as one of the most powerful technique to solve a broadly range of complex large-scale Combinatorial Optimisations Problems (COP).

Freuder and O’Sullivan [3] proposed a number of grand challenges for Constraint Programming (CP). A common aspect amongst these challenges is the requirement for approaches that can solve a range of different problems without the need of a human expert to fine tune its parameters or design dedicated algorithms. Following this line, in this paper we present a novel constraint-based Large Neighbourhood Search that learns from problem structure and search performance in order to create complex and diverse neighbourhoods without the need of a domain specific algorithm. We hypothesise that good neighborhoods can be identified through combining information regarding the problem structure with information learnt during search.

2 Related Work

Large neighborhood search, Fig. 1, was first proposed by Shaw in 1998 [14] as a means of applying CP techniques to large vehicle routing problems. In its basic form, an initial solution is generated and then refined in successive iterations. Each iteration involves firstly the selection of a subset of variables (the neighborhood), whose assignment is relaxed while all other variables have their assignment fixed to the value in the current solution. The neighborhood of unassigned variables can then be solved using a systematic approach, like CP or MIP, to find the optimal solution to the neighborhood given the assignment of the non-neighborhood variables. A key aspect, as highlighted in the figure, is how the neighborhood is selected in each iteration.

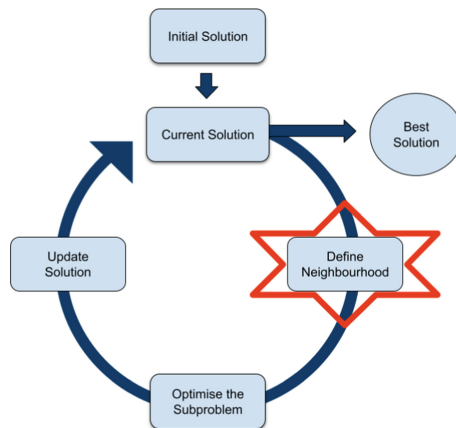


Fig. 1. Large neighbourhood search

2.1 LNS: Domain Independent Neighbourhood Heuristic

A large number of dedicated neighborhood operators have been proposed for different problems. Our focus in this work concerns approaches which have tried to create more generic LNS neighborhood selection methods. To date, much of the literature in this area has focused on portfolio approaches to automatically define the best neighbourhood selection heuristic from a predefined list. For example Laborie and Godard [6] proposed to tackle 21 variations of Single-Mode Scheduling Problems by applying a reinforcement learning method to select, in each iteration, the most suitable neighbourhood selection heuristic for a given instance from a predefined portfolio. The main drawback of this approach can be observed when heuristics from the portfolio have different run-times. Under these circumstances the heuristics with smaller run-time will be used more often as the reward function is given by $\Delta c / \Delta t$ i.e. the size of improvement in objective value divided by the run-time to achieve that improvement. Typically large improvements can be made earlier on while as we approach the optimal solution the improvements are smaller.

In order to address this drawback, Thomas and Schaus [16] proposed a new weight-update mechanism for the portfolio approach. This mechanism works by evaluating the neighborhood heuristic based on its performances obtained in an evaluation window which starts on β iterations before last improvement until the current iteration. That way the windows will always keep information from a fixed part of the search before any stagnation. Even though this approach proved its efficiency on a broad range of problems, we believe that there are two points of further investigation. Firstly, in the results presented in [16], we observe that the random neighbourhood selection performs well in a wide range of problems, even when compared to more sophisticated approaches (we will elaborate on this in the following section). Secondly, these approaches are highly dependent on the list of neighbourhood selection heuristics in the portfolio, thus they cannot be fully classified as domain independent approaches.

To the best of our knowledge the first effective domain independent approach was the Propagation Guided Large Neighbourhood Search (PGLNS) [11]. Here Perron et al. proposed choosing the neighborhood variables based on analysis of the impact of each frozen variable in turn. They tested a number of configurations, however the approach that performed better than the domain specific neighbourhood selection heuristic Interval-Based [10] was a configuration that alternated between three neighborhood selection methods. In one, neighbourhoods are created by starting from an empty solution and incrementally freezing variables based on the propagation impact until achieving the desire neighbourhood size. The second neighbourhood method built by starting from a complete solution and incrementally relaxing variables based on the propagation impact until achieving the desire neighbourhood size. The final neighborhood was generated randomly. The first two approaches are highly efficient to learn from variable relationships, but they do not use information about the variables behaviours during the search process, which we believe to be highly beneficial to generate better neighbourhoods. For instance, when a variable is already assigned to its

optimal value, or already has been selected many times, there is no reason to keep selecting this variable only because it has a strong relationship with other variables.

On the other hand there are some domain independent approaches that focus only on variables behaviours aspects, and do not consider the structural relationship between variables. Carchrae and Beck in [2] proposed a Cost-based method to select neighbourhoods, where the variable impact on the overall objective function is the main component to select the variables that will compose the neighbourhood. Their results demonstrated the importance of a stochastic element to ensure a high variety of neighbourhoods, mainly when the instance problem is not so large. Lombardi and Schaus [9] also proposed a heuristic that relies on the cost impact capability of the variables. Their calculation of cost impact is based on lower bound cost before and after assigning the variable its value from the current best solution during a permutation of orderings of the variables. These impacts are then used to weight a roulette wheel style selections strategy for the neighborhood operator.

2.2 Exploration vs Exploitation on Neighbourhood Selection

Many LNS approaches in the literature have reported impressive performance of the simple random neighbourhood selection method, even when we compare it with more sophisticated heuristics, [2, 11, 12, 16]. A highly deterministic approach may choose very similar neighborhoods multiples times resulting in a huge computational time spent on neighbourhoods that do not have as much capacity for improvement; while ignoring some parts of the search space. On the other hand, a complete stochastic approach has a poor exploration of any knowledge from the problem, variables and their connections, but if the neighbourhoods are relatively large, the likelihood of selecting a small number of connected variables where lies some improvement is considerable.

3 Problem Definition

The Car Sequencing problem was originally defined as a Constraint Satisfaction Problem (CSP) [1, 15] that aims to allocate a set of cars on a production line of options' installation over a fixed number of timeslots (e.g. one day of timeslots). Each bay has its own capacity, i.e. the number of cars they can work on in a segment of the production line. Furthermore each bay can install only one type of option.

In order to transform this problem to a Constraint Optimisation Problem (COP), we add a new class of car where no option is needed, similar to [10, 11]. They used the concept of empty slots providing buffers which are then to be minimised, i.e. minimise the number of extra time slots needed to allocate all cars. The novelty in our formulation is the use of the number of options not placed on the original production line as objective function, with the logic that cars with fewer option requirements would be easier to slot in on a subsequent day.

This approach allows the search to distinguish between two partial solutions even when both have the same number of original cars placed on the production line by prioritising the solution where the placed cars have more options installed, since the cars with less options installed are more likely to find a place in the following iterations.

The problem can be more formally defined as follows:

Definition 1 (Option). *An option $o \in O$ is an extra item to be installed on some specific configuration of a given car, e.g. Parking Assist, Speed Limit Assist, Air Conditioning.*

Option o is characterized by: the window size, WS_o , on the production line; and the maximum number of the option, MC_o , that can be installed in the window.

Definition 2 (Configuration). *A configuration $c \in C$ is a version of a car with a particular set of options. c is characterized by: the number of needed cars $cars_c$; and $REQ_{c,o} \forall o \in O$ that defines whether an option is required by the configuration.*

Definition 3 (Position). *A position $p \in P$ is a place in the queue of the car production line.*

Definition 4 (Solution). *A solution S is an assignment of $\forall c \in C$ to a position $p \in P$. We will formally represent the assignment by $PC_p = c$.*

$$\text{minimize } \left(\sum_{c \in C} \sum_{o \in O} (REQ_{c,o}) * cars_c \right) - \sum_{p \in P} \sum_{o \in O} (REQ_{PC_p,o}) \quad (1)$$

subject to:

$$\sum_{p \in P} (PC_p = c) \leq cars_c \quad \forall c \in C \quad (2)$$

$$\sum_{j=p}^{p+WS_o} REQ_{PC_j,o} \leq MC_o \quad \forall p \in P, \forall o \in O \quad (3)$$

Constraint 2 guarantees that for each configuration c the maximum number of produced car is $cars_c$. While constraint 3 ensures that no bay is overloaded. In other words, for an option o on any sequence of WS_o cars, the maximum number of these cars that requires option o is MC_o .

4 Neighbourhood Selection Heuristic

The method of neighbourhood selection is a key component in any LNS technique. Using an efficient heuristic to select the next set of neighbours that have high probability of being optimised can greatly increase performance. However,

a deterministic approach can result in ignoring some parts of the search space and end up with a relatively poor local minimum solution. On the other hand, a completely stochastic approach may spend a huge computational time on neighbourhoods that do not have scope for improvement.

Our proposed approach exploits the structural relationship between decision variables to guide the search process towards connected neighbourhoods, and information learnt during search to try to choose neighbourhoods with high likelihood of improvement.

Algorithm 1: Neighbourhood Selection Heuristic

```

randomVars  $\leftarrow$  selectNRandomVars(NRandomVars);
bestVar  $\leftarrow$  selectBestVar(randomVars);
relaxVar(bestVar);
while checkSize() do
    randomVars  $\leftarrow$  selectNVarsRelatedTo(NRandomVars, bestVar);
    bestVar  $\leftarrow$  selectBestVar(randomVars);
    relaxVar(bestVar);
end

```

Algorithm 1 describes our domain independent neighbourhood selection heuristic. The heuristic works by incrementally relaxing variables selected according to one of the criteria described in the next subsection. In order to maintain a greater degree of diversification, we first select a random subset of variables (*selectNRandomVars*), and then choose the best amongst this according to the criteria (*selectBestVar*). After a variable is selected, the next selection is constrained to the variables that are involved in constraints with the variables already selected (*SelectNVarsRelatedTo*), except for global constraints that involve more than half of the decision variables. It should be noted that the benefit of the *selectNRandomVars* function isn't just an increase in diversification, it also reduces the computational effort as we only need to find the best amongst this subset as opposed to the best amongst all variables.

4.1 Neighborhood Heuristics

We investigated the performance of the following four heuristic criteria:

Weighted Variable Usage (*V-Usage*): This heuristic prioritises diversification. Each variable has a counter which is incremented when the variable is chosen in the neighborhood of an iteration. The heuristic biases selection to those that have been chosen the least. However, in order to not penalise centroid variables that have to change their values in order to allow other variables to be able to change to the optimal value, the usage score is divided by the number of times that the variable changes its value after a sub-problem optimisation that improves the whole solution. Therefore the criteria is choose the variable with minimum value of *usage/improvements*.

Weighted Variable Cost (*V_Cost*): The criteria for this heuristic is the impact of a variable on the cost (objective function). This cost is calculated by measuring the impact of removing each variable from the current solution.

This heuristic considers the hypothesis that the best neighborhood should involve variables with the higher cost associated with them. The fundamental difference between our variable cost score and the variable cost score from Lombardi and Schaus [9] is that while Lombardi's approach is calculated based on variation of lower bound cost during a range of re-application of the current best solution on a sample of permutation ordering of the variables, our variable cost is calculated based on the impact of unassigning the variable from a full solution.

Weighted Variable Conflicts (*V_Conflicts*): The number of conflicts that each variable was involved on previous iterations. The hypothesis on using this weight is that variables involved in many constraint conflicts are the ones most difficult to find their optimal values, so if we find the optimal values for these variables the others will be easily optimised. We use the variable conflict score implemented on Gecode solver, that is calculated based on the definitions of Conflict history base for SAT problem [7, 8]. For more detail of how the variable conflict is calculated in Gecode, please see [5].

Weighted Variable Failures (*V_Fails*): The number of leaf failures that each variable was involved on previous iterations divided by the variable domain size after being relaxed and propagating arc-consistency based on the fixed variables on a given iteration. We hypothesise that variables with a high number of failures are the ones most difficult to find their optimal values, so they need to be relaxed more frequently. For more details of how the Failures criteria is calculated in Gecode, please see [13].

For the latter three approaches, the scores were normalized by dividing it by the number of times that the variable was relaxed in the previous iterations. Therefore, impact of behavior in earlier iterations will not dominate.

5 Evaluation

5.1 Experimental Design

We implemented our proposed approach using Gecode 6.2 [4]. For comparison, we have also tested 4 neighbourhood selection heuristics from the literature: PGLNS¹ [11], Interval-based [10], Cost-Impact (referred to as *CGLNS* in results) [9], and Random which simply chooses the variables for the neighborhood randomly.

It should be noted that we are using the best configuration presented in [11], that iterates through the following three neighborhood operators: Propagation Guided; Reverse Propagation Guided; and purely random selection. However, we defined the neighbourhood size based on the number of relaxed variable

¹ The description of the PGLNS approach from [11] miss some details, thus implementation differences may exist.

Table 1. Configurations parameters for the benchmark experiment.

Parameter	Value
Runtime	120 s
Neighbourhood size	10 slots
Failure threshold	200
NRandomVars	10

instead of the search space size, in order to compare all approach on the same neighbourhood size.

The experiments were run on a Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-70-generic) with 16 Core and 32 GB, with a runtime cutoff of 2 min per approach on each instance. Furthermore, as all approaches have a strong stochastic element, the presented results are the average of 5 runs with different seeds. Table 1 presents the parameter configurations that was used to run the experiments.

5.2 Instances

The experiments will use the three sets of hard instances available on the CSPLib [15]. There are 10 instances in each set and the total number of cars per instance is 200, 300 and 400 respectively. This allows us to also empirically analyse the scalability performance and behaviour of each approach as problems grow in size.

5.3 Results

The results are presented in Table 2 in terms of average cost across the five runs, and the associated standard deviation of the cost. We further provide details on the number of best solutions found for each approach per problem set.

We see that all approaches provide significant improvement over the initial solution. Amongst the comparison approaches, PGLNS performs best with significant improvements over the two other domain-independent heuristics (Random and CGLNS), and was consistently better than the problem-specific heuristic (Interval) on all problem sizes.

Amongst the approaches proposed in this work, we find that all perform well. Although PGLNS outperforms them on the smallest instance size, as the instance size increases, so does the improvement over PGLNS for each. This can be seen more clearly in the number of total best solutions found. For the smallest instances, PGLNS finds most, but can only find one for the largest instances compared to six for *V_Usage*.

The *V_Fails* heuristic achieved the best results on the two largest set of instances (300 and 400 cars), while on the set of instance with 200 cars, PGLNS and Interval Based found better solutions on average. Interestingly we did not find a large difference in behaviour of our four heuristics in terms of solution

Table 2. Five runs on three problem sets of 10 instances with 120 s cutoff per instance run. Results per problem set in terms of: average and standard deviation of cost; and number of instances for which a method found best solution across methods tested.

Approach	Average cost			Standard deviation cost			Total best solution		
	Size ₂₀₀	Size ₃₀₀	Size ₄₀₀	Size ₂₀₀	Size ₃₀₀	Size ₄₀₀	Size ₂₀₀	Size ₃₀₀	Size ₄₀₀
<i>Initial Sol</i>	92.0	142.7	187.6	6.9	13.1	11.0	—	—	—
<i>Random</i>	19.6	33.9	45.2	4.6	8.5	6.8	0	0	0
<i>Interval</i> [10]	15.6	27.7	34.6	4.1	6.0	6.2	4	2	0
<i>PGLNS</i> [11]	15.4	27.1	33.9	3.9	6.3	5.9	4	3	1
<i>CGLNS</i> [9]	38.4	64.8	72.3	30.7	52.8	55.7	1	0	0
<i>V_Conflict</i>	16.5	26.6	33.0	3.6	6.3	6.7	2	4	3
<i>V_Fails</i>	16.6	26.5	33.0	3.5	6.3	6.6	3	3	2
<i>V_Cost</i>	15.9	26.6	33.3	3.6	6.0	5.4	3	6	3
<i>V_Usage</i>	16.5	26.7	33.1	3.5	6.2	6.7	3	3	6

quality. This suggests the importance of the concept of “variable relationship” which underpins all our heuristics.

We also generated search statistics, Table 3, for the different approaches in order to gain further insight and understanding into their behavior. These results answer a number of questions. Firstly, we see that both Random and CGLNS performed significantly more iterations than the other approaches, between 5 and 10 times as many. This explains the surprising result that CGLNS had significantly worse performance than even the random neighborhood selector, despite exploring approximately four times as many iterations as PGLNS or Interval.

Table 3. Analysis of search behaviour for different methods averaged across runs and instances.

Approach	Iterations			Nodes per iteration		
	Size ₂₀₀	Size ₃₀₀	Size ₄₀₀	Size ₂₀₀	Size ₃₀₀	Size ₄₀₀
<i>Random</i>	80, 422	53, 958	33, 155	13	8	11
<i>Interval</i>	12, 493	8, 687	5, 849	207	203	204
<i>PGLNS</i>	12, 086	9, 662	6, 051	195	193	199
<i>CGLNS</i>	57, 967	38, 818	25, 227	27	24	25
<i>V_Conflict</i>	6, 664	4, 898	3581	376	374	372
<i>V_Cost</i>	9, 185	5, 965	3782	298	312	328
<i>V_Fails</i>	7, 285	4, 516	3589	376	374	373
<i>V_Usage</i>	6, 501	4, 511	3897	376	374	374

This may seem counter-intuitive, more iterations means more neighborhoods explored which sounds in theory like it should be beneficial. The reason for this not being the case for Random and CGLNS is that many of the neighborhoods selected had scope for very few, if any, variable changes. This is evidenced by the average nodes explored per iteration by these two approaches in comparison to the other approaches. Neither of these approaches take into account the relationship between variables selected. In other words, they relax disconnected variables, and propagating the assignment of the non-relaxed variables results in the domains of most relaxed variables reducing to the previous value.

On the other hand, we see the opposite is the case for the approach we propose, irrespective of the heuristic criteria. Comparing to PGLNS and Interval, our approaches performed over 25% fewer iterations, but explored nearly twice as many nodes per iteration on average. All our approaches use the concept of “variable relationship” in order to build out a connected subset of variables from the initial variable selection. We see a consistent trend whereby this resulted in exploring more nodes per iteration, as more combinations could be tried since the relaxed variables were connected and were not having their domains as restricted by the non-frozen variables.

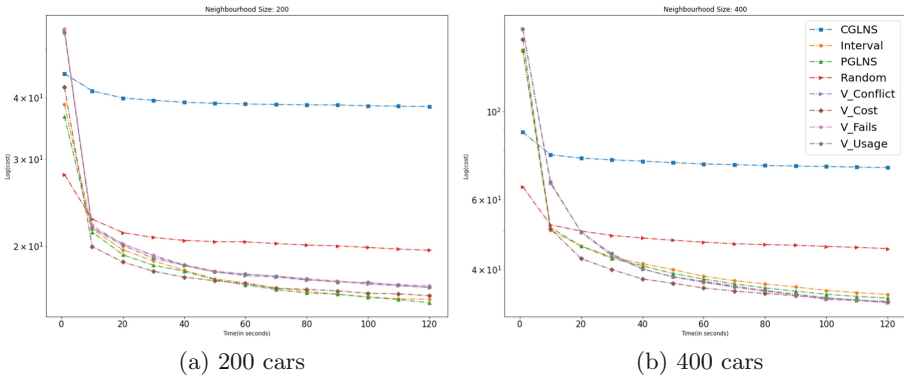


Fig. 2. Evolution of solution quality across time for different instance sizes.

Figure 2 shows the evolution of average cost improvement over 10 instances across 2 min of search for the different configurations described above. We note that CGLNS and more so Random were able to make larger improvements in the first second as they explore more neighborhoods and improvements over the weak initial solution are easy to achieve. However, they quickly stagnate.

As we can see, and showed already in Table 2 PGLNS and Interval-Based have better performance overall on the instances of smallest size (200 cars), whilst our Variable-Relationship neighbourhood selection heuristics got better results on the instances of largest size (400 cars) over 120 s. This behaviour was somewhat expected, since on small size instances it is not essential to prioritise variables with high probability of generated big improvement, as the 120 s search time can

guarantee enough iterations to investigate many of the possible neighbourhoods. However, as the instance size increases, the need to prioritise more promising neighbourhoods in each step of the search process also increases, and the Variable Relationship heuristics work better.

We can observe this behaviour in the first 40 s of Graph 2a where the *V_Cost* heuristic got the best performance as it prioritise neighbourhoods with variables that have more impact on the objective function. Interestingly *V_Cost* was significantly better than other approaches after 40 s for both instance sizes, and indeed for size 300 (not shown). This may in part be the combination of searching more neighborhoods than our other heuristics (as evidenced by greater number of iterations in Table 3) while still keeping the Variable-Relationship. These results suggest alternating between our different neighborhood operators may produce better results.

6 Conclusion and Future Works

In this paper, we proposed an approach that combines knowledge extracted from the problem structure and search state information to generate complex and diversified neighbourhoods without the need of a domain specific algorithm. Our heuristic works by incrementally relaxing variables based on its state, and their relationship to other variables selected. In particular, after each variable is selected by the heuristic, the next selection is constrained to the variables that are involved in constraints with the variables already selected.

We empirically evaluated our approach using public instances of Car Sequencing Problem [15]. Comparing our results against domain specific heuristic, SOA generic approaches, and pure random relaxation demonstrated the effectiveness of Variable-Relationship Guided LNS mainly on large instances. Further analysis of search behaviour, in terms of average nodes explored per iteration, provided insight into why these approaches performed so well.

To the best of our knowledge the Variable-Relationship Guided LNS is the first domain independent neighbourhood selection heuristic to combine information from problem structure and that learnt through search performance. Even though the empirical results prove that good neighborhoods can be identified through combining information regarding the problem structure with information collected during search on an optimisation version of the Car Sequencing Problem, there are some promising avenues for future work such as:

- The combination of different types of search state information in the same search process. The main challenge here is to define the relative importance of each information in the construction of the neighbourhood. We believe that Machine Learning/Deep Learning are key to address this challenge in a generic and adaptive way.
- Identifying more complex variable relationships (i.e. for variables not directly connected by a constraint). Graph Convolution Networks could be beneficial to learn more robust relationships of variables based on the graph representation of the constraint relationship between decision variables.

References

1. Artigues, C., Hebrard, E., Mayer-Eichberger, V., Siala, M., Walsh, T.: SAT and hybrid models of the car sequencing problem. In: Simonis, H. (ed.) CPAIOR 2014. LNCS, vol. 8451, pp. 268–283. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07046-9_19
2. Carchrae, T., Beck, J.C.: Cost-based large neighborhood search. In: Workshop on the Combination of Metaheuristic and Local Search with Constraint Programming Techniques (2005)
3. Freuder, E.C., O’Sullivan, B.: Grand challenges for constraint programming. In: Constraints, pp. 1–13 (2014). <https://doi.org/10.1007/s10601-013-9155-1>
4. Gecode Team: Gecode: Generic constraint development environment (2006). <http://www.gecode.org>
5. Habet, D., Terrioux, C.: Conflict history based heuristic for constraint satisfaction problem solving. *J. Heuristics* **27**(6), 951–990 (2021). <https://doi.org/10.1007/s10732-021-09475-z>
6. Laborie, P., Godard, D.: Self-adapting large neighborhood search: application to single-mode scheduling problems. In: Proceedings MISTA-07, Paris 8 (2007)
7. Liang, J., Ganesh, V., Poupart, P., Czarnecki, K.: Exponential recency weighted average branching heuristic for sat solvers. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1 (2016)
8. Liang, J.H., Ganesh, V., Poupart, P., Czarnecki, K.: Learning rate based branching heuristic for SAT solvers. In: Creignou, N., Le Berre, D. (eds.) SAT 2016. LNCS, vol. 9710, pp. 123–140. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40970-2_9
9. Lombardi, M., Schaus, P.: Cost impact guided LNS. In: Simonis, H. (ed.) CPAIOR 2014. LNCS, vol. 8451, pp. 293–300. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07046-9_21
10. Perron, L., Shaw, P.: Combining forces to solve the car sequencing problem. In: Régim, J.-C., Rueher, M. (eds.) CPAIOR 2004. LNCS, vol. 3011, pp. 225–239. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24664-0_16
11. Perron, L., Shaw, P., Furnon, V.: Propagation guided large neighborhood search. In: Wallace, M. (ed.) CP 2004. LNCS, vol. 3258, pp. 468–481. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30201-8_35
12. Pisinger, D., Ropke, S.: Large neighborhood search. In: Handbook of Metaheuristics, pp. 399–419. Springer, Heidelberg (2010)
13. Schulte, C., Tack, G., Lagerkvist, M.Z.: Modeling and programming with gecode. Schulte, Christian and Tack, Guido and Lagerkvist, Mikael, vol. 1 (2010)
14. Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: Maher, M., Puget, J.-F. (eds.) CP 1998. LNCS, vol. 1520, pp. 417–431. Springer, Heidelberg (1998). https://doi.org/10.1007/3-540-49481-2_30
15. Smith, B.: CSPLib problem 001: Car sequencing. <http://www.csplib.org/Problems/prob001>
16. Thomas, C., Schaus, P.: Revisiting the self-adaptive large neighborhood search. In: van Hoeve, W.-J. (ed.) CPAIOR 2018. LNCS, vol. 10848, pp. 557–566. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93031-2_40

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Unimodal and Multimodal Representation Training for Relation Extraction

Ciaran Cooney^(✉), Rachel Heyburn, Liam Madigan, Mairead O’Cuinn,
Chloe Thompson, and Joana Cavadas

Aflac NI, Belfast, Northern Ireland

{ccooney,rheyburn,lmadigan,mocuinn,cthompson,jcavadas}@aflac.com

Abstract. Multimodal integration of text, layout and visual information has achieved SOTA results in visually rich document understanding (VrDU) tasks, including relation extraction (RE). However, despite its importance, evaluation of the relative predictive capacity of these modalities is less prevalent. Here, we demonstrate the value of shared representations for RE tasks by conducting experiments in which each data type is iteratively excluded during training. In addition, text and layout data are evaluated in isolation. While a bimodal text and layout approach performs best ($F1 = 0.684$), we show that text is the most important single predictor of entity relations. Additionally, layout geometry is highly predictive and may even be a feasible unimodal approach. Despite being less effective, we highlight circumstances where visual information can bolster performance. In total, our results demonstrate the efficacy of training joint representations for RE.

Keywords: Relation extraction · Multimodal deep learning · Joint representation training · Information retrieval

1 Introduction

With many sectors such as healthcare, insurance and e-commerce now relying on digitization and artificial intelligence to exploit document information, Visually-rich Document Understanding (VrDU) has become a highly active research domain [11, 14, 21, 24]. VrDU is the task of analyzing scanned or digital business documents to allow structured information to be extracted for downstream business applications [21]. Sub-fields including Named-Entity Recognition (NER) [2], layout understanding [7] and document classification [22] all seek to extract meaningful information from documents. Another sub-field of VrDU, relation extraction (RE) offers the possibility of linking named entities in documents so that a paired relationship can be identified [3, 5, 6, 11, 23]. Typically, relations are defined in a question-answer (Q/A) format and the RE task is to define a function which predicts if a pair of entities in a document are related or not [11, 23].

Concurrent with recent developments in VrDU, advances in multimodal deep learning have seen novel methods applied across fields as diverse as medical imaging [16], neurotechnology [4] and early prediction of Alzheimer’s disease [17].

Commercial and opensource optical character recognition engines such as AWS Textract¹, Microsoft Read API² and PyTesseract³ enable extraction of detailed text and geometric information from visually-rich documents, which along with visual information has led to a plethora of multimodal architectures being applied to VrDU tasks [12, 14, 19, 21, 22, 24]. These approaches enable learning of joint representations in a single end-to-end training procedure with the aim of maximising the total information in a document. Although transformer-based architectures are prominent in this field [13, 23], other methods for optimizing RE tasks, such as graph neural networks [3, 6], have been reported.

Datasets for RE facilitate the use of multimodal representations for training co-adaptive networks [11, 23]. However, despite the growth of multimodal approaches in VrDU tasks, the extent to which learning joint representations is an enhancement remains unclear, as does the relative capacity of each data type. Although text is likely more predictive than either geometric layout or visual information, the extent to which this is the case and the interaction between modes is not known. Even in studies with ablation tests, consideration of the effects of training without text representations has not been applied [9].

We aim to address uncertainty regarding the predictive capacity of different data by performing a series of experiments with different multimodal and unimodal configurations. We apply our analysis to the RE task, due to it being an unresolved information extraction challenge relevant to several industry applications, and one that could benefit from appropriately trained joint representations. Our contributions are summarised as follows: (1) We prove the efficacy of using joint representations for VrDU RE. Specifically we demonstrate that a text/layout configuration yields the best performance. (2) We analyse the asymmetric predictive capacity of text, layout and visual data, exhibiting the anticipated relative importance of text over the other data while highlighting where layout and visual information can be effectual. (3) We present a simplified classifier for RE based on the LayoutXLM classification head [23].

Section 2 describes previous works related to multimodal approaches to document understanding tasks and questions current understanding of the impact of different modalities. Section 3 reports our methodology, including the dataset used, model architecture, and experimental procedures. In Sect. 4, we present the results of our experiments. Section 5 contains limitations and suggestions for future work. In Sect. 6 we forward concluding remarks.

2 Related Work

Two datasets exist for the VrDU RE task. These are FUNSD [11] and XFUND [23]. Both contain annotations which include an indication of linked entities

¹ <https://aws.amazon.com/textract/>.

² <https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/overview-ocr>.

³ <https://pypi.org/project/pytesseract/>.

comprising of two entity IDs which are linked, or an empty array indicating no relationship.

The provision of text, geometry and document images in these datasets enables the use of multimodal methods for document understanding. The LayoutLM family of models [21–23] utilise combined text and position embeddings to leverage the layout of the document, with LayoutLMv2 extending this approach to fully incorporate visual information. The approach of [18] uses a similar trifecta of inputs to perform tasks on the FUNSD and MedForm datasets. [1] use a multimodal approach for text and image-based document classification, while others focus on text and layout representations [12, 13, 15].

The use of multimodal approaches poses an important question: *what are the relative effects of the different data types?* It is not always clear from reported results, even in studies that do present ablation findings, what the impact of different modalities is. Particularly since text since is usually retained in conducted experiments [9, 15]. For industry applications, the additional training and inference costs associated with large-scale multimodal approaches must be mitigated by performance benefits. The original XFUND paper does not report on the relative impact of the different components (text, layout and visual information) [23], informing the approach taken here.

3 Methodology

We use the XFUND dataset⁴ to experiment with different modalities for the RE task [23]. The dataset consists of document images for form understanding in seven languages. Annotations corresponding to each of the documents contain a unique identifier, class label, bounding box coordinates ($x_{left}, y_{top}, x_{right}, y_{bottom}$), text and a linking indicator. This linking indicator facilitates the use of XFUND in VrDU RE. Entities for RE are designated key-value pairs corresponding to questions and answers in the forms. For further information on dataset collection and curation, see [23]. Dataset statistics differ from those reported in [23] and are therefore presented in Table 1. *ZH*, *JA*, *ES*, *FR*, *IT*, *DE* and *PT* correspond to Chinese, Japanese, Spanish, French, Italian, German and Portuguese, respectively.

Table 1. Train/Test split for XFUND data.

	ZH	JA	ES	FR	IT	DE	PT
Train	187	194	243	202	265	189	233
Test	65	71	74	71	92	63	85

⁴ <https://github.com/doc-analysis/XFUND>.

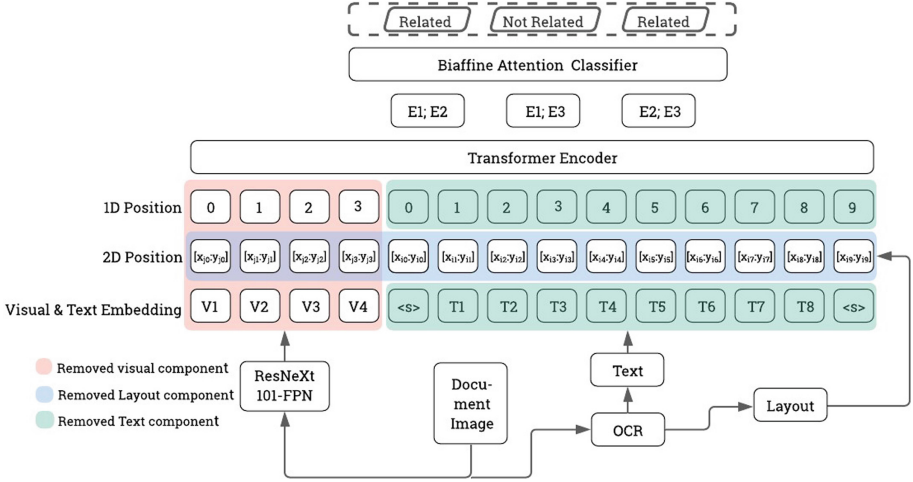


Fig. 1. Multimodal transformer with data exclusions color-coded. Pink denotes exclusion of visual components, blue exclusion of layout, and green exclusion of text representations. (Color figure online)

3.1 LayoutXLM for Relation Extraction

The multimodal deep learning architecture we use to perform our experiments is LayoutXLM, a pretrained transformer for document understanding [23], based on the LayoutLMv2 architecture [21]. The model ingests text, layout (bounding boxes) and visual information which are encoded in embedding layers (Fig. 1). It utilises traditional position embeddings to model word position in a sequence, and 2D-position embeddings to consider relative spatial position. A visual backbone encodes image representations using ResNeXt 101-FPN [20].

For the RE task, a bespoke classification layer is attached to the pretrained model for further fine-tuning. In the original LayoutXLM, a bi-affine classifier receives representations of Q/A entities which are a processed version of the first token vector and an entity type embedding for each. A feed-forward neural network is applied to these representations before they are fed to the bi-affine classifier. Here, we have slightly modified the classification layer to simplify the feed-forward neural network and therefore reduce the number of parameters. In this work the feed-forward neural network consists of a single fully-connected layer, leaky relu activation and dropout ($p = 0.2$).

3.2 Experimental Procedures

We conduct experiments for six different multimodal and unimodal configurations for each of the seven sets in the multilingual XFUND data. Experiments consist of fine-tuning the pretrained LayoutXLM model on various configurations of the available data. The six model configurations are: (1) Multimodal text, layout and visual (MM), (2) bimodal text and layout (text/layout), (3) bimodal

text and visual (text/visual), (4) bimodal layout and visual (layout/visual), (5) unimodal layout (layout) and (6) unimodal text (text). We initially planned to include a unimodal visual experiment but early results indicated this was not feasible for the RE task.

For experiment 1, there are no further modifications to the network beyond those specified in Sect. 3.1. For experiment 2, all visual components of the architecture are removed. This includes the visual backbone and all related embeddings, including 2D and 1D visual position embeddings (Fig. 1; pink). The model is therefore only trained on text and layout information. In experiment 3, layout information is removed from the network in the form of 2D position embeddings. Included in this step is the removal of 2D position embeddings from the visual component (Fig. 1; blue). Experiment 4 excludes all text information, including tokenized text and the associated 1D and 2D position embeddings (Fig. 1; green). Tasks 5 and 6 combine the relevant exclusions applied in experiments 2, 3 and 4. Studies often neglect to experiment with the exclusion of text [9, 15], despite its importance when analysing the predictive capacity of multimodal approaches.

Due to the nature of the ablation experiments we hypothesised that a degree of variation in optimal learning rates across the different data types and perhaps between the individual datasets was likely. For this reason, learning rate is the only hyperparameter optimized. Three learning rates were optimized with simple gridsearch: $5e^{-5}$, $1e^{-5}$ and $5e^{-6}$. All other learning parameters are identical across experiments. For fine-tuning, batch size is 2, and all models are allowed to train for 50 epochs.

4 Results

For all experiments, F1 score is the primary evaluation metric, with precision and recall also reported. Although multimodal results have previously been reported for the XFUND RE task [23], we chose not to include them in our reporting due differences in network configuration, dataset statistics and training procedures.

4.1 Bimodal Training Outperforms Trimodal

F1 scores obtained from models trained with each of the six different network configurations are reported in Table 2 and Fig. 2. Additional recall and precision scores are in Tables 3 and 4. Overall results validate the utility of training on joint representations for the VrDU RE task, while also exhibiting asymmetric predictive capacity across the different modalities (Table 2). Here, the bimodal text/layout configuration outperforms the three-pronged multimodal approach with mean F1 scores of 0.684 and 0.673, respectively. This is consistent with previous work demonstrating the utility of text and layout information without the requirement of visual information [12]. These two training configurations significantly outperform all other approaches with scores 8.08% and 6.93% greater than the next best approach (Fig. 2). The other bimodal approaches, text/visual and layout/visual, result in reasonably strong F1 scores of 0.604 and 0.558 and

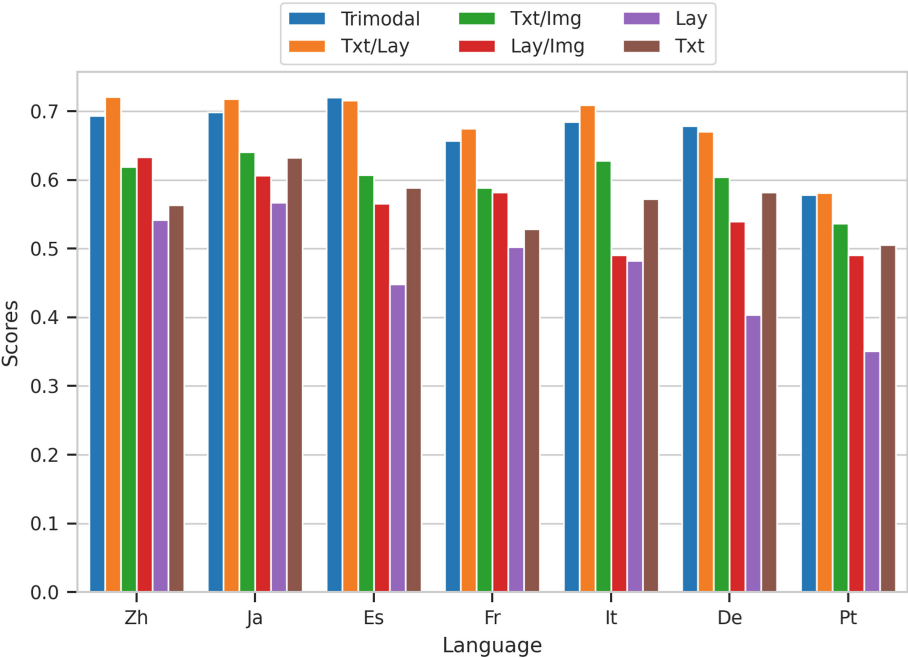


Fig. 2. Bar chart reporting the validation F1 scores for each language dataset with each network configuration.

further suggest that joint representations can be effective for this task. However, the overall picture indicates that in this particular application of multimodal deep learning there is a hierarchy of predictive capacity with text on top and visual information at the bottom.

Pairwise comparison of results from full multimodal training and the text/layout configuration indicate that the impact of visual information is negligible or that its inclusion is even counterproductive (Table 2). However, comparison of unimodal and bimodal results suggest visual information can be effective in the correct context. Most obvious is the positive impact of visual information when compared to the unimodal layout results. In this case the inclusion of visual data improves upon layout only results by improving the F1 score from 0.471 to 0.558. In fact, the combination of these two modalities results in similar performance to the unimodal text method. It is possible that this approach is useful in scenarios where text quality is degraded but visual and layout information are sufficient to classify the document. However, results do not suggest that visual information is necessary for the RE task.

Significant variation in performance corresponding to the different document languages may have been expected, particularly as there is a clear dichotomy between those using Kanji characters and those using the Latin alphabet. Despite

some of the latin languages exhibiting less effective impact of layout information, Fig. 2 indicates similar levels of overall performances across datasets, despite differences in the number of document samples per set (Table 1).

Table 2. XFUND F1 scores for different training configurations.

	MM	Txt/Lay	Txt/Im	Lay/Im	Layout	Text
ZH	0.6935	0.7212	0.6192	0.6334	0.5417	0.5636
JA	0.6987	0.7181	0.6406	0.6061	0.5674	0.6321
ES	0.7198	0.7159	0.6069	0.5657	0.4483	0.5885
FR	0.6573	0.6747	0.5888	0.5820	0.5021	0.5285
IT	0.6841	0.7090	0.6281	0.4906	0.4825	0.5724
DE	0.6782	0.6701	0.6041	0.5397	0.4035	0.5821
PT	0.5779	0.5812	0.5367	0.4909	0.3511	0.5053
Mean	0.6728	0.6843	0.6035	0.5583	0.4709	0.5675

Table 3. Recall scores for XFUND data for different training configurations.

	MM	Txt/Lay	Txt/Im	Lay/Im	Layout	Text
ZH	0.6109	0.7607	0.6754	0.7011	0.6639	0.6440
JA	0.5638	0.7540	0.6601	0.6619	0.6932	0.6762
ES	0.6475	0.7210	0.6817	0.6807	0.4487	0.6136
FR	0.6231	0.7278	0.6365	0.7278	0.6956	0.5774
IT	0.6314	0.7090	0.6584	0.5649	0.5966	0.7009
DE	0.6990	0.6518	0.6267	0.6158	0.4279	0.5931
PT	0.4851	0.6640	0.5308	0.5891	0.4082	0.5089
Mean	0.6087	0.7126	0.6385	0.6488	0.5620	0.6163

Table 4. Precision scores for XFUND data for different training configurations.

	MM	Txt/Lay	Txt/Im	Lay/Im	Layout	Text
ZH	0.6109	0.6855	0.5717	0.5777	0.4576	0.5010
JA	0.5638	0.6854	0.6223	0.5589	0.4802	0.5934
ES	0.6475	0.7108	0.5469	0.4768	0.4478	0.5654
FR	0.6231	0.6288	0.5478	0.4848	0.3928	0.4872
IT	0.6314	0.6862	0.6004	0.4336	0.4050	0.4837
DE	0.6990	0.6894	0.5831	0.5804	0.3817	0.5715
PT	0.4851	0.5167	0.5427	0.4207	0.3080	0.5016
Mean	0.6087	0.6575	0.5736	0.5046	0.4104	0.5291

4.2 Text is the Anchor for Relation Extraction

As expected, results clearly indicate the most significant drop-off in F1 score occurs when text is excluded. Of the six approaches, the two for which text is excluded are the poorest performing. The fact that the unimodal text method exhibits better classification performance than the bimodal layout/visual approach is a strong indicator of the dominance of text for the RE task.

Notwithstanding, the inclusion of supplementary data to enhance performance is extremely valuable. Results show that each of the three configurations that included the addition of other modalities alongside text produced improved performance. In this case, text is necessary but not sufficient to achieve the best possible performance. Clearly, layout information provides very important supplementary information for the RE task. Not only is it used along with text in the highest performing configuration but it also exhibits reasonable performance in the absence of text, in both unimodal (0.471) and bimodal (0.558) networks.

Despite clear dominance, there is notable variation in the strength of predictions across languages in the absence of text representations. For ZH and FR there is minimal difference or even improvement when text is replaced with layout information, whereas for IT and DE this difference is substantial (Table 2). This suggests that document diversity across regions or business sectors may modulate the effectiveness of different approaches to training joint representations. It may not always be obvious *a priori* whether or not supplementing text data with other modalities will actually enhance model performance. Given trade-offs associated with speed and complexity, particularly with the inclusion of visual information, efforts should be made to evaluate the value proposition when applying multimodal techniques in information retrieval tasks.

4.3 Training Variability is Data Dependent

The model's receptivity to learning the RE task from text is further illuminated by training loss trends for each configuration (Fig. 3). A much sharper decline towards convergence is present in those model configurations considering text than those excluding it. Each of the four approaches using text converge on a similar loss after 50 epochs. The other two methods (layout and layout/image) exhibit shallower learning trajectories, greater variance between datasets and less ability to converge within the prescribed training time. Again this validates the extent to which text is the most important anchor in training even when other data modalities are also included. The higher variance in loss exhibited by the non-text approaches may also indicate that the effectiveness of layout and image data may be more dependent on specific datasets than text. Another factor influencing these results is the optimal learning rate selected for each experiment. While this varied somewhat between the different language sets within experiments, there is more obvious variation between experiments. The full multimodal approach and the text/layout approach are both trained with an optimal learning rate of $5e^{-5}$. Experiments with text excluded used learning rates of either $1e^{-5}$ or $5e^{-6}$.

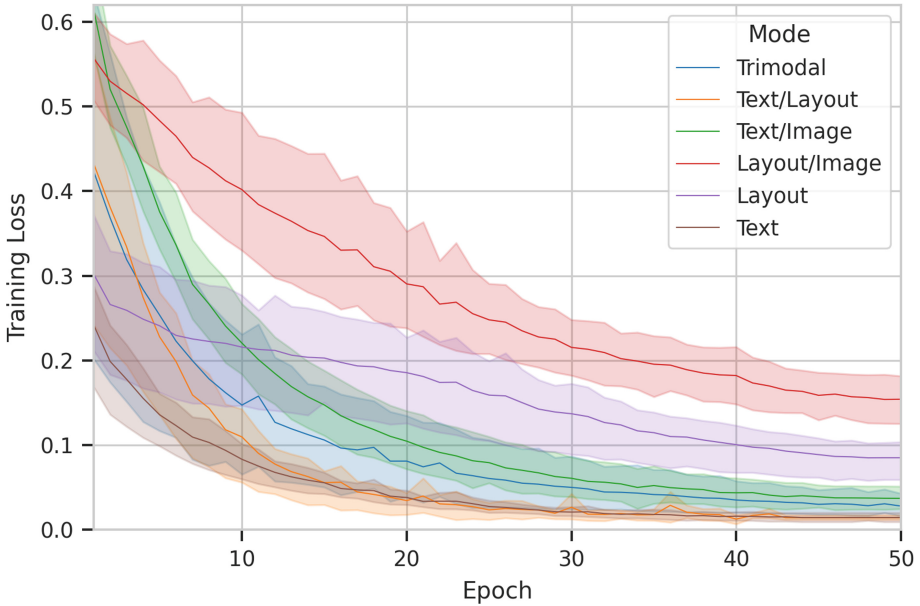


Fig. 3. Training loss over 50 epochs for each network configuration. Shadows indicate the range of loss across each language dataset for a given epoch.

5 Limitations and Future Work

Limitations associated with this work include the relative sparsity of data currently available for the VrDU RE task and the constraint of applying the experiments to LayoutXML only. To the best of our knowledge FUNSD [11] is the only other dataset currently available for the RE task. Increasing the diversity of datasets and also the volume of samples within datasets could allow us to validate these results further. Additionally, other multimodal architectures exist with different approaches to encoding and combining modalities [8, 10]. Extending this work to include analysis of these different methods would provide a stronger basis on which to judge the relative contributions of text, layout and visual information to the RE task.

As well as addressing these limitations, future work may involve a large-scale analysis of multimodal approaches to a variety of VrDU tasks. This could include document classification, semantic entity recognition, and key information extraction tasks and ablation analysis could be applied to a variety of model architectures in order to fully understand how their performance differs according to modality. In addition, time and complexity analysis is required to understand the relative utility of different approaches within business environments that may process documents at high daily volumes. This would facilitate understanding of the efficiency and cost value of the different methods.

6 Conclusions

Multimodal methods can have trade-offs with respect to complexity, performance and speed when applied to industry applications. We trained a multimodal transformer using several data configurations to understand the impact of training joint representations for the VrDU RE task. The bimodal text and layout approach resulted in the best performance, even beating the full multimodal configuration. Individually, text accounts for a greater portion of the predictive capacity than either layout or visual data. Unimodal text achieved higher mean F1 score than bimodal layout/visual data and all configurations with text included outperformed those with text excluded. Nevertheless, both layout and visual information was proven to be effective in specific conditions. Although, visual information is currently exhibiting value as a supplementary data source to boost overall performance, our results show that layout information is extremely important to the RE task. We also showed that training varies depending on the inclusion/exclusion of different data types. Future work is required to examine methods for optimizing joint representation training for document understanding, including how to best combine different data in multimodal approaches.

References

1. Audebert, N., Herold, C., Slimani, K., Vidal, C.: Multimodal deep networks for text and image-based document classification. arXiv preprint [arXiv:1907.06370](https://arxiv.org/abs/1907.06370) (2019)
2. Carbonell, M., Fornés, A., Villegas, M., Lladós, J.: A neural model for text localization, transcription and named entity recognition in full pages. *Pattern Recogn. Lett.* **136**, 219–227 (2020)
3. Carbonell, M., Riba, P., Villegas, M., Fornés, A., Lladós, J.: Named entity recognition and relation extraction with graph neural networks in semi structured documents. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 9622–9627. IEEE (2021)
4. Cooney, C., Folli, R., Coyle, D.: A bimodal deep learning architecture for EEG-fNIRS decoding of overt and imagined speech. *IEEE Trans. Biomed. Eng.* **69**, 1983–1994 (2021)
5. Dang, T.A.N., Hoang, D.T., Tran, Q.B., Pan, C.W., Nguyen, T.D.: End-to-end hierarchical relation extraction for generic form understanding. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 5238–5245. IEEE (2021)
6. Davis, B., Morse, B., Price, B., Tensmeyer, C., Wiginton, C.: Visual fudge: form understanding via dynamic graph editing. arXiv preprint [arXiv:2105.08194](https://arxiv.org/abs/2105.08194) (2021)
7. Gralinski, F., et al.: Kleister: a novel task for information extraction involving long documents with complex layout. CoRR abs/2003.02356 (2020). <https://arxiv.org/abs/2003.02356>
8. Gu, Z., et al.: XYLayoutLM: Towards layout-aware multimodal networks for visually-rich document understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4583–4592 (2022)
9. Hong, T., Kim, D., Ji, M., Hwang, W., Nam, D., Park, S.: BROS: a pre-trained language model focusing on text and layout for better key information extraction from documents. arXiv preprint [arXiv:2108.04539](https://arxiv.org/abs/2108.04539) (2021)

10. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: LayoutLMv3: pre-training for document ai with unified text and image masking. arXiv preprint [arXiv:2204.08387](https://arxiv.org/abs/2204.08387) (2022)
11. Jaume, G., Ekenel, H.K., Thiran, J.P.: FUNSD: a dataset for form understanding in noisy scanned documents. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), vol. 2, pp. 1–6. IEEE (2019)
12. Li, C., et al.: StructuralLM: Structural pre-training for form understanding. arXiv preprint [arXiv:2105.11210](https://arxiv.org/abs/2105.11210) (2021)
13. Li, Y., et al.: StrucText: structured text understanding with multi-modal transformers. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 1912–1920 (2021)
14. Liu, X., Gao, F., Zhang, Q., Zhao, H.: Graph convolution for multimodal information extraction from visually rich documents. arXiv preprint [arXiv:1903.11279](https://arxiv.org/abs/1903.11279) (2019)
15. Pramanik, S., Mujumdar, S., Patel, H.: Towards a multi-modal, multi-task learning based pre-training framework for document representation learning. arXiv preprint [arXiv:2009.14457](https://arxiv.org/abs/2009.14457) (2020)
16. Sharif, M.I., Khan, M.A., Alhussein, M., Aurangzeb, K., Raza, M.: A decision support system for multimodal brain tumor classification using deep learning. *Complex Intell. Syst.* 1–14 (2021)
17. Venugopalan, J., Tong, L., Hassanzadeh, H.R., Wang, M.D.: Multimodal deep learning models for early detection of Alzheimer’s disease stage. *Sci. Rep.* **11**(1), 1–13 (2021)
18. Wang, Z., Zhan, M., Liu, X., Liang, D.: DocStruct: a multimodal method to extract hierarchy structure in document for general form understanding. arXiv preprint [arXiv:2010.11685](https://arxiv.org/abs/2010.11685) (2020)
19. Wei, M., He, Y., Zhang, Q.: Robust layout-aware IE for visually rich documents with pre-trained language models. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2367–2376 (2020)
20. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. arXiv preprint [arXiv:1611.05431](https://arxiv.org/abs/1611.05431) (2016)
21. Xu, Y., et al.: LayoutLMv2: multi-modal pre-training for visually-rich document understanding. arXiv preprint [arXiv:2012.14740](https://arxiv.org/abs/2012.14740) (2020)
22. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: LayoutLM: pre-training of text and layout for document image understanding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1192–1200 (2020)
23. Xu, Y., et al.: LayoutXLM: multimodal pre-training for multilingual visually-rich document understanding. arXiv preprint [arXiv:2104.08836](https://arxiv.org/abs/2104.08836) (2021)
24. Zhang, P., et al.: TRIE: end-to-end text reading and information extraction for document understanding. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 1413–1422 (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Run-Time Norms Synthesis in Dynamic Environments with Changing Objectives

Maha Riad^(✉) , Saeedeh Ghanadbashi, and Fatemeh Golpayegani

School of Computer Science, University College Dublin, Dublin, Ireland
{maha.riad,saeedeh.ghanadbashi}@ucdconnect.ie, fatemeh.golpayegani@ucd.ie
<https://mas3.ucd.ie>

Abstract. Normative Multi-Agent Systems (NorMAS) can model real-world applications as multi-agent systems and facilitate the coordination of the social behaviour of various entities (agents) interacting in an environment using norms. Aligning such norms with the objectives of the agents is crucially important to ensure that applying the norms would not affect the achievement of their objectives. However, when the environment is dynamic, agents can face unseen situations and might need to change their objectives accordingly. Therefore, it becomes more challenging to understand the change, synthesise norms, and align them with such dynamic objectives. This paper introduces a Dynamic Objectives and Norms Synthesizer and Reasoner (DONSR) model to align objectives and norms using a utility-based approach. An ontology-based schema, forward reasoning, and backward reasoning are used to identify the change in the environment and synthesise new objectives. Case-based reasoning enables the dynamic changing and reasoning of previously created objectives and synthesising norms. DONSR is evaluated using multiple simulated traffic scenarios, including different unseen situations (emergency events).

Results show that norms can be synthesised and maintained efficiently while the objectives are being created and changed. Further, DONSR showed its efficacy in handling unseen situations, creating new objectives, and aligning them with the created norms.

Keywords: Normative multi-agent systems · Norms synthesis · Dynamic objectives

1 Introduction

Multi-agent systems (MAS) model complex systems that consist of autonomous agents with various objectives [4, 7]. These objectives are achieved by agents interacting together, competing, or cooperating [5]. Normative multi-agent systems (NorMAS) can coordinate the behaviour of agents [12, 14] using social norms that prohibit, obligate, and give permission for actions that would prompt the effective interaction of a social group of agents in a multi-agent system [3, 10].

For example, in a traffic scenario, if vehicles are considered as agents, if an ordinary vehicle is aware of the norm of giving priority to emergency vehicles, this will avoid accidents.

While it is important to coordinate the system behaviour by applying norms, it is essential to ensure that the norms appliance does not affect the achievement of the objectives of the agents. For example, if the system objective is to minimise the average waiting (stopping) time of vehicles, this objective should still be reachable while the norms are applied. Therefore, several recent works proposed various techniques for aligning norms and objectives. [1, 2] used reasoning techniques to guide the agents to align with objectives. [18] used formal argumentation techniques to reason about the system's objectives and norms. However, in these approaches, the agents need to have reasoning capabilities. In [14, 15], we developed a model that coordinates the system's objectives with norms using a utility-based approach. However, the model does not address dynamic objectives. For choosing the best set of norms aligned with specific objectives, [17] proposed using quantitative approaches (e.g., optimisation technique) and [16] proposed using qualitative techniques (e.g., ranking technique). In contrast, [11] finds the objective with the best performance based on pre-defined norms. These approaches have a significant disadvantage of matching norms with a single objective or having a subset of preferred objectives. However, in reality, all objectives need to be aligned (coordinated) with the whole societal norms regardless of their internal compatibility. Moreover, these models do not consider heterogeneous environments where agents may support varying objectives.

Despite these efforts, there is a gap in having a technique for aligning and reformulating norms and objectives simultaneously in a dynamic environment where agents, network, and situations keep changing. Operating in such an ever-changing environment, agents need to evolve their objectives to cope with the unseen situations [8], and adapt their norms and behaviour to match these changes [14, 15]. For example, in the context of a traffic network, if roadworks occur on one of the roads, the system's objective can evolve to minimise the number of vehicles travelling on this road. Subsequently, norms can change at this time to avoid entering this road.

To solve these problems, we introduce a Dynamic Objectives and Norms Synthesizer and Reasoner (DONSR) model. DONSR represents a normative multi-agent system that is responsible to: (1) Operate in a dynamic environment with unseen situations. (2) Reason objectives and reformulate the objective set based on the changing situations online. (3) Synthesise efficient norms online. (4) Ensure that the process of objectives reasoning does not affect the process of the norms synthesising and appliance and their effectiveness. (5) Align multiple norms with the evolving objectives formulated from the unseen situation. To reach these previously listed goals, DONSR includes the following components:

- An Objective Reasoner Component: which is responsible for reasoning the changes and deciding whether to change the objective, leave it, or create a new one.

- An Objective Formulator Component: which uses backward reasoning to create new objectives [when needed] when an unseen situation occurs.
- A Norm Synthesizer Component: which is responsible for online norms synthesising using case-based reasoning technique.
- A Norm Reasoner Component: which is responsible for aligning the objectives with norms, using a utility-based technique that transforms the current objective chosen by the objectives reasoner component to decide which norm to apply in case of multiple applicable norms.

The remainder of this paper is as follows. Section 2 covers the relevant background related to the techniques used to formulate DONSR. In Sect. 3, an example of a dynamic normative multi-agent system is stated to assist in elaborating DONSR and to be used as the evaluation scenario. DONSR is illustrated in detail in Sect. 4 and then evaluated in Sect. 5. Finally, the conclusion is covered in Sect. 6.

2 Background

2.1 Ontology

Ontologies provide machine-understandable semantics and augment human intelligence [6]. An ontology describes concepts C , properties P , relationships R in a specific environment [20]. It is possible to use a relationship for a particular type of instance (domain) with a particular value (range). An inference rule is an implication of the form: If J_1, J_2 up to J_n are inferable, then J is inferable (see Eq. (1)). Using Semantic Web Rule Language (SWRL), ontology engineers express the inference rules manually [9].

$$J_1, J_2, \dots, J_n \rightarrow J \quad (1)$$

In **forward reasoning**, state observations are used as inputs, and inference rules are applied to extract additional facts until the goal is reached. For example, we can conclude from “A” and “A implies B” to “B”. **Backward reasoning** is based on starting with the goal and chaining through inference rules to find the facts that support it. For example, we can conclude from “not B” and “A implies B” to “not A”.

2.2 Case-Based Reasoning

Case-based reasoning [13] algorithm defines new problems (situations) as cases, and then it searches for similar cases collected from old experiences to find the best solution that was used before.

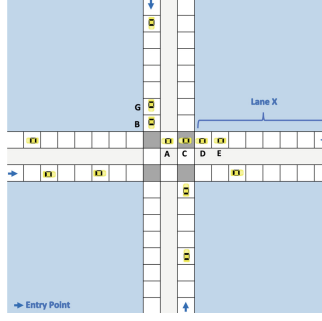


Fig. 1. Traffic grid

3 Running Example (Traffic Scenario)

To facilitate the illustration of concepts in our proposed model and for evaluation purposes, a traffic scenario is used. This scenario includes two main roads with 4 intersections, each with two lanes in opposite directions (see Fig. 1). In this scenario, the vehicles are modelled as agents, and the norms are used to avoid accidents, e.g., a norm is created to avoid going (moving forward) if there is another vehicle in front. The vehicles are of two types, emergency vehicles (e.g., ambulance, police vehicle) and ordinary vehicles. We assume that the intersections are unsignalized, and a traffic manager guides the vehicles to decide their subsequent actions based on the environment. To model the concepts in the traffic environment, we use the ontology shown in Fig. 2. The traffic manager tries to avoid accidents by communicating the synthesised norms to vehicles and, at the same time, aims to reach its current objective. The traffic manager's objectives can be to:

- **(Objective 1) minimise the average waiting time of vehicles:** minimising vehicles' waiting (stopping) time is the default objective of the system, as it is used to avoid congestion and maximise the flow of vehicles.
- **(Objective 2) minimise the waiting time in a specific lane:** This might be the case when a road is a bottleneck, and it is downstream, so we would like to minimise the queue length in it.
- **(Objective 3) minimise the waiting time of emergency vehicles.** When an incident happens, we are expected to minimise the waiting time of emergency vehicles so that they can get to the incident location as soon as possible.

4 DONSR: Dynamic Objectives and Norms Synthesizer and Reasoner Model

We propose the Dynamic Objectives and Norms Synthesizer and Reasoner (DONSR) Model to represent dynamic normative multi-agent systems. DONSR

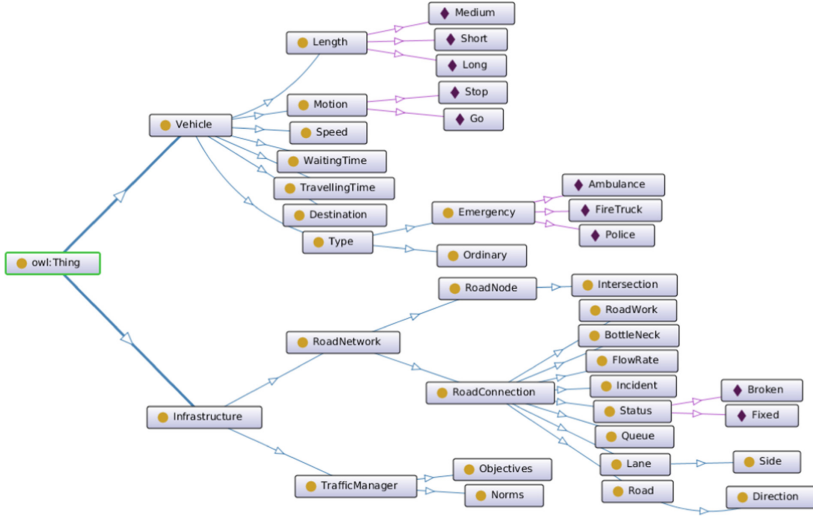


Fig. 2. Ontology for traffic environment, as represented by OntoGraf.

aims to enable online norms and objectives synthesising and reasoning, in addition to aligning the synthesised norms and objectives, and ensuring that none of the processes negatively affect the other processes' effectiveness. To reach this, DONSR carries out three main functionalities in every time-step: (A) Objectives reasoning and formulation: this is the process of reviewing the current objective and changing it if required (Algorithm 1). (B) Norms synthesising: this is the process of creating a new norm if a new 'behaviour' conflict was detected. For example, in the traffic scenario, accidents will be the result of behaviour conflicts, and when DONSR detects a new accident, it will create a new norm (Sect. 4.3). (C) Norm reasoning: this process takes place when there are unmatchable norms. Unmatchable norms is the result of having two(or more) applicable norms that can be applied in the same context, however, their application would result in a conflict. For example, in the traffic scenario, in Fig. 1, if there are two norms defined, n1: stop when there is a vehicle on the right of the intersection. n2: stop when there is a vehicle on the left of the intersection. If both vehicles A and B apply n1 and n2, respectively, both vehicles would not move, resulting in a deadlock. In this case, the norm reasoning process takes place to decide which of these norms (n1 and n2) to be applied and which to ignore (Sect. 4.4). DONSR meets functionalities (A), (B), and (C) using four main components coloured in grey in Fig. 3. The components are:

4.1 The Objective Reasoner Component

We assume the traffic manager models its observation using a schema described by an ontology Ont_D . This schema is used when semantic descriptions are needed (e.g., to explain unexpected events) and is composed of surrounding concepts and

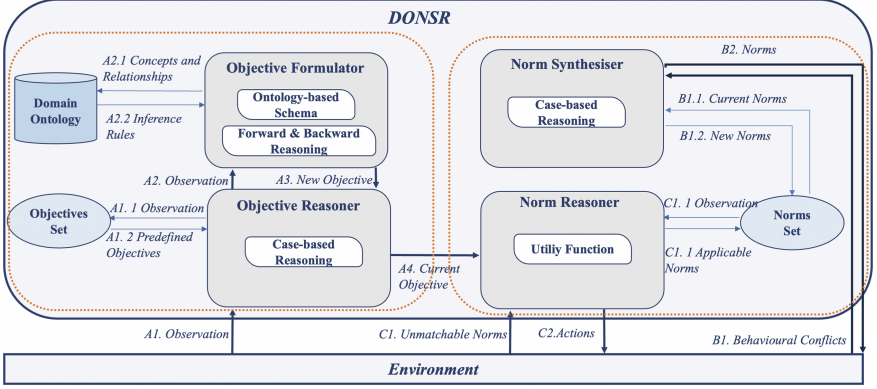


Fig. 3. Dynamic objectives and norms synthesizer and reasoner model

relationships between concepts perceived by the traffic manager. For instance, the concept “Vehicle” can be defined along with the relationship “hasWaiting-Time”. These relationships enable automated reasoning.

The traffic manager has an objective set O in the form of (if situation X happens \rightarrow objective Y should be applied). The traffic manager continuously reasons about the objectives it is pursuing, and when an objective needs to be changed or generated, two cases are possible [8]:

- **Choosing a predefined objective.** Using a case-based reasoning technique, the traffic manager can choose a predefined objective o_c from an objective set based on its current observation obs_i (see lines 6–7 of Algorithm 1).
- **Creating a new objective.** In the absence of a suitable objective in an objective set, the traffic manager uses backward reasoning over inference rules to create a new objective (see lines 8–10 of Algorithm 1).

4.2 The Objective Formulator Component

When an unseen situation is observed, and the traffic manager can not find a suitable objective from an objective set, it uses backward reasoning over inference rules to extract new objectives. The ontology-based schema may include the relationship “hasFlowRate($?r$, Decreased)”. It means there is a decreasing flow rate in the road r . Also, the traffic manager has no predefined objective to handle this new situation. However, inference rules may declare that “hasEnteringVehicles($?r$, Increased) \rightarrow hasFlowRate($?r$, Decreased)”. So to increase the flow rate (as a new objective), the traffic manager needs to decrease the number of entering new vehicles into the road (hasFlowRate($?r$, Increased) \rightarrow hasEnteringVehicles($?r$, Decreased)). That means that a traffic manager can add the new objective “if road r has decreased flow rate \rightarrow the objective is minimising the number of vehicles entering the road r ” to the set of objectives, although that was not part of the original one. One can also say that the new objective was “discovered via inferencing”. We have given two

Algorithm 1. Objectives reasoning & formulation

```

1: for each  $t$  do
2:   Input:  $O, Ont_D, Obs$ 
3:   Output:  $o_c$ 
4:   for each  $obs_i \in Obs$  do
5:      $case \leftarrow obs$ 
6:     if  $case_i \in Cases$  then
7:        $o_c \leftarrow sol_{case_i}$ 
8:     else //comment: Objective Formulator
9:        $IR \leftarrow Query(obs)$ 
10:       $o_c \leftarrow Reason(IR)$ 
11:     end if
12:   end for
13: end for

```

examples of unseen situations and their relevant inference rules in the traffic environment in the following.

- **Example 1:** When a bottleneck happens in a specific road, the traffic manager infers that to reduce the bottleneck in road $r1$, the waiting time of all instances of vehicle b on the road $r1$ should be decreased. This is reached by applying backward reasoning over the inference rules shown in Table 1. Afterwards, the traffic manager uses this inference to create a new objective in the objective set: if road r has bottleneck – $>$ the objective is minimising the waiting time of vehicles on the road r .

Table 1. An example of inference rules, inferring minimising the waiting time of all instances of vehicle b through backward reasoning.

Inference rules
TrafficManager(?i), Intersection(?s), Road(?r), Lane(?l), Vehicle(?b), isOn(?b, ?l), consistOf(?r, ?l), hasWaitingTime(?b, Increased)
– $>$
hasBottleNeck(?r)

- **Example 2:** Suppose an unseen situation occurs when ambulance a enters intersection s , according to the inference rules shown in Table 2, the traffic manager infers through backward reasoning that it should minimise the waiting time of ambulance a until it passes through the intersection. Then, the new objective is added to the traffic manager's objective set: if the vehicle a with unknown type is at the intersection s – $>$ the objective is minimising the waiting time of vehicle a .

4.3 The Norm Synthesizer Component

The Norm Synthesizer component monitors behavioural conflicts (accidents in case of the traffic scenario) and uses it to create norms. We used the norms

Table 2. An example of inference rules, inferring minimising the waiting time of the ambulance a through backward reasoning.

Inference rules
TrafficManager(?i), Intersection(?s), Road(?r), Lane(?l), Vehicle(?a), isOn(?a, ?l) consistOf(?r, ?l), hasType(?a, <i>Emergency</i>), hasWaitingTime(?a, <i>Increased</i>) – > atIntersection(?a, ?s)

synthesising algorithm used in [15], which is based on case-based reasoning. The norms synthesizer checks the accidents (behavioural conflicts) at each time-step and checks if it is similar to a previous case (context of the accident and action taken primary to the accident). The same solution is implemented if an identical case was found with a successful solution. If it is a new case, a new norm is created. A norm is defined as $n_i = (\alpha, \theta(a_i))$, where α is the pre-condition that should exist for the norm to be applied, while θ is a denotic operator to be applied on an action a_i . A denotic operator is either prohibition, obligation, or permission. In the traffic context, α will include the directions of the neighbouring vehicles in the three cells in front of the reviewed vehicle. For example, in Fig. 1, to define a norm for Vehicle A to prohibit its movement in its current context to avoid accidents, the norm will be $n_a = (left(<), front(-), right(<), Proh(Go))$.

4.4 The Norm Reasoner Component

This component resolves the unmatchable norms problem, and aligns the norms and the objectives. When there are two applicable unmatchable norms in the same context, the Norm Reasoner is responsible to decide which norm to apply and which to ignore. For example, if both vehicle A and B will apply the norms $n_a = (left(<), front(-), right(<), Proh(Go))$, and $n_b = (left(<), front(-), right(-), Proh(Go))$ respectively in the next time-step, the Norm Reasoner calculates the utility gained from applying each of the norms, and apply the norm that gives the highest utility. The utility calculated is not only concerned with the directly benefiting agents (Vehicle A & B) from the decision, but also the utility of indirect agents (Vehicle C, D, D, E & G) is calculated. This approach of calculating direct and indirect utility is named 'Accumulated Utility' in [15]. To align the norms and the objectives, the utility is constructed based on the current objective specified by the Objective Reasoner (Sect. 4.1), and sent to the Norm Reasoner Component in step A4 in Fig. 3. We used the same technique used in [15] for converting objectives to the utility function, in which the utility is calculated by getting the inverse of the minimisation objective and the exact value of the maximisation objective. In the traffic example, the utility of each of the four objectives in Sect. 3 sequentially will be:

$$u_1 = -1 * ((wt_e + wt_{ord})/|V|) \quad (2)$$

wt_e : waiting time of emergency vehicles

wt_{ord} : waiting time of ordinary vehicles

$|V|$: number of vehicles

$$u_2 = -1 * wt_{LaneX} \quad (3)$$

wt_{LaneX} : waiting time at of vehicles at lane X

$$u_4 = -1 * wt_e \quad (4)$$

In DONSR, to ensure that none of the norms processes or the objectives processes affect each other, we implement each process in separate components. As represented in Fig. 3 by the orange dotted frames, none of the objectives' components interfere with the norms' components except to notify the Norms Reasoner of the new objective, if the old objective evolved. However, to align the norms and the objectives, we build the utility function used in the Norms Reasoner based on the system's objective to ensure its achievement as well.

5 Empirical Evaluation

5.1 Simulated Environment

We simulate the traffic scenario (in Sect. 3) represented in Fig. 1 by a 19×19 grid using SUMO [19]. The ratio of creating emergency vehicles compared to the number of ordinary vehicles is 12:100. The destination and route of the vehicles were chosen randomly by the simulator while they are created. Every time-step, the simulator prepare (2 to 4) vehicles to start their trip, however, only if there are available entry points they can enter (indicated with blue arrows in Fig. 1). In each time-step, the vehicle can only move 1 cell *Go* or *Stop*.

5.2 Experimental Scenarios

We simulated four scenarios and compared the results with UNS [15], which uses a utility-based approach for aligning norms and objectives, but does not detect environmental changes and evolve objectives. The used utility-function in the current evaluation of UNS is based on minimising the average waiting time (objective 1). In our four scenarios as well, the default objective is minimising the average waiting time of vehicles. However, depending on the scenario, if a change was recognised in the environment, DONSR starts its objective reasoning process and may change the objective accordingly. Also, if an ambulance was seen at an intersection (at any of the scenarios), the objective changes at this time-step (at this intersection only) to minimise the waiting time of emergency vehicles (objective 3). Accordingly, The norm reasoner will use u_2 defined in Eq. 3 in the case of norms reasoning. The simulated scenarios are:

- **Scenario A:** This is the basic scenario, only two objectives (objective 1 and objective 3) are used. objective 3 is applied whenever an ambulance is recognised and gets back to the default objective (objective 1) afterwards.

- **Scenario B:** In this scenario, the default objective changes to objective 2 from time-step 500 to 1000 only at the intersection following Lane X (Check Fig. 1).
- **Scenario C:** The objective used for the intersection at the end of Lane X in Fig. 1, changes every 300 steps, switching between the default objective and objective 2.
- **Scenario D:** This is a faster version of scenario C, where the objective used for the intersection at the end of Lane X, changes every 50 steps, switching between the default objective and objective 2.

5.3 Results

Run-Time Norm Synthesising of Efficient Norms. Figure 4 shows the ability of DONSR to synthesise norms, through reflecting how the effectiveness of norms resulted in zero collisions after the norms set was synthesised. Moreover, it can be seen in all scenarios, even in scenario D (in which the objectives are changing while the norms are still being synthesised), how the norms synthesising process is not affected by the objectives changing process.

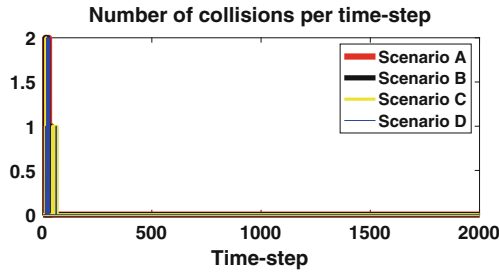


Fig. 4. Number of collisions per time-step

Objective 1: Minimising the Average Waiting Time of Vehicles. This is the default objective used in all of the scenarios and in UNS. As it is used in DONSR interchangeable with other objectives, it is important to analyse the extent its original performance was negatively affected. As seen in Table 3, the highest effect, although insignificant, is in Scenario C and D, which is expected as objective 2 was used more frequently compared to Scenario B (which used it from time-step 500 to 1000 only). Nevertheless, in Scenario A, where this objective was the default objective, and objective 3 was only applied when an ambulance is at the intersection, the average waiting time was improved by 0.093% compared to UNS.

Objective 2: Minimising the Waiting Time in Lane X. In Table 3, the waiting time in Lane X was decreased in the scenarios that involve objective 2 (Scenario B, C and D). Scenario C and D have higher improvement compared to B, as scenario C and D used it as the main objective to be applied several times, while in scenario B objective 2 was used only once between time-step 500 and 1000.

Table 3. Objectives results comparison versus UNS

Objective	UNS	Scenario A	Scenario B	Scenario C	Scenario D
Average waiting time of all vehicles (Obj1)	49.432	49.386	49.331	49.580	49.580
Improvement in Obj1	–	0.093%	0.205%	–0.299%	–0.299%
Average waiting time in Lane X of all vehicles (Obj2)	1.955	1.978	1.683	1.597	1.564
Improvement in Obj2	–	–1.165%	13.917	18.327%	20%
Average waiting time of emergency vehicles (Obj3)	48.308	45.666	46.669	47.893	45.037
Improvement in Obj3	–	5.468%	3.393%	0.858%	6.771%

Objective 3: Minimising the Waiting Time of Emergency Vehicles. As seen in Table 3, all of DONSR scenarios improved the waiting time of emergency vehicles compared to UNS, because it is assumed that UNS can only apply a fixed utility, which is assumed to be formulated based on objective 1 (minimise the average waiting time) only.

6 Conclusion

In this paper, we proposed DONSR, a novel Dynamic Objectives and Norms Synthesizer and Reasoner model, used for run-time norms and objectives alignment, synthesising, and reasoning. DONSR aims to operate in a dynamic environment in which new situations appear that can result in changing objectives. In such an environment, DONSR can formulate new objectives, if needed, using ontology-based schema, forward reasoning, and backward reasoning. Moreover, DONSR synthesises norms and reasons objectives online depending on the situation using case-based reasoning. Furthermore, DONSR ensures that objectives are aligned when applying norms by using utility functions constructed based on the system’s objectives. We evaluated DONSR with several traffic scenarios with different changing objectives. Results showed that DONSR was able to synthesise effective norms that can avoid collisions, evolve three objectives, and further reach these objectives. As future work, we look forward to defining a decentralised mechanism in which the objectives and norms of the agents are reasoned by the agents themselves, in addition to DONSR (central unit).

References

1. Aydoğan, R., Kafali, Ö., Arslan, F., Jonker, C.M., Singh, M.P.: NOVA: value-based negotiation of norms. *ACM Trans. Intell. Syst. Technol. (TIST)* **12**(4), 1–29 (2021)
2. Bench-Capon, T., Modgil, S.: Norms and value based reasoning: justifying compliance and violation. *Artif. Intell. Law* **25**(1), 29–64 (2017). <https://doi.org/10.1007/s10506-017-9194-9>
3. Cranefield, S., Savarimuthu, B.T.R.: Normative multi-agent systems and human-robot interaction (2021)

4. Dorri, A., Kanhere, S.S., Jurdak, R.: Multi-agent systems: a survey. *IEEE Access* **6**, 28573–28593 (2018)
5. Edenhofer, S., Stifter, C., Madkour, Y., Tomforde, S., Kantert, J., Müller-Schloer, C., Hähner, J.: Bottom-up norm adjustment in open, heterogeneous agent societies. In: 2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W), pp. 36–41. IEEE (2016)
6. Fong, A.C.M., Hong, G., Fong, B.: Augmented intelligence with ontology of semantic objects. In: International Conference on Contemporary Computing and Informatics (IC3I), pp. 1–4. IEEE (2019)
7. Ghanadbashi, S., Golpayegani, F.: An ontology-based intelligent traffic signal control model. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 2554–2561. IEEE (2021)
8. Ghanadbashi, S., Golpayegani, F.: Using ontology to guide reinforcement learning agents in unseen situations. *Appl. Intell.* **52**(2), 1808–1824 (2022). <https://doi.org/10.1007/s10489-021-02449-5>
9. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosz, B., Dean, M., et al.: SWRL: a semantic web rule language combining OWL and RuleML. *W3C Member Submission* **21**(79), 1–31 (2004)
10. Mashayekhi, M., Ajmeri, N., List, G.F., Singh, M.P.: Prosocial norm emergence in multi-agent systems. *ACM Trans. Auton. Adapt. Syst. (TAAS)* **17**(1–2), 1–24 (2022)
11. Montes, N., Sierra, C.: Value-guided synthesis of parametric normative systems (2021)
12. Morales, J., Wooldridge, M., Rodríguez-Aguilar, J.A., López-Sánchez, M.: Off-line synthesis of evolutionarily stable normative systems. *Auton. Agent. Multi-Agent Syst.* **32**(5), 635–671 (2018). <https://doi.org/10.1007/s10458-018-9390-3>
13. O’Mahony, E., Hebrard, E., Holland, A., Nugent, C., O’Sullivan, B.: Using case-based reasoning in an algorithm portfolio for constraint solving. In: Irish Conference on Artificial Intelligence and Cognitive Science, pp. 210–216 (2008)
14. Riad, M., Golpayegani, F.: A Normative multi-objective based intersection collision avoidance system. In: Jezic, G., Chen-Burger, Y.H.J., Kusek, M., Sperka, R., Howlett, R.J., Jain, L.C. (eds.) *Agents and Multi-Agent Systems: Technologies and Applications 2022. Smart Innovation, Systems and Technologies*, vol 306. Springer, Singapore (2022). https://doi.org/10.1007/978-981-19-3359-2_25
15. Riad, M., Golpayegani, F.: Run-time norms synthesis in multi-objective multi-agent systems. In: Theodorou, A., Nieves, J.C., De Vos, M. (eds.) *International Workshop on Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems*, pp. 78–93. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16617-4_6
16. Serramia, M., Lopez-Sanchez, M., Rodriguez-Aguilar, J.A.: A qualitative approach to composing value-aligned norm systems. In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1233–1241 (2020)
17. Serramia, M., López-Sánchez, M., Rodríguez-Aguilar, J.A., Morales, J., Wooldridge, M., Ansotegui, C.: Exploiting moral values to choose the right norms. In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 264–270 (2018)
18. Shams, Z., Vos, M.D., Oren, N., Padget, J.: Argumentation-based reasoning about plans, maintenance goals, and norms. *ACM Trans. Auton. Adapt. Syst. (TAAS)* **14**(3), 1–39 (2020)
19. SUMO: www.eclipse.org/sumo/

20. Zouaq, A., Nkambou, R.: A survey of domain ontology engineering: Methods and tools. In: Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds.) *Advances in Intelligent Tutoring Systems*, pp. 103–119. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-14363-2_6

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Computational Phenotyping of Decision-Making over Voice Interfaces

Lili Zhang^{1,2}(✉) , Ruben Mukherjee², Piyush Wadhai², Willie Muehlhausen²,
and Tomas Ward^{1,2}

¹ Insight SFI Research Centre for Data Analytics, Galway, Ireland

² Dublin City University, Dublin, Ireland

lili.zhang27@mail.dcu.ie

Abstract. Research on human reinforcement learning and decision-making behaviour has traditionally used visual-based symbols and graphics in the experimental paradigms. Such research leads to improved understanding of human decision-making and has application in fundamental research in cognitive neuroscience. In clinical domains, the approach holds out the possibility for the development of computationally-derived biomarkers suitable for use in psychiatry. Scaling this experimental approach through pervasive computing can help create larger datasets which will be necessary for normative studies. This will require the expansion of these experimental approaches beyond conventional visual representations. People receive information and interact with their environments through various senses. In particular, our sense of hearing in conjunction with speech represents a ubiquitous modality for learning and for updating our knowledge of the world. Consequently, it represents an important path for the investigation of human decision-making which is now experimentally accessible via rapid advances in voice-enabled intelligent personal assistants (IPAs). Examples include Amazon's Alexa technology and Google's Voice Assistant. However, to date no studies have demonstrated the feasibility of delivering such experimental paradigms over such voice technologies. Consequently in this study, we compared the performance of the same group of participants on the traditional visual-based and for the first time, a conversational voice-based, two-armed bandit task. Reinforcement learning models were fitted to the data to represent the characteristics of the underlying cognitive mechanisms in the task. Both model-independent behavioural measures and model-derived parameters were compared. The results suggest that participants demonstrated higher shifting rates in the voice-based version of the task. The computational modelling analysis revealed that participants adopted similar learning rates under the two versions of the interfaces, but more decision noise was introduced in the voice-based task as reflected by the decreased value of the inverse temperature parameter. We suggest that the elevated shifting rate is derived from the increased noise in the voice interface instead of a change in the learning strategy of the participants. Higher intensity of the control adjustments (click touch versus speak) might be one of the sources of noise, thus it is important to think further regarding the design of the voice interface if we wish to apply voice-enabled IPAs to measure human decision-making in their daily environments in the future.

Keywords: Reinforcement learning · Decision-making · Computational phenotyping · Voice interface

1 Introduction

Decision-making is a high-level cognitive process based on various cognitive processes like perception, attention, and memory [9]. The most conventional decision-making research in psychology and cognitive science generally begins with developing a theory or hypothesis about what should happen in pre-defined behavioral paradigms given to the subjects, in which the behavioral paradigms are designed to mimic cognitive operation in the real world. Economic and reinforcement learning theories have been widely applied to formalize the computations taking place in the brain in these simulated scenarios [8, 18]. Particularly, reinforcement learning has been a valuable framework widely used for understanding the underlying deficits of cognitive processes of patients with mental illnesses [1, 3, 10, 11, 15]. The computational parameters extracted from such theory-driven models have been utilized as a promising phenotyping tool for humans. More importantly, the phenotypes represented by the computational parameters can be linked to the activities of the neural substrates [13].

Human decision-making research has focused for a long time on the investigation of reinforcement learning and decision-making from visual input or symbolic information, although various forms of modalities have been examined in animal studies [2, 16]. Nevertheless, humans interact with their environment using a wide array of senses. In particular, much of the time, voiced-based natural language communication is the dominant modality for learning and making decisions in the real world, especially in social contexts. Investigating the influence of information presentation formats, e.g. visual versus voice, on reinforcement learning and decision-making processes is beneficial to various fields, such as human-machine interaction, cognitive science, psychiatry, economics, and marketing. The format of the information presented, which serves as input to human cognitive processes, may guide, constrain, and even determine cognitive behaviour [20]. Particularly, it is necessary to investigate how people learn and make decisions when they are provided with information via a voice interface based on natural language instead of visual symbols given the fact that apart from the visual-based interfaces, speech has become a more prominent way of interacting with automatic systems in the past few years. It is also a more natural interface for many users. Voice-enabled intelligent personal assistants (IPAs) like Amazon Alexa, Apple Siri, Google Assistant, and Microsoft Cortana, that use input such as the user's voice and context information to provide assistance is widely available on smartphones [19]. With the growth of intelligent personal assistants, the level of spoken interactions with technology is unprecedented. Home-based devices such as Amazon Echo, Apple HomePod, and Google Home are increasingly using speech as the primary form of interaction. Across industries, voice-enabled IPAs are assisting with customer service, technical support, scheduling tasks, and many other personalized services [5]. Voice interfaces have

been an indispensable part of our daily life. Importantly, the ubiquity of voice-based products makes it possible to capture human decision-making data under various contexts with high ecological validity.

However, the voice user interface has hardly been applied in the realm of cognitive and decision-making research. Given the same decision-making paradigm, will a subject behave differently interacting with conventional visual instructions and stimuli compared to the auditory version? Although in previous studies it has been shown that visual and auditory stimuli are processed differently at the input stage [12], it is unclear whether these differences result in a downstream effect in cognitive processing. Most of the findings to date stem from research on the impacts of modality effects. Penney [14] reviewed research on the effects of visual and auditory presentation on short-term retention of verbal stimuli, developing the separate streams hypothesis of modality effects. According to this theory of modality effects, there are separate processing streams for auditory and visually presented information in short-term memory. Accordingly, encoding as both visual and auditory representations improves the chances of successful retrieval, as both subsystems can be used to recover information. The classic example of using these modality effects is the educational practice of presenting information to-be-learned both graphically and with textual information through an auditory mode [7].

No study has ever tested whether people perform differently on reinforcement learning and decision-making tasks when interacting with the voice-enabled IPAs using natural language as compared to conventional visual-based interfaces. If the answer is in the affirmative, how does it influence performance? If, on the other hand, the characterization is equivalent, is it feasible to use these auditory-based paradigms such that they can be embedded into widely accessible IPAS for sampling the computational phenotypes that reflect the status of the general population, and for clinical applications, those with psychiatric conditions? The present study compares people's performance on conversational voice-based and visual-based two-armed bandit tasks in order to test if there are significant differences as a function of stimulus modality and to provide empirical evidence for the feasibility of adopting voice-enabled IPAs for ecologically sampled human decision-making phenotyping in future cognitive studies. Both the superficial behavioural measures and the cognitive processes represented by the reinforcement models were compared across the two versions of the task.

2 Method

2.1 Participants

This study was approved by the local ethics committee of the School of Computing, Dublin City University. The participants in this study were recruited through advertisements and poster boards on the university campus. A total of 30 participants participated in the experiment. The age range of the subjects was from 20 to 40 years.

2.2 Procedure

People who were interested in participating in the experiment were given an URL link to the study. After reading through the plain language statement and completing the informed consent in the first two pages of the link, the participants were first required to provide some basic demographic information. They were then directed to the gamified two-armed bandit task. Two interfaces for the task, i.e. a visual interface and a conversational voice interface, were developed (details about the task are introduced in the next section). The participant had to play both versions, one after the other, but the sequence of the two interfaces was randomly assigned across the subjects. A “wash-out” task was included between each task to reduce any after-effects. This was done by allowing the subjects to play a minesweeper game for 5 min. The workflow of the complete experiment is illustrated in Fig. 1.

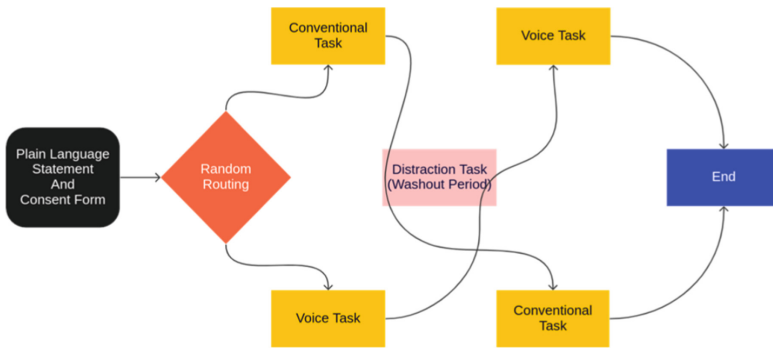


Fig. 1. Workflow diagram of the experimental procedure.

2.3 The Two-Armed Bandit Task

In order to make the task less monotonous, we placed the participants in a story-based scenario where they had to undertake a journey in a forest. In this journey, they would pass through crossroads, i.e. junctions and interact with two leprechauns distinguished by different colours standing at each junction. The participants are initially given 1,000 gold coins at the beginning of their journey through the forest. They have to navigate through the trees and bushes to reach home. As they make their way through the forest, they will come upon the series of junctions already described. At each junction, there will be two leprechauns, one with a blue colouring and the other coloured red, who may steal gold coins from the participant. Although unknown to the participants, they will have to pass through such junctions 120 times. The probability that a leprechaun will steal some gold coins fluctuates independently and slowly based

on a Gaussian distribution for each leprechaun. There is always one leprechaun who on average is less prone to stealing but that can change slowly over time. This leprechaun represents the most beneficial choice when selecting a leprechaun to go past. After choosing the leprechaun, the participants are provided with feedback indicating if they lost gold coins or not. The participants are instructed that the chance of the stealing leprechaun being blue or red depends only on the recent outcome history. The aim of the participants is to learn and choose the better leprechaun that steals less from them as more as possible to reserve more gold coins when they get back home.



Fig. 2. The visual-based two-armed bandit task. The participant is initially given 100 gold coins. In the first example trial (the first row), the blue leprechaun was chosen and it stole one gold coin from the participant and ran away. In the second example trial (the second row), the red leprechaun was chosen and the ‘Good choice!’ feedback was given. (Color figure online)

Two versions of the task were developed, i.e. the traditional visual-based version and the conversational voice-based version. A screenshot for the visual-based version of the task is shown in Fig. 2. This implementation is developed using HTML, CSS, and Javascript. The server is run on a Heroku instance with the data stored on a managed MongoDB service. A series of open-source libraries were utilized for the visual and voice features. There are no visual aesthetics in the conversational voice-based interface. The interaction was maintained over the entire system-participant interaction. Initially, the system narrated the scenario for the participant. The script of the instruction for the voice-based interface was exactly the same as that for the visual-based interface. Participants had to click or touch the leprechauns to make the selection in the visual version of the task, whereas they spoke out their choices, i.e. ‘blue’ or ‘red’, in response to the query from the voice interface. The lexicographic strategy was implemented even though the voice interface is able to recognize ‘the blue leprechaun’, ‘blue leprechaun’, ‘blue one’ etc., as long as the most-important attribute ‘blue’ and ‘red’ were included. Based on the subject’s response, whether the coins were lost or saved was calculated. The result was confirmed as a response to the subject on each iteration of coins lost or saved. In the visual version of the task, the good

feedback was shown as a text dialogue saying “Good choice!”, while the negative feedback was presented as the leprechaun taking the gold coins and illustrated as the it running away. In the voice version of the task, the participant was informed “Yay! Good selection!” for selecting the leprechaun if it did not steal coins and ‘Oops! Bad Selection.’ for selecting a leprechaun that stole gold coins on that round.

2.4 Wash-Out Task

A spatial memory test was developed to distract the participants during the wash-out period between the two versions of the two-armed bandit task. A screenshot of the spatial memory game is shown in Fig. 3.

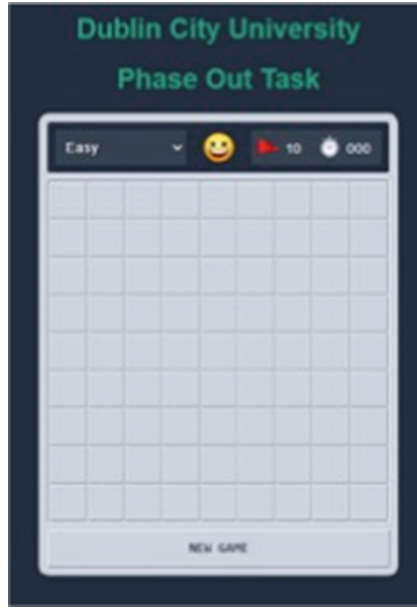


Fig. 3. The screenshot of the wash-out task.

2.5 Comparison of the Model-Independent Behavioural Measures

We firstly compared participants’ performance on the task in terms of superficial behavioural statistics that should capture fundamental aspects of learning, the probability of shifting to the other option, p_{shift} , and the probability of choosing the correct action $p_{correct}$. We calculated the probability of shifting after receiving a loss versus no loss, and the overall shifting rate as a function of the task version.

2.6 Computational Modelling Analysis

The Reinforcement Learning Model. The participant choice and the outcome (whether or not gold coins were lost) in each trial were recorded and the data fitted to a simple reinforcement learning model. It is assumed in this model that participants first learn the expected value of each leprechaun based on the history of previous outcomes and then use these values to make a decision about what to do next. The most classic model of learning is the Rescorla-Wagner learning rule [17] whereby the value of option k is updated in response to the loss p_t in trial t according to:

$$Q_{t+1}^k = Q_t^k + \alpha(p_t - Q_t^k) \quad (1)$$

where α is the learning rate, which ranges from 0 to 1 and captures the extent to which the aversive prediction error $p_t - Q_t^k$, updates the value. The p_t was encoded as -1 if there was a loss occurred and 0 if no loss was caused. The initial value for each of the options Q_0^k is assumed to be zero.

The simplest model of decision-making is to assume that participants choose the most valuable option. However, this assumption is not consistent with what is observed when people select between options. A basic property of findings on option selection is that people do not seem to always choose the better of two options. If they did, we would expect to see that their choices followed a step function, as long as one option has a higher value than the other, they always choose the former one and vice versa. Instead, people's choices follow a sigmoid-like pattern, more step-like when there is a large difference between the option values, but, as that difference narrows, people start to choose the higher-valued one with less consistency (i.e., they are increasingly likely to choose the "objectively" lower-valued option). Thus, one choice rule with these properties is known as the 'softmax' choice rule, which chooses option k with probability:

$$p_t^k = \frac{\exp(\beta Q_t^k)}{\sum_{i=1}^K \exp(\beta Q_t^i)} \quad (2)$$

where β is the inverse temperature parameter that controls the level of stochasticity in the choice, ranging from $\beta = 0$ to $\beta = \infty$. $\beta = 0$ represents the participant was completely randomly making the choices, whereas $\beta = \infty$ means they were deterministically choosing the option with the highest value.

Hierarchical Bayesian Estimation of Parameters. A hierarchical Bayesian procedure was used to estimate distributions over model parameters at both individual- and population-level for the two sets of datasets on visual-based and voice-based tasks separately. Specifically, each parameter was assigned an independent population-level distribution that was shared across participants for each dataset. The standard deviation for the population-level distribution was estimated separately for each parameter. Posterior distributions were estimated using Hamiltonian Monte Carlo with an NoU-Turn Sampler (HMC with NUTS) as implemented in Stan [4] via its RStan interface. The Gelman-Rubin index \hat{R}

(Rhat) was used to assess the convergence of the MCMC samples [6]. \hat{R} values close to 1.00 indicate that the MCMC chains have converged to stationary target distributions. There were no population-level parameters with R values greater than 1.1 (most were below 1.01). Four chains were run with 1000 warming up and 4000 samples each.

In order to examine the effects of the task modality on the underlying cognitive process, we compared the posterior distributions of the population-level parameters across the two versions of the task using the 95% Highest Density Interval (HDI). Specifically, we calculated the difference in the population-level parameters for the two datasets and reported the 95% HDI of the difference. If this HDI did not overlap zero, we consider there to be a meaningful difference between the performance on the two versions of the reinforcement learning task. The individual estimates for mean values of each parameter were also extracted and compared in order to examine the effects of the task version on each individual.

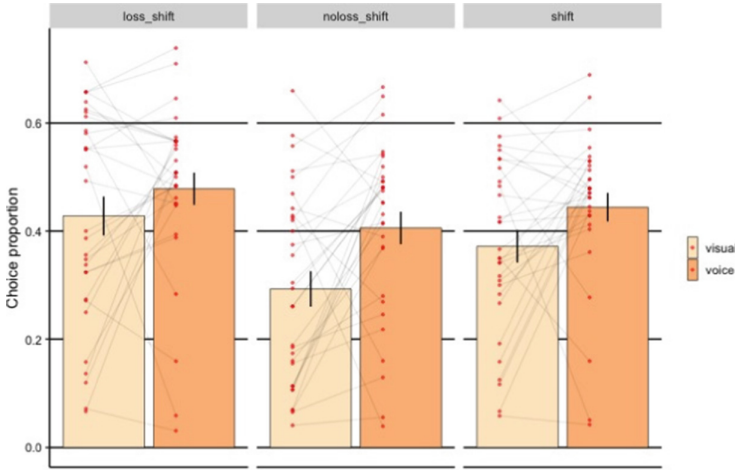


Fig. 4. The probability of shifting to the other option after receiving a loss (left) and no loss (middle), and the overall probability of shifting regardless of the outcome as a function of the task version (right). The probability of loss-shift was not significantly different in the two versions of the task, whereas the no loss-shift and the overall shift rate were significantly elevated in the voice-based version of the task. Each dot represents a participant and error bars represent 1 standard error of mean.

3 Results

3.1 Comparison of the Model-Independent Behavioural Measures

Three model-independent behavioural measures were compared between participants' performance on the visual-based and the voice-based interfaces. The probability of loss-shift, no-loss-shift and shift regardless of the outcome across the trials for each participant was calculated for the visual-based and voice-based versions of the task, respectively and is shown in Fig. 4. Each dot represents a participant and error bars represent 1 standard error of mean. Visually, the group mean of the loss-shift and no loss-shift of the participants in the voice-based task were both higher than that in the visual-based task. The paired t-test shows that the no loss-shift probability on the voice-based task was significantly increased compared to the visual-based version ($t = -1.11, p = 0.28$), whereas there is no significant difference in terms of the loss-shift probability between the two versions of the task ($t = -2.62, p = 0.01$). Additionally, the overall tendency to shift in the voice-based version of the task was marginally significantly increased ($t = -2.09, p = 0.05$).

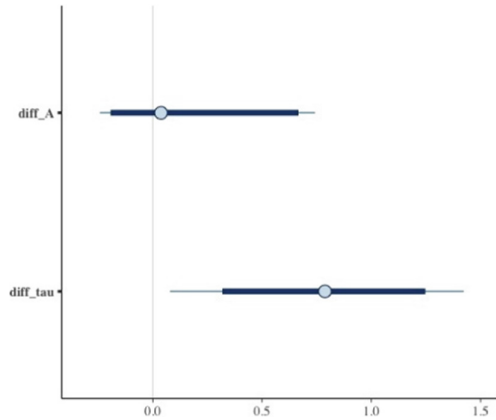


Fig. 5. The posterior means along with the 95% HDI for the difference of the group means of the learning rate diff_A and the inverse temperature diff_{τ} between the two versions of the task. The 95% posterior intervals excluded zero for the effect of the task version upon the learning rate parameter, whereas it included zero for the inverse temperature parameter.

3.2 Comparison of the Cognitive Parameters

Both the group- and individual-level free parameters, i.e. the learning rate and the inverse temperature, contained in the reinforcement learning model were

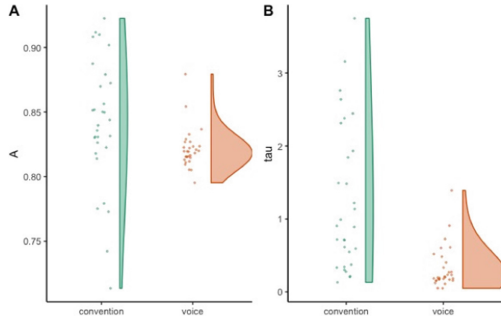


Fig. 6. The mean of the posterior distribution of the individual-level learning rate (left panel) and the inverse temperature parameter (right) for each participant on the visual-based versus voice-based version of the task. Each dot represents one participant.

estimated for the two versions of the task. In order to evaluate the influence of the task version on the overall performance, the difference of the group-level posterior distribution for the learning rate parameter diff_A and the inverse temperature parameter diff_{τ} between the visual-based task and the voice-based task was calculated and illustrated in Fig. 5. Participants adopted similar learning rates in the two versions of the task at the group level as the 95% HDI of the diff_A excluded zero. However, the group-level inverse temperature on the voice-based version of the task was significantly decreased compared to the visual-based task, indicating the participants were more deterministic in terms of choosing the option with the highest expected value on the visual-based task. Figure 6 demonstrates the mean of the posterior distribution of each individual parameter for each participant. The means of the posterior distributions of the individual-level parameters for the visual-based task were more decentralized compared to that for the voice-based task.

4 Discussion

The current study compares for the first time performance during voice and visual aversive two-armed bandit task conditions with an otherwise identical experimental protocol. Furthermore, this is the first time as far as the authors are aware that such a learning task has been conducted over a voice interface. Although the participants demonstrated equivalent loss-shift rates, the overall shifting rate and the probability of shifting in trials where no loss was caused were significantly elevated in the voice-based version of the task. The comparison of the underlying cognitive parameters revealed that participants adopted similar learning strategies for the two versions of the task, though more decision noise was present in the voice-based version of the task. The increased source of the decision noise may reflect the difference in terms of the format of the input information (visual versus auditory) impacts the overall weight given to the two options at that moment in the decision process. Another possible explanation

could be that responding to colour questions when no colours have been seen would be confusing and difficult for participants. A parallel task with questions suited to auditory modality (e.g. left versus right with stereo auditory input) would be useful in the future study. Additionally, the change of the control adjustments (click/touch versus speech) may also contribute to the alteration of the decision-making process. More noise might be included in the process when the outcome probability of each option and the intensity of control were evaluated simultaneously. Although efforts have been made to improve the efficiency of the system-subject interaction, use of the voice interface may be a more deliberate action for the participants in this experiment, especially given the fact that sometimes the participants needed to repeat their answers several times before the system identified what was said. We suspect that the elevated shifting rate in the voice-based version of the task may be the behavioural-level representation of the decision noise as reflected by the inverse temperature parameter given the learning rate on the two versions of the task was not significantly different. Overall, we anticipate future work in this area as natural speech interfaces present opportunities for human phenotyping based on learning behaviour in uncertain environments. In particular, given the ability to perform these experiments outside the laboratory, it is plausible that the human behaviour captured may be more representative of real-world behaviour and more valuable in terms of ecological validity.

5 Conclusion

The rapid advancement of voice-enabled IPAs provides opportunities to investigate how people learn and make decisions in the context of using natural language to communicate. It is necessary to examine if people perform equivalently on decision-making learning tasks when interacting with voice interfaces versus the conventional and to some degree validated approaches of texts and graphic stimuli. As such, these findings suggest that stimulus modality has no influence on the learning strategy in the reinforcement learning task, although more decision noise was introduced in the voice-based interface. These findings have implications for the presentation of reinforcement learning tasks in experimental settings. It is important for example to further enhance the efficiency and ease of interacting with the voice interface if we wish to use voice-based IPAs as sensors to measure human decision-making in their daily environments in the future. What is clear however is that characterisation of human behaviour in a way that may be useful for the derivation of computational biomarkers in the case of clinical applications for example, is possible over contemporary pervasive computing technologies.

References

1. Adams, R.A., Huys, Q.J., Roiser, J.P.: Computational psychiatry: towards a mathematically informed understanding of mental illness. *J. Neurol. Neurosurg. Psychiatry* **87**(1), 53–63 (2016)
2. Aluisi, F., Rubinchik, A., Morris, G.: Animal learning in a multidimensional discrimination task as explained by dimension-specific allocation of attention. *Front. Neurosci.* **12**, 356 (2018)
3. Brown, V.M., et al.: Reinforcement learning disruptions in individuals with depression and sensitivity to symptom change following cognitive behavioral therapy. *JAMA Psychiat.* **78**(10), 1113–1122 (2021)
4. Carpenter, B., et al.: Stan: a probabilistic programming language. *J. Stat. Softw.* **76**(1) (2017)
5. Cohen, P., Cheyer, A., Horvitz, E., El Kaliouby, R., Whittaker, S.: On the future of personal assistants. In: Proceedings of the 2016 CHI Conference Extended abstracts on Human Factors in Computing Systems, pp. 1032–1037 (2016)
6. Gelman, A., Rubin, D.B.: Inference from iterative simulation using multiple sequences. *Stat. Sci.* **7**, 457–472 (1992)
7. Ginns, P.: Meta-analysis of the modality effect. *Learn. Instr.* **15**(4), 313–331 (2005)
8. Glimcher, P.W., Fehr, E.: *Neuroeconomics: Decision Making and the Brain*. Academic Press, Cambridge (2013)
9. Gonzalez, C.: 13 decision-making: a cognitive science perspective. *The Oxford Handbook of Cognitive Science*, p. 249 (2016)
10. Huys, Q.J., Browning, M., Paulus, M.P., Frank, M.J.: Advances in the computational understanding of mental illness. *Neuropsychopharmacology* **46**(1), 3–19 (2021)
11. Montague, P.R., Dolan, R.J., Friston, K.J., Dayan, P.: Computational psychiatry. *Trends Cogn. Sci.* **16**(1), 72–80 (2012)
12. Paivio, A.: *Imagery and Verbal Processes*. Psychology Press, London (2013)
13. Patzelt, E.H., Hartley, C.A., Gershman, S.J.: Computational phenotyping: using models to understand individual differences in personality, development, and mental illness. *Pers. Neurosci.* **1**, e18 (2018)
14. Penney, C.G.: Modality effects and the structure of short-term verbal memory. *Mem. Cogn.* **17**(4), 398–422 (1989). <https://doi.org/10.3758/BF03202613>
15. Pike, A.C., Robinson, O.J.: Reinforcement learning in patients with mood and anxiety disorders vs control individuals: a systematic review and meta-analysis. *JAMA Psychiatry* **79**, 313–322 (2022)
16. Prichard, A., Chhibber, R., Athanassiades, K., Spivak, M., Berns, G.S.: Fast neural learning in dogs: a multimodal sensory fMRI study. *Sci. Rep.* **8**(1), 1–9 (2018). <https://doi.org/10.1038/s41598-018-32990-2>
17. Rescorla, R.A.: A theory of pavlovian conditioning: variations in the effectiveness of reinforcement and nonreinforcement. *Curr. Res. Theory* **2**, 64–99 (1972)
18. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. MIT Press, Cambridge (2018)
19. Weinschenk, S., Barker, D.T.: *Designing Effective Speech Interfaces*. John Wiley & Sons Inc, Hoboken (2000)
20. Zhang, J.: The nature of external representations in problem solving. *Cogn. Sci.* **21**(2), 179–217 (1997)


Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Latent Space Cartography for Geometrically Enriched Latent Spaces

Niall O' Mahony^(✉) , Anshul Awasthi, Joseph Walsh, and Daniel Riordan

Confirm Smart Manufacturing Research Centre, IMaR Research Centre,
School of Science Technology Engineering and Maths (STEM),
Munster Technological University, Kerry Campus, Tralee,
Co. Kerry, Ireland
niall.omahony@mtu.ie

Abstract. There have been many developments in recent years on the exploitation of non-Euclidean geometry for the better representation of the relation between subgroups in datasets. Great progress has been made in this field of Disentangled Representation Learning, in leveraging information geometry divergence, manifold regularisation and geodesics to allow complex dynamics to be captured in the latent space of the representations produced. However, interpreting the high-dimensional latent spaces of the modern deep learning-based models involved is non-trivial. Therefore, in this paper, we investigate how techniques in Latent Space Cartography can be used to display abstract and representational 2D visualisations of manifolds.

Additionally, we present a multi-task metric learning model to capture in its output representations as many metrics as is available in a multi-faceted fine-grained change detection dataset. We also implement an interactive visualisation tool that utilises cartographic techniques that allow dimensions and annotations of graphs to be representative of the underlying factors affecting individual scenarios the user can morph and transform to focus on an individual/sub-group to see how they are performing with respect to said metrics.

Keywords: Latent space cartography · Geometrically enriched latent space · Disentangled representation learning · Fine-grained change detection · Multi-task metric learning

1 Introduction

Geometrically Enriched Latent Space refers to the encoding of meaningful information into the dimensions/distance definition of the intermediate latent space where the feature vectors, otherwise known as representations or embeddings,

This work was supported by the Science Foundation Ireland under Grant No. 16/RC/3918 (CONFIRM).

© The Author(s) 2023

L. Longo and R. O'Reilly (Eds.): AICS 2022, CCIS 1662, pp. 488–501, 2023.

https://doi.org/10.1007/978-3-031-26438-2_38

can be said to lie. This practice often draws from theory of information geometry and non-euclidean geometry to transform latent space so that inter-class relationships or distance-metric correlations can be made more apparent.

We propose that Latent Space Cartography (creating maps of the latent space of representations learned by neural networks) be used to observe the distribution of representations per each interacting factor in a system so that the influence of each variable on the respective outputs of interest may be better understood. Based on these understandings, prior information on these salient background variables may be exploited at the inference stage of the DML approach by using a clustering algorithm to improve classification performance. This research proposes such a methodology establishing the saliency of query background variables and formulating clustering algorithms for better separating latent-space representations at run-time.

The remainder of this paper is organized as follows. In Sect. 2, we introduce different applications where better understandings can be achieved from visualisations of geometrically enriched latent space. In Sect. 3, various techniques in latent space cartography that have been proposed by recent research are reviewed. In Sect. 4, we describe details of some of the techniques we have implemented towards encoding sub-group-specific patterns into latent space. Lastly, in Sect. 5, we conclude with a brief summary and discussions.

2 Applications

2.1 Characterisation of Biological/Physical Processes

Data representations play a crucial role in the statistical analysis of biological data for both data exploration, e.g. through visualization, or task-specific predictions where limited data is available and for automation, e.g. whiling down large amounts of data to the most interesting/anomalous parts to make it possible for a human to interact with the whole dataset. Toward the representation of protein sequences, [4] demonstrate that several contemporary machine learning practices yield suboptimal performance in characterising basic relations humans can identify manually, and demonstrate that taking representation geometry into account significantly improves interpretability and lets the models reveal biological information that is otherwise obscured. Disentangled Representation Learning (DRL) has also proven to be useful in medical diagnoses where [32] trained a SVM learns latent space to exploit the correlations among multi-modalities and demonstrated that multiple diversified classifiers working on said space improve the diagnosis performance.

Theoretical research interests related to modelling complex systems require, not only for system dynamics to be captured and detected by a model but also for these changes to fit with what we currently understand about the system, e.g., to comply with the equations we have derived. Incorporating domain knowledge can be hugely advantageous as the theoretical model provides guidance with which

an effective model is supposed to follow; it helps an optimised solution to be more stable and avoid over-fitting, it allows training with less data, it would be more robust to unseen data, and thus it is easier to be extended to applications with changing distributions [2]. However, this type of approach is only applicable to problems that have been studied extensively, as explaining the origin of change in terms of individual variables is generally a tough task unless the variables are independent.

Applications where theoretically grounded DRL has been implemented include climate change [27] and dynamic systems [26] and data networks [29]. These works implement techniques related to knowledge injection. Generally, they use an architecture based on graph networks to incorporate prior knowledge given as a form of partial differential equations (PDEs) over time and space. These PDEs can comprise very sophisticated mathematics, e.g., Lagrangian [17] and Hamiltonian mechanics [8].

2.2 Fine-Grained Change Detection

Fine-Grained Change Detection (FGCD) is the process of identifying differences in the state of an object or phenomenon where the differences are class-specific and are difficult to generalise. There are many applications requiring efficient, effective algorithms for reliably detecting variation, like remote sensing [22,33], surveillance [19] and healthcare [10].

By definition, FGCD requires an algorithm that can recognise change across a range of scenarios, where there are many underlying factors which vary dependent on subject/situation being observed. Many of the state-of-the-art technologies in computer vision and time series analysis, which leverage big data and deep learning, struggle to predict subtle deviations for each individual subject/situation below the resolution that generalising for the entire population allows. Therefore, instance-based learning approaches are more applicable. In particular, we focus on techniques that can be applied to the representations learned by artificial intelligence in multi-task, multi-modal, open-set and online learning settings.

Multi-Task Metric Learning (MTML) is one such approach where instead of trying to predict a single score, the output is mapped to space where you can observe subgroup relatedness or simultaneously learn multiple related tasks. Our research has converged towards such an approach because of the way it learns to map its output to a latent space and how this may be exploited to infer relationships between feature variability and auxiliary background information. We investigate how the mapping element of MTML may be exploited in situations where the salient features vary over time or due to changing underlying variables. The research problems considered by this research consist primarily of studies in the field of Fine-grained Visual Recognition FGVR. However, many of the techniques discussed are applicable to the representations learned by neural networks regardless of the nature of the input data. Fine-grained recognition problems are typical wherever biological subjects are concerned, not to mention the variability that can be introduced by abiotic factors. Examples of such variable features

include age/gender variations in human/animal subjects in classification tasks for medical/ecological studies [11, 14], seasonal/time-of-day variations in long-term datasets [7], and differences in lighting, surroundings and camera pose in systems that have to be deployed in a range of scenarios [28].

3 Related Work

Recently, interactive tools for visualising latent space have been developed, initially focusing on a specific domain and a narrow set of tasks, and even more recently, such interactive elements have been compiled into integrated tools. Latent space cartography [6] seeks to guide users through a comprehensive workflow that supports tasks common to latent spaces across various input data types and DRL algorithms. These tasks include changing the desired type and complexity of projection algorithms, querying, filtering and highlighting groups of embeddings and visualising the similarity of these groupings with attribute vector arithmetic [15].

3.1 Dimensionality Reduction

The interpretation of latent space often requires subtle and implicit domain knowledge, for which human judgment is essential. However, dimensionality reduction techniques are often essential for visualising multi-dimensional latent spaces as humans have difficulty in reasoning about space beyond three dimensions. Common projection methods include t-distributed stochastic neighbour embedding (t-SNE) and principal component analysis (PCA). T-SNE is popular for exploring very high-dimensional data and with data with many embedding groups if the perplexity of the output projection is interpreted appropriately [31]. Uniform manifold approximation and projection (UMAP) is another non-linear technique that better preserves inter-cluster relationships. These non-linear algorithms highlight cluster structures but can obscure linear relationships among points. PCA is a linear transformation and so preserves linear relationships [15], which might be beneficial if further inferences can be drawn from the relative distances between embeddings [21].

3.2 Latent Space Alignment

Latent space visualisations can seem arbitrary and not very meaningful when the dimensions of projections of the latent space are not aligned/scaled to important metrics specific to the application. The performance of DRL crucially determines the type and performance of the algorithm for delineating the separation between feature sets to a manageable number of dimensions.

Expressing representations in relation to familiar metrics can be useful in the visual evaluation of model performance by highlighting cases where there was an underlying pattern not explained by the primary tasks (e.g., scene change detection) of an DRL approach but due to some other ancillary variables (e.g.,

weather). This may be applied to DRL to reveal the interactions of background/ancillary variables by these variables to the axes of latent space/manifold visualisations, i.e., it may be useful to be able to tell why an object was classified to belong to a particular sub-class through observation of where that object lies on a space projection. We propose that by using interactive latent space cartography, which allows custom axes and colours according to selectable variables of interest, such relationships may become easily revealed [21].

If such auxiliary variables are known before inference, it may also be useful to narrow down the results to instances that are more likely in light of this new knowledge. This is known as knowledge injection and has been implemented in different ways depending on the type of DRL. Auxiliary knowledge can be encoded as sparse input to metric learning techniques, as rules for more accurate relation extraction in generative approaches [18], or to predict missing links in knowledge graphs [9, 25]. Alternatively, a clustering algorithm, e.g., k-means clustering, could be formulated taking as input the salient background variables and outputting a function that maps the latent space to valid classifications, thus maximising the inter-class variance.

3.3 Latent Space Overlays

Transformed space, colourisation, textured plot overlays, contour maps (equidistant lines) and interpolation paths can help make sense of the measure and progression of change in relation to meaningful metrics [6] and can also be useful in navigation tasks [24].

In [4], the representation and colours correspond to the representation of hierarchical relationships where internal nodes (ancestors) are depicted as branches encoded in black, leaf nodes (inferred aphylogenetic trees) in gray.

In [1], they compare and trace interpolants between representations in latent space and the demonstrate the effect of several interpretable ambient metrics on the shortest paths (geodesics) and demonstrate that their interpolant successfully avoids high cost regions on the data manifold, since it utilizes the high-level semantic information that is encoded into the ambient Riemannian metric space.

Equidistant lines around selected data points in the latent space. Equidistance refers to the observation space and illustrates how regarding the latent space as a Riemannian manifold can help to separate classes. Other techniques investigated by [3] include the encoding of embedding classes by colour and shading of the background according to the magnification factor in a MNIST dataset and effectively show the difference between Geodesic and Euclidean interpolation in the latent space.

4 Methodology

The problems to be solved in our application (the specifics of the metrics/labels will remain generalised in this paper) share common challenges around the recognition of subgroup features in dynamic unregulated environments for which

include the need to recognise objects never seen before during training, with non-IID instances, in real-time, on limited hardware, in such a way to align dependent information and predict various different types of data.

In working through these problems we have realised a unified metric learning approach which incorporates multi-task, geometry preserving and sparse metric learning and have applied the following steps:

1. The learned feature vector characteristic of the base network and its variants was passed to a metric/manifold learning framework developed using PyTorch [23].
2. A metric/manifold learning method was extended to work in a multi-task setting.
3. Riemannian optimisation was applied to impose orthogonality on manifold embedding features to preserve the geometry of the label space in the output representations.
4. The output representations were passed to a latent space cartography tool for transforming and visualising the representations for interpreting the changes with regard to specific label dimensions.

4.1 Metric Learning Setup

The metric learning method adopted from [12] uses triplet loss in conjunction with a ConvNet base network as described in [16] with the parameters: ResNet-v1-50 base network 12000 training iterations, 7000 iterations to weight decay, 24 examples per class group, 4 classes per batch, 128 dimension embeddings with a fully connected network head.

4.2 Multi-task Label Acquisition

We implement our experiments on our custom dataset which contains both 3D camera images and records 228 individuals. 9 annotations were assigned to each data point. Some of the annotation labels (3) are up to date at the time of image capture as they are fed from an automated system, while some (2) are provided by human experts on an intermittent basis (every 3 weeks) where missing values were replaced with the closest observation. The remaining 4 values indicate the forward and backward gradient of these expert-provided scores in comparison with the next nearest score for that subject in the past/future.

There are multiple sources of information available in the system application and our design is to minimize a multi-task loss which is a summation of the loss corresponding to each of these attributes. A class label matrix was generated to be stored with the data to contain 9 labels, 5 of which were discrete and 4 were continuous, i.e. the discrete labels are class labels, e.g. for the identification/grouping of individual subjects and the continuous labels are for health/state indicators/scores [20].

4.3 Triplet Mining

A challenge that has not been well explored in the literature is tuple sampling strategies for metric learning in a multi-label setting. We have discussed the importance of methods such as semi-hard triplet mining. Many of these sampling strategies can be quite computationally expensive as they require a forward pass of the network being trained to be run on every image in the dataset. This is why many approaches use mini-batches to fit the computational load within the limits of their machine. Many method randomly sample a set of images, assuming that a few random parameter vectors will represent certain elementary transformation operations like translation, scaling, rotation, contrast, and colorization.

We also randomly sample the training dataset for anchors and then iteratively sample for positives and negatives as follows: To form a set of triplets: $\mathcal{T} = \{(\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-)\}_{i=1}^{|\mathcal{T}|}$, each element of which consists of the following: i) \mathbf{x}_i , an arbitrary example with (\mathbf{y}_i) , which denotes the class label matrix assigned to \mathbf{x}_i . ii) \mathbf{x}_i^+ , another arbitrary example with $(\mathbf{y}_i^+) = (\mathbf{y}_i)$ for at minimum n elements in the label matrix. iii) \mathbf{x}_i^- , such that $(\mathbf{y}_i^-) = (\mathbf{y}_i)$ for less than n elements. The examples \mathbf{x}_i , \mathbf{x}_i^+ and \mathbf{x}_i^- are referred to as the *anchor*, *positive* and *negative* respectively.

This novel multi-task triplet mining strategy has a number of advantages when combined with manifold optimisation techniques which will be made clear as we now explain how it is integrated with the manifold optimisation step. It was found that enforcing too many labels to overlap ($n > 5$) impeded the convergence of the loss function as there was a limited number of anchor-positive pairs in the dataset. Allowing very little overlap of labels ($n < 3$) was also detrimental to performance (quantified by training loss) supposedly as it allows too much variance between training iterations.

4.4 Manifold Optimisation

Extra final layers were added to the output embeddings with dimensions appropriate for the manifold optimisation step. Due to the nature of constraints on our metric parameters, we leverage the optimization technique implemented by [5] called Riemannian Conjugate Gradient Descent (RCGD) to jointly learn the parameters for generating metric space representations and regularising them to lie on a manifold that imposes orthogonality between the elements of output feature vectors. This is made possible because manifold structure allows for a number of operations to be carried out which are advantageous in our application. The first is that operations on manifolds can be exploited for regularising the distribution of embeddings such that the model can be taught to have the divergence of embedding clusters be indicative of fine-grained shift in health/state indicators.

It is necessary to capture *intra-class variances* that may occur in our dataset, e.g. due to shift in health/state whether it be gradual and natural over the a period or sudden. Such variances can be captured by adding an orthogonality constraint to the Mahalanobis matrix used for distance estimation [13]. In Dutta

et al.’s application, the dataset is unlabelled and pseudo labels are used. However, our dataset is labelled with up to 9 labels, and so we guide our triplet mining strategy for multiple labels. The following constraints were used: learning rate = 0.01, step size = 50, gamma = 0.1 (see [5] for details) The motivation for this methodology is that the random change up in which parts of the label matrix match for each training iteration induces orthogonality between the clusters that are formed. This constraint ensures these relationships are apparent while also allowing the push-pull nature of metric loss to distribute embeddings that fall between class centres appropriately while also relying on notions of geometric similarity within the source data.

4.5 Loss Function

Similarly to [30], we implement shared representation layers to encode the task relatedness. This means that there are two main components to the loss function, metric loss and manifold loss. Firstly, the metric loss (triplet loss) has a push pull effect on embeddings in latent space dependent on whether they are positive/negative pairs which is determined upon some notion of similarity which we define in Sect. 4.3.

Secondly, a manifold loss learns a transformation matrix that is used to transform the latent space to a Riemannian manifold. The motivation for adopting the methodology of [5] is that enforcing orthogonality between multiple tasks allows natural clusters in the data to be detected, while also simultaneously learning from manual/system annotations where provided. Our application requires that the *intra-class variances* that may occur in our dataset, e.g. due to shift in health/state indication scores whether it be gradual and natural over a period or sudden. Such variances can be captured by adding an orthogonality constraint to the Mahalanobis matrix used for distance estimation. This constraint ensures these relationships are apparent while also allowing the push-pull nature of triplet metric loss function to distribute embeddings that fall between class centres appropriately relying on notions of geometric similarity within the data.

4.6 Latent Space Visualisation

To investigate how the mapping element of MTML may be exploited in situations where the salient features vary over time or due to changing underlying variables, an adaption of Latent-Space-Cartography-Tool by [15] was created. The features of the tool that allow the visualisation of fine-grained change in health/state indicators are shown in the figures below.

In order to visualise the embedding space in the first place, it is necessary to apply dimensionality reduction techniques such as PCA and TSNE to reduce the 128 dimensions to a 2D/3D plot as shown in Fig. 1. To see fine-grained change, i.e. specific to a individual/subgroup, it is necessary that the tool be able to filter out the desired embeddings for said individual/subgroup as shown with the selection on the left in Fig. 1. Plot dimensions can also be aligned with class label as made possible with our manifold alignment techniques, as shown

in Fig. 2. The tool was adapted to work with the SQL database our system maintains and also display the data available for each scenario upon overlaying its associated embedding with the mouse as shown in Fig. 1.

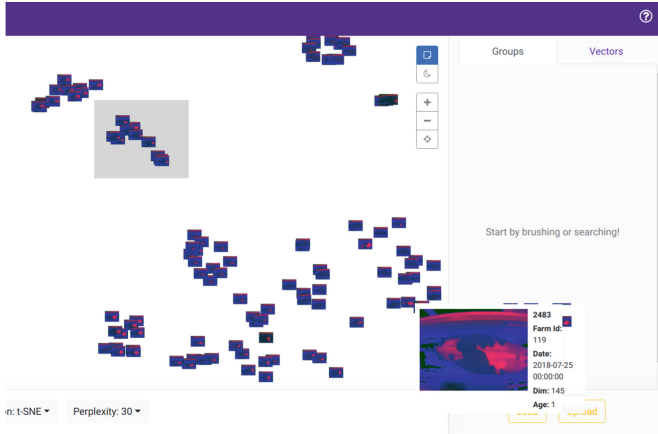


Fig. 1. Select/highlight/isolate by region interface on latent space cartography tool. Different projection algorithms and hyperparameters are selectable in the bottom left of screen. Label details for individual embeddings are also viewable upon selection.

It is difficult to benchmark these results given the bespoke nature of the given dataset other than comparing to works which have visualised/mapped similar problem sets. In that context our visualisations demonstrate an ability to identify fine-grained subgroups as was done by [5] and is visible in Fig. 3 where the smaller clusters correctly separate individual subjects (animals). Finally, we demonstrate how the tool can create interesting overviews with respect to various health/state indicators as the dimensions of the graph can be encoded/ set up to show the condition score metric and equidistant lines used to demonstrate how that metric diverges over the latent manifold. This technique has also been applied by [3] and similarly demonstrates how geometric distance is effective in showing the distribution of background factors.

This development is key to the system as it allows the selection of projection algorithms and hyperparameters which allow the latent space to be visualised in a way that reflects the scale and nature of relevant background variable allowing them to be exploited at the inference stage and also providing a better interface for the user that allows the deviation of individual/population/system health/state attributes to be visualised with greater precision.



Fig. 2. Search by label interface on latent space cartography tool. Note projection axes may also be displayed according to class label outputs (selectable in bottom right). As can be seen by the middle two clusters of historical embeddings for each entity (i.e. animals) that would have previously been classified as having the same label (e.g. condition score 3), the clusters stretch out indicating a shift in condition which would have otherwise gone unnoticed.

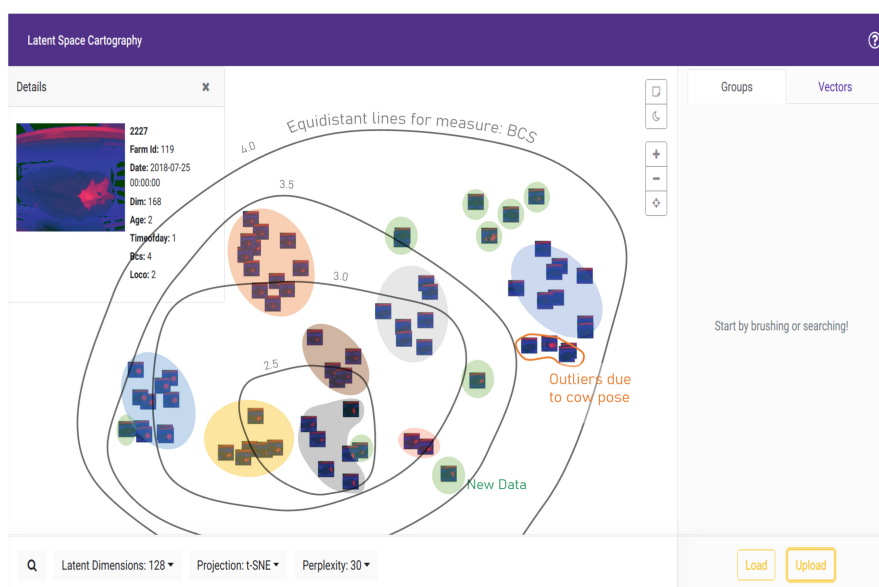


Fig. 3. As has been annotated in our FGCD dataset for Animal Health Monitoring, the embeddings are clustered according to one label and dispersed according to another label when the appropriate projection complexity and axes are selected. The distribution of the dispersion is visualised with equidistant lines.

5 Conclusion and Future Work

This paper has extended both the work in [20] in understanding how non-Euclidean geometries can be used to enhance the latent space of representations in system characterisation and FGCD applications. In summary manifold regularisation may be adapted/integrated to many types of learning architecture including supervised (metric) and unsupervised (generative) models and in particular our Multi-task Manifold/Metric Learning approach. The principles of Riemannian optimisation and the induction of non-Euclidean Geodesics into latent space has been shown to be useful in many application domains. The usefulness is subject to the resulting interpretability, however, which relies on tools for Latent Space Cartography including dimensionality reduction, latent space alignment and overlays.

This paper applies the methodology to an animal health monitoring application. The approach learns to map its output to a latent space which may be disentangled to infer relationships between feature variability and auxiliary background information. In this investigation, we proposed an approach to learn a discriminative distance metric along a manifold with the addition of a novel multi-task metric sampling technique. Also developed was an interface which used projection algorithms to allow the latent space to be visualised in a way that reflects the scale and nature of relevant background variable allowing them to be exploited at the inference stage. Regarding long-term adaption to changing conditions a SQL database management approach for the gallery of embeddings has been deployed so new observations can be compared with historical records for change detection. The addition of these aspects to the solution has helped identify important attributes relating to health/state of the subjects/systems of interest.

References

1. Arvanitidis, G., Hauberg, S., Schölkopf, B.: Geometrically enriched latent spaces. In: The Proceedings of Machine Learning Research, vol. 130 (2020). <http://arxiv.org/abs/2008.00565>
2. Borghesi, A., Baldo, F., Milano, M.: Improving deep learning models via constraint-based domain knowledge: a brief survey. arXiv (2020). <http://arxiv.org/abs/2005.10691>
3. Chen, N., Klushyn, A., Kurle, R., Jiang, X., Bayer, J., van der Smagt, P.: Metrics for deep generative models. In: International Conference on Artificial Intelligence and Statistics, AISTATS 2018, vol. 84, pp. 1540–1550 (2018)
4. Detlefsen, N.S., Hauberg, S., Boomsma, W.: Learning meaningful representations of protein sequences. *Nat. Commun.* **13**(1), 1914 (2022). <https://doi.org/10.1038/s41467-022-29443-w>
5. Dutta, U.K., Harandi, M., Sekhar, C.C.: Unsupervised deep metric learning via orthogonality based probabilistic loss. *IEEE Trans. Artif. Intell.* **1**(1), 74–84 (2021). <https://doi.org/10.1109/tai.2020.3026982>
6. Frenzel, M.F., Teleaga, B., Ushio, A.: Latent space cartography: generalised metric-inspired measures and measure-based transformations for generative models (2019). <http://arxiv.org/abs/1902.02113>

7. Germain, H., Bourmaud, G., Lepetit, V.: Efficient condition-based representations for long-term visual localization (2018)
8. Greydanus, S., Dzamba, M., Yosinski, J.: Hamiltonian neural networks. *Adv. Neural Inf. Process. Syst.* **32** (2019)
9. Gsponer, S., et al.: Background knowledge injection for interpretable sequence classification. In: *The 8th International New Frontiers in Mining Complex Patterns Workshop 2019*, Wüzburg, Germany (2019). <http://arxiv.org/abs/2006.14248>, <http://www.di.uniba.it/loglisci/NFMCP2019/program.html>
10. Han, S.W.: Efficient change detection methods for bio and healthcare surveillance. Ph.D. thesis, Georgia Institute of Technology (2010)
11. Hanlon, M., Jackson, B., Rice, J., Walsh, J., Riordan, D.: Audio pre-processing and neural network models for identification of orthopedic reamers in use. In: *2020 31st Irish Signals and Systems Conference, ISSC 2020*. Institute of Electrical and Electronics Engineers Inc. (2020). <https://doi.org/10.1109/ISSC49989.2020.9180175>
12. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification (2017). <http://arxiv.org/abs/1703.07737>
13. Li, L., Sung, M., Dubrovina, A., Yi, L., Guibas, L.J.: Supervised fitting of geometric primitives to 3D point clouds. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 2647–2655 (2019). <https://doi.org/10.1109/CVPR.2019.00276>
14. Liang, B., Wu, P., Tong, X., Qiu, Y.: Regression and subgroup detection for heterogeneous samples. *Comput. Stat.* **35**(4), 1853–1878 (2020). <https://doi.org/10.1007/s00180-020-00965-5>
15. Liu, Y., Jun, E., Li, Q., Heer, J.: Latent space cartography: visual analysis of vector space embeddings. Technical report 3 (2019). <https://doi.org/10.1111/cgf.13672>
16. Mahony, N.O., Campbell, S., Carvalho, A., Krpalkova, L., Riordan, D., Walsh, J.: Improving accuracy and latency in image re-identification by gallery database cleansing. In: Arai, K. (ed.) *Intelligent Computing. LNNS*, vol. 283, pp. 911–921. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-80119-9_60
17. Finzi, M., Wang, K.A., Wilson, A.G.: Simplifying hamiltonian and lagrangian neural networks via explicit constraints. In: *Neural Information Processing Systems Conference, NeurIPS 2020* (2020). <https://proceedings.neurips.cc/paper/2020/file/9f655cc8884fda7ad6d8a6fb15cc001e-Paper.pdf>
18. Minervini, P., Demeester, T., Rocktäschel, T., Riedel, S.: Adversarial sets for regularising neural link predictors. In: *Uncertainty in Artificial Intelligence - Proceedings of the 33rd Conference, UAI 2017* (2017). <http://arxiv.org/abs/1707.07596>
19. De Oca, V.M., Jeske, D.R., Zhang, Q., Rendon, C., Marvasti, M.: A cusum change-point detection algorithm for non-stationary sequences with application to data network surveillance. *J. Syst. Softw.* **83**(7), 1288–1297 (2010). <https://doi.org/10.1016/j.jss.2010.02.006>
20. O’ Mahony, N., et al.: Regressing relative fine-grained change for sub-groups in unreliable heterogeneous data through deep multi-task metric learning. *Sens. Transducers J.* **252**(5), 50–57 (2021). https://www.sensorsportal.com/HTML/DIGEST/P_3234.htm
21. O’Mahony, N., et al.: Understanding and exploiting dependent variables with deep metric learning. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) *IntelliSys 2020. AISC*, vol. 1250, pp. 97–113. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-55180-3_8

22. Panuju, D.R., Paull, D.J., Griffin, A.L.: Change detection techniques based on multispectral images for investigating land cover dynamics. *Remote Sens.* **12**(11), 1781 (2020). <https://doi.org/10.3390/rs12111781>, <https://www.mdpi.com/2072-4292/12/11/1781>
23. PyTorch: PyTorch (2019). <http://pytorch.org/>
24. Recanatesi, S., Farrell, M., Lajoie, G., Deneve, S., Rigotti, M., Shea-Brown, E.: Predictive learning as a network mechanism for extracting low-dimensional latent space representations (2018). <https://doi.org/10.1101/471987>
25. Rocktäschel, T., Singh, S., Riedel, S.: Injecting logical background knowledge into embeddings for relation extraction. In: NAACL HLT 2015–2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, pp. 1119–1129. Association for Computational Linguistics, Stroudsburg (2015). <https://doi.org/10.3115/v1/n15-1118>, <http://aclweb.org/anthology/N15-1118>
26. Senanayake, R., Ott, L., O'Callaghan, S., Ramos, F.: Spatio-temporal Hilbert maps for continuous occupancy representation in dynamic environments. In: Advances in Neural Information Processing Systems (NIPS), pp. 3925–3933 (2016)
27. Seo, S., Liu, Y.: Differentiable physics-informed graph networks. In: 2017 Advances in neural information processing systems, pp. 4967–4976 (2019). <http://arxiv.org/abs/1902.02950>
28. Shi, W., Zhang, M., Zhang, R., Chen, S., Zhan, Z.: Change detection based on artificial intelligence: state-of-the-art and challenges. *Remote Sens.* **12**(10), 1688 (2020). <https://doi.org/10.3390/rs12101688>
29. Smith, A.L., Asta, D.M., Calder, C.A.: The geometry of continuous latent space models for network data. *Stat. Sci.* **34**(3), 428–453 (2019). <https://doi.org/10.1214/19-STS702>
30. Wang, L., Zhu, D.: Tackling ordinal regression problem for heterogeneous data: sparse and deep multi-task learning approaches. *Data Min. Knowl. Disc.* **35**(3), 1134–1161 (2021). <https://doi.org/10.1007/s10618-021-00746-8>
31. Wattenberg, M., Viégas, F., Johnson, I.: How to use t-SNE effectively. *Distill* **1**(10), e2 (2017). <https://doi.org/10.23915/distill.00002>, <http://distill.pub/2016/misread-tsne>
32. Zhou, T., Thung, K.H., Liu, M., Shi, F., Zhang, C., Shen, D.: Multi-modal latent space inducing ensemble SVM classifier for early dementia diagnosis with neuroimaging data. *Med. Image Anal.* **60**, 101630 (2020). <https://doi.org/10.1016/J.MEDIA.2019.101630>
33. Zhu, L., Zhang, J., Sun, Y.: Remote sensing image change detection using super-pixel cosegmentation. *Inf. (Switz.)* **12**(2), 1–23 (2021). <https://doi.org/10.3390/info12020094>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Personalised Filter Bias with Google and DuckDuckGo: An Exploratory Study

Awais Akbar¹(✉), Simon Caton², and Ralf Bierig¹

¹ Department of Computer Science, Maynooth University, Kildare, Ireland
awais.akbar.pk@gmail.com

² School of Computer Science, University College Dublin, Dublin, Ireland

Abstract. Personalisation in search has improved performance, focus, and user experience to a great extent, however, it also arguably polarises informational perspectives. This paper seeks to illustrate an experimental methodology to quantify how three situational user variables affect personalisation across two search engines: Google and DuckDuckGo. We find that the presence of cookies and prior search history markedly affect the first page of search results on both platforms, but that prior (shallow) browsing history has no observable effect. We also find that there is very little in common between the results of both search engines. We argue that these results advocate more consideration of how personalisation fosters filter biases.

Keywords: Personalisation · Filter bias · Simulation experiment · Search engine

1 Introduction

Search engines (and the web in general) are highly aligned to users' perceived interests. While this often delivers "relevant" content, it is arguably informational polarisation and can negate the serendipity of search through the development of "Filter Bubbles". Whilst researchers have begun to investigate search engine performance in relation to user information needs (e.g. [8, 33, 35]), we argue that methodologies that more directly quantify the contribution of situational user variables to personalised results are needed to better understand potential biases and their implications.

To provide some initial empirical insights, we investigate the extent to which search result personalisation is informed (or influenced) by: 1) user's information stored in *browser cookies*; 2) user's *prior search history*; and 3) user's *prior browsing history*. To derive empirical results, we investigate two common search engines: Google and DuckDuckGo, with three experiments designed to expose any discernible differences in search engine behaviour by analysing the content of Search Engine Results Page(s) (SERP). To investigate these aspects, we leverage a simulation-based controlled experiment, i.e. we instrument an automated search process within an engineered user context. Our methodology controls for noise, specifically the carry-over effect [15], to accurately attribute the differences

to personalization in the returned results. The carry-over effect is a phenomenon that occurs within a browsing session when a user immediately searches for one query after another. In this case, the search for the first query may influence the results received by the immediate search for the second query. This strategy has been documented on Google by Hannak et al. [15].

Our motivation for a simulation-based study is that it provides significant control over key variables (cookie information, prior browsing/search history, the ordering and nature of search terms as well as situational context of search, i.e. browser headers etc.) that can provide initial insights for the design of a more expansive user-based study. Thus, the contribution of this paper is a set of empirical results that investigate the impact of browser cookies in general (without being logged into the search engine's ecosystem), the impact of prior user searches, and the impact of users' prior browsing behaviour on SERPs as a juxtaposition over two major search engines: Google and DuckDuckGo using a variety of search terms from multiple categories. In general, we find that both search engines are influenced, albeit differently, by our construed situational search context in a manner that is indicative of personalisation biases.

2 Related Work

While personalisation for the web and for search services has been explored and practised for the past two decades [7, 26], its downsides have been equally established [28]. Personalised search engines help people to focus and increase their effectiveness, but they also potentially overexpose their users with information experiences that are highly aligned to their long-standing digital profiles. Pariser [28] coined the term “Filter Bubble” to describe this effect, defining it as “the personal ecosystem of information that’s been created by the personalisation algorithms”. He argued that Google’s personalisation algorithms provide users with information that reinforces their ideas and hides the information that opposes their viewpoints, thus, decreasing the diversity of their views. Due to this interference, users might not see the contrasting viewpoints on a moral or political issue [6]. As a result, they will be trapped in a filter bubble without even knowing what they are missing [28]. This may lead to fewer serendipitous information encounters in the short term, and narrower views, informational blind spots, or radical polarisation in the longer term [4, 28]. Awareness of bias in news and media has gained substantial attention on its own [5, 19, 34] as well as in relation to personalised search and news services [10, 12, 15, 23, 32].

While relevance and link structure of online resources are decisive factors in determining the placement of search results, studies indicate that several other factors such as politics [21], economics and social biases [2], etc. play a role in ranking and may lead to biased results [29]. Bias based on geographical location also occurs because popular search engines are in the USA. A study by Vaughan and Thelwall [40] testing three main search engines for national bias discovered that websites based in the USA were much better covered. A new study by Cooper et al. [9] identified significant variations when extracting scientific articles for composing review papers. The study compared results from

the same queries across 12 countries. Some of its geographical locations (based on the IP address) suppressed more than half of its relevant results. Bias can also be caused by search engines showing popular search results first [17] and learning from user click behaviors. Google's auto-complete feature has also been shown to be biased towards more popular searches and sometimes offers some questionable choices¹. White [42] investigated inherent search engine biases and their effect on information quality. He showed that half of the time, the combined effect of inherent biases and user preferences leads people to incorrect beliefs. Epstein and Robertson [12] investigated the impact of search results on the election outcome and showed that voting preferences of undecided users can change by at least 20% due to biases in search results. At present, search engines retrieve and present biased information to users. Google, for instance, provides personalised search results based on ~57 different signals including user's search history, location, past click behavior, etc. [28]. Thereby, creating a filter bubble by limiting the search results that we get for a particular topic.

Our work differs in a number of ways from previous research. While the prior work [12, 20, 21, 24, 31, 32, 39, 41] aimed to quantify personalisation bias in web search, the studies were rather limited to the political searches only. Furthermore, the authors did not control the noise in search results i.e. the carry-over effect [15]. Besides, some other studies [9, 18, 27] also focused on search bias quantification, nevertheless, only single-user features such as geolocation was considered. For instance, Cooper et al. [9] used a virtual private network (VPN) to conduct the same Google searches in 12 different countries to study the impact of users' geographical location on returned results. The authors find that the user's location appears to be influencing the results returned in response to the searches conducted for systematic reviews. Likewise, Silver et al. [18] conducted a series of searches over a period of 30 days using 240 queries. The results collected from the 59 GPS coordinates in the US revealed that location-based personalisation leads to ~40–50% change in search results for localised queries and a minimal change in results for more general queries. Similarly, the impact of location on search results personalisation was studied in [27], however, only image search results were considered and the queries were also kept limited to Covid-19. The authors conducted the same search experiments in four different parts of Europe and compared results across the countries. Surprisingly, they only found a ~46% overlap in search results, which became minimal when the queries were expressed in different languages. Other researchers [14, 23] quantified search bias in Google News, thereby, limiting the scope of their research to news outlets.

Our research, compared to the previous works, has a wider scope. It spans a wide category of search terms, includes various user features, and also controls for noise. Comparing it with the work of Hannak et al. [15], we maintain the IP address as a control variable. Hannak et al. also focused on training the browser

¹ The Wired article from 2018 reports on this issue <https://www.wired.com/story/google-autocomplete-vile-suggestions>. Note that Google's auto-complete suggestions can now be reported. Nevertheless, the issue remains relevant.

profiles to represent various demographic properties (e.g. age, gender, ethnicity), however, we focused on training our browser profiles to absorb the history of search and browsing behaviour. Furthermore, we included DuckDuckGo as a second, more neutral counterpart. We chose DuckDuckGo mainly because of two reasons: first, it claims to respect users’ privacy and does not track them during web search sessions, and second, it is the most widely used privacy-protecting search engine. About 35 billion queries were searched on DuckDuckGo during the year 2021, with a monthly average of 3 billion searches and a daily average of up to 101 million searches². In addition to the choice of search engines, we also used different search terms to measure personalisation. It is reported in the literature that the magnitude of personalisation varies with search terms [15, 21]. Lastly, the personalisation algorithms of Google have a changing nature. It is known that Google continuously updates its data sources and personalisation algorithms over time. For instance, there have been changes in Google’s privacy policy in recent years [13, 30], which allowed Google to aggregate users’ data throughout its services (e.g., Gmail, Search, DoubleClick, Google Analytics, etc.) for content personalisation and targeted advertising. Therefore, using a third-party tracking and analytics network, Google now infers users’ browsing history and personalises search results in a more effective way. We believe these changes in privacy policy have also produced a need for a more recent study on the subject as a significant amount of time has already passed since the prior research was conducted.

3 Methodology

Table 1. Search terms from four different categories

Category	Queries
News (6)	Corona virus; stimulus check; boris johnson; public health england; Weather tomorrow; democratic debate
Health (6)	Pandemic; covid-19 antibody testing; face masks; quarantine; Hand sanitizer; immune system
Sports (3)	FIFA 2022, ICC Men’s T20 World Cup; kobe bryant;
Science (5)	Brain out; weathering with you; locust, asteroid; What dinosaur has 500 teeth

We investigate three situational aspects of search: cookies, past search, and past browsing history; corresponding to three separate experiments. All experiments use both the Google and the DuckDuckGo search services with the query collection shown in Table 1. Default settings were used in both search engines. All experiments were conducted with an in-house tool built on PhantomJS³,

² <https://duckduckgo.com/traffic>.

³ <https://phantomjs.org>.

a headless-browser framework that allows simulating real user interaction with search engines collecting SERP in real-time. We avoided search engine APIs as they have been suspected of presenting results differently [25]. All the experiments run during the summer of 2020 in Dublin, Ireland.

The *cookie-tracking experiment* captures any personalisation bias that is driven by user information collected, stored, and maintained in browser cookies. Search engine providers can use cookies to create a user model even though the user is currently not logged into their ecosystem [15]. To evaluate the impact of cookies, we conducted a series of web searches during which all cookies throughout the search session were either enabled or disabled.

The *search-history experiment* investigates the personalisation of search results over time. We conducted a series of web searches once per day, for four consecutive days, once with cookies enabled and once with cookies disabled. Experiments run every day from 12 noon GMT for approximately 5 h.

The *browsing-history experiment* reviews the interactive effects of search personalisation. We examined whether Google and DuckDuckGo personalise search results based on users' browsing history outside the usual search activity. First, all news-related queries (see Table 1) were searched. The script then browsed four news domains from four countries (*dw.com* (Germany), *news.com.au* (Australia), *cbc.ca* (Canada), and *scmp.com* (China)) and followed two random links on each portal to simulate a brief episode of shallow browsing. The script then ran all news-related queries again and compared SERPs with the earlier search results.

All experiments used a set of 20 queries covering four topics (news, health, sports, and science, as shown in Table 1). Similar to other researchers (e.g. [15, 16]), we selected queries from Google Trends⁴, and WebMD⁵ for the health-related topics. Google Trends was chosen as a platform for query collection as it shows the queries that remained popular over a particular period of time and also sorts the queries based on different categories, geographical locations, etc. We chose the queries that were trending in the last year but did not limit query selection to any particular region/city, that is, the selected region was "Worldwide". Between each subsequent search in all experiments, our script waited for 15 min to prevent any "carry-over effects" [15].

4 Results

We measured search result personalisation as the difference in URLs (links to the target page) between two SERPs. We only consider the first SERP from both search engines, based on prior findings [3] which showed that users often limit interactions to the first SERP. If 1 of the 10 search results (web links) differed across the two SERPs, we define the personalised difference to be 10% regardless of any ordering differences.

⁴ <https://trends.google.com/trends/>.

⁵ <https://www.webmd.com/>.

As part of the *cookie-tracking experiment* we executed our set of queries in a single session on both search engines once with cookies enabled, and once with cookies being cleared between individual queries. We found that cookie-based personalisation with Google is relatively high ($\sim 37\%$) in comparison to DuckDuckGo ($\sim 20\%$). This implies that Google changed on average 3 or 4 results, while DuckDuckGo adapted about 2 results between these two conditions. Results from the *search-history experiment* are more differentiated and therefore depicted in Fig. 1. Here, our query collection was repeatedly submitted over four consecutive days. During this time, personalisation ranges from 28% to 41% (3–4 adapted results) with Google and 9% to 28% (1–3 adapted results) with DuckDuckGo. Specifically, search-history-based personalisation with cookies ranges from 32–35% for Google and 12–28% for DuckDuckGo. Without cookies, personalisation varied from 28–41% (Google) and 9–27% (DuckDuckGo). The upper two lines in Fig. 1 show the differences in personalised results for Google, whereas the lower two lines show the variations in personalisation for DuckDuckGo. Note that the first day is used as a reference, and is therefore 0 for all cases.

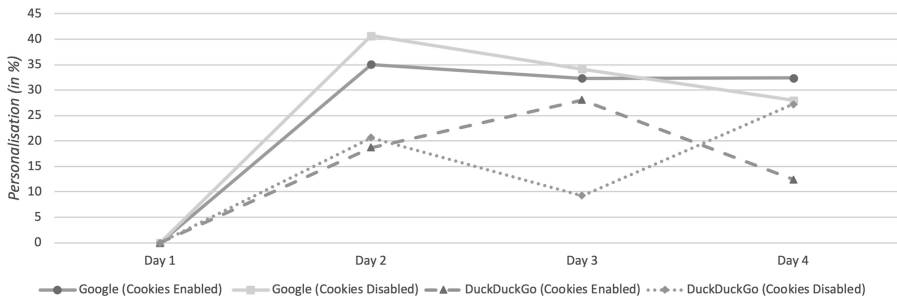


Fig. 1. Search-History Based Personalisation in Google and DuckDuckGo (in %)

The *browsing-history experiment* reviewed the impact of simulated shallow browsing on search result personalisation. Usually, SERPs provide a section that shows the latest news in relation to a submitted query. During this experiment, our script executed only news-related queries both before and after browsing the links on four different news domains. Neither Google’s “Top Stories” nor DuckDuckGo’s “Recent News” revealed personalised adaptations in response to our simulated browsing behaviour. However, Google returned localised results while DuckDuckGo remained neutral. This suggests that DuckDuckGo does not use IP addresses as a personalisation signal which supports DuckDuckGo’s claim of not tracking its users. Nevertheless, our results show some evidence that DuckDuckGo may use other signals to personalise search results: e.g. search history.

Additionally, we found that very few search results were commonly shared between search services – on average between 2–8%, as shown in Fig. 2. Specifically, Google and DuckDuckGo share only about 2% and 4% of their results in the news and health categories. While for the other two categories (science and sports), this percentage is slightly higher ($\sim 6\%$ with cookies disabled, and $\sim 8\%$

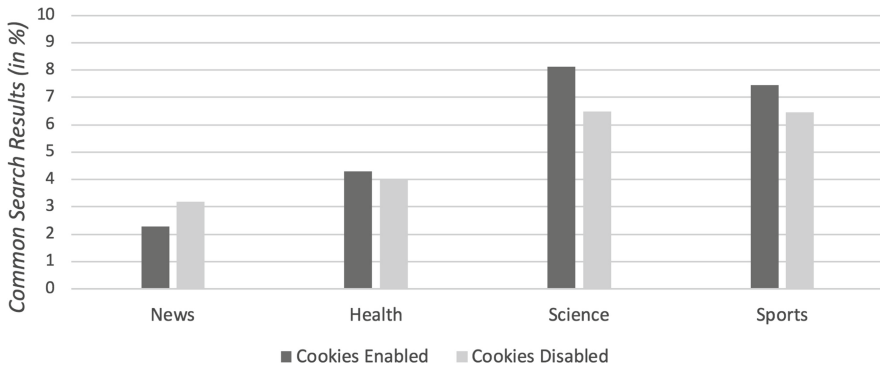


Fig. 2. Overlap between Google and DuckDuckGo results, based on queries in the four topical categories (in %)

with cookies enabled). To the best of our knowledge, this is a finding that has not been investigated previously and it is generally surprising that there is that little in common between Google and DuckDuckGo, even on rather objective queries covering categories such as science or health-related topics. Prior studies, however, focused on measuring the overlap between other search engines – e.g. Google, MSN, Yahoo, and Ask Jeeves [38], and Google and Bing [1], with [38] finding only a minimal overlap of about 1%. In a later study, Spink et al. [37] found less commonality in the first page results of four search engines compared to their previous study. Similarly, Ding and Marchionini [11] studied the distinctiveness in the search results (of InfoSeek, Lycos, and OpenText), and Selberg and Etzioni [36], who conducted a study to measure the overlap in search results (of Galaxy, Infoseek, Lycos, OpenText, Webcrawler and Yahoo) both found that the search engines returned the results that were unique to each other.

We have presented a methodology to evaluate personalisation bias in common search engines for three different situational variables: 1) browser cookies, 2) users' search history, and 3) users' browsing history. Our results show that Google adapts on average about 40% of its first results page, whereas DuckDuckGo adapts about 20%. Even though DuckDuckGo claims that it does not track its users, we found the service appears to perform certain forms of personalisation in response to different situational variables, and that this spans multiple query categories. While our results indicate that users' search history influences SERP variation for both Google and DuckDuckGo, Google search results depicted increased levels of personalisation. This indicates that further research is needed to quantify the effects of search history on search personalisation. Shallow browsing appears to not significantly affect personalised results for both search services. However, as we have only simulated a simple browsing episode, further research is needed to conclusively exclude this parameter as a potential source for personalisation bias.

5 Conclusion and Future Work

In this paper, we explored the potential for personalisation biases in search under different experimental user context settings: information stored in cookies, search history, and browsing history. Our results have shown that personalisation biases exist in both Google and DuckDuckGo, even if a user is not actively logged into the search engine ecosystem. As a result, users are consistently provided with adapted answers for their queries which may alter judgment and decision making [12, 22, 42]. While personalisation can be a useful measure to help people overcome handling an overabundance of information, we need to be aware of the cost of personalisation. This is less about users settling for “incorrect” answers, but rather the potential for over-exposure to one-sided viewpoints that reinforce beliefs on a potentially critical subject matter – a filter bubble that conveniently allows people to avoid learning alternative and competing views inhibit healthy information society.

Furthermore, all the previous studies in the literature find a small overlap in the first search result page of different search engines for a variety of search terms. There could be many reasons for this little overlap. First, there are constraints on the search engines in the portion of the web they index, owing to disk storage, computational power, and network bandwidth. Different technologies are used by search engines for finding the pages and indexing them. Furthermore, proprietary algorithms are deployed by search engines for determining the results’ ranking and their demonstration to users. Hannak et al. [15] consider implicit personalization as a plausible reason. From our study, we form the opinion that the use of different search engines could be beneficial for users. It increases information viewpoint diversity since each search engine share a different perspective on a topic, therefore, the filter bubble effect can be mitigated using different search engines.

This work has derived its empirical findings via a simulation-based approach, a natural extension would be to use these findings to inform the design of a larger-scale user study to both corroborate and extend the findings. Similarly, there would be several additional user context variables that could be further explored. Key examples here are location, and web browser (as well as specific settings). An additional extension of this research involves exploring additional search engines, and also conducting similar experiments on popular news outlets such as the New York Times and the Washington Post for detecting bias in the provision of news stories.

References

1. Agrawal, R., Golshan, B., Papalexakis, E.: Overlap in the web search results of google and Bing. *J. Web Sci.* **2**, 17–30 (2016). <https://doi.org/10.1561/106.000000005>
2. Baeza-Yates, R.: Bias on the web. *Commun. ACM* **61**(6), 54–61 (2018). <https://doi.org/10.1145/3209581>, <https://dl.acm.org/doi/10.1145/3209581>
3. Bar-Ilan, J., Keenoy, K., Yaari, E., Levene, M.: User rankings of search engine results. *J. Am. Soc. Inf. Sci. Technol.* **58**(9), 1254–1266 (2007). <https://doi.org/10.1002/asi.20608>, <http://doi.wiley.com/10.1002/asi.20608>

4. Bierig, R., Caton, S.: Special issue on de-personalisation, diversification, filter bubbles and search. *Inf. Retr. J.* **22**(5), 419–421 (2019). <https://doi.org/10.1007/s10791-019-09365-w>, <http://link.springer.com/10.1007/s10791-019-09365-w>
5. Bourgeois, D., Rappaz, J., Aberer, K.: Selection bias in news coverage: learning it, fighting it. In: *The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018*, pp. 535–543. Association for Computing Machinery, Inc., April 2018. <https://doi.org/10.1145/3184558.3188724>
6. Bozdag, E.: Bias in algorithmic filtering and personalization. *Ethics Inf. Technol.* **15**(3), 209–227 (2013). <https://doi.org/10.1007/s10676-013-9321-6>, <http://link.springer.com/10.1007/s10676-013-9321-6>
7. Brusilovsky, P., Maybury, M.T.: Special issue: from adaptive hypermedia to the adaptive web. *Commun. ACM* **45**(5), 30–33 (2002). <https://doi.org/10.1145/506218.506239>
8. Clarke, C.L., et al.: Novelty and diversity in information retrieval evaluation. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR 2008*, p. 659. ACM Press, New York, New York, USA (2008). <https://doi.org/10.1145/1390334.1390446>, <http://portal.acm.org/citation.cfm?doid=1390334.1390446>
9. Cooper, C., Lorenc, T., Schauburger, U.: What you see depends on where you sit: the effect of geographical location on web-searching for systematic reviews: a case study. *Res. Synth. Methods* (2021). <https://doi.org/10.1002/jrsm.1485>
10. Dillahunt, T.R., Brooks, C.A., Gulati, S.: Detecting and visualizing filter bubbles in google and Bing. In: *Conference on Human Factors in Computing Systems - Proceedings*, vol. 18, pp. 1851–1856. Association for Computing Machinery, New York, New York, USA, April 2015. <https://doi.org/10.1145/2702613.2732850>, <http://dl.acm.org/citation.cfm?doid=2702613.2732850>
11. Ding, W., Marchionini, G.: A comparative study of web search service performance. In: *ASIS Annual Meeting*, pp. 136–42 (1996)
12. Epstein, R., Robertson, R.E.: The search engine manipulation effect (SEME) and its possible impact on the outcomes of elections. *Proc. Natl. Acad. Sci. U. S. A.* **112**(33), E4512–E4521 (2015). <https://doi.org/10.1073/pnas.1419828112>, <https://www.pnas.org/content/112/33/E4512>
13. Google: Google’s Privacy Policies. <https://policies.google.com/privacy/archive?hl=en-US>
14. Haim, M., Graefe, A., Brosius, H.B.: Burst of the filter bubble?: Effects of personalization on the diversity of google news. *Digit. Journal.* **6**(3), 330–343 (2018). <https://doi.org/10.1080/21670811.2017.1338145>
15. Hannak, A., et al.: Measuring personalization of web search. In: *WWW 2013–Proceedings of the 22nd International Conference on World Wide Web*, pp. 527–537. ACM Press, New York, New York, USA (2013). <https://doi.org/10.1145/2488388.2488435>, <http://dl.acm.org/citation.cfm?doid=2488388.2488435>
16. Hoang, V.T., Spognardi, A., Tiezzi, F., Petrocchi, M., De Nicola, R.: Domain-specific queries and web search personalization: some investigations. In: *Electronic Proceedings in Theoretical Computer Science, EPTCS*, vol. 188, pp. 51–58. Open Publishing Association, August 2015. <https://doi.org/10.4204/EPTCS.188.6>
17. Introna, L.D., Nissenbaum, H.: Shaping the web: why the politics of search engines matters. *Inf. Soc.* **16**(3), 169–185 (2000). <https://doi.org/10.1080/01972240050133634>

18. Kliman-Silver, C., Hannak, A., Lazer, D., Wilson, C., Mislove, A.: Location, location, location: the impact of geolocation on web search personalization. In: Proceedings of the 2015 Internet Measurement Conference, pp. 121–127. IMC 2015, Association for Computing Machinery, New York, NY, USA (2015). <https://doi.org/10.1145/2815675.2815714>
19. Knoche, M., Popović, R., Lemmerich, F., Strohmaier, M., Stroh-maier, M.: Identifying biases in politically biased wikis through word embeddings. In: Proceedings of the 30th ACM Conference on Hypertext and Social Media. ACM, New York, NY, USA (2019). <https://doi.org/10.1145/3342220.3343658>
20. Krafft, T.D., Gamer, M., Anna, K.: What did you see? A study to measure personalization in Google's search. *EPJ Data Sci.* **8**, 1–23 (2019). <https://doi.org/10.1140/epjds/s13688-019-0217-5>, <https://epjdatascience.springeropen.com/articles/10.1140/epjds/s13688-019-0217-5>
21. Kulshrestha, J., et al.: Search bias quantification: investigating political bias in social media and web search. *Inf. Retr. J.* **22**(1–2), 188–227 (2019). <https://doi.org/10.1007/s10791-018-9341-2>, <https://link.springer.com/article/10.1007/s10791-018-9341-2>
22. Lai, C., Luczak-Roesch, M.: You can't see what you can't see: experimental evidence for how much relevant information may be missed due to Google's web search personalisation. In: Weber, I., et al. (eds.) *SocInfo 2019. LNCS*, vol. 11864, pp. 253–266. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34971-4_17
23. Le, H., Maragh, R., Ekdale, B., High, A., Havens, T., Shafiq, Z.: Measuring political personalization of google news search. In: *The World Wide Web Conference on WWW 2019*, pp. 2957–2963. Association for Computing Machinery (ACM), New York, New York, USA (2019). <https://doi.org/10.1145/3308558.3313682>, <http://dl.acm.org/citation.cfm?doid=3308558.3313682>
24. Martinovic, M.: Exploring the effect of search engine personalization on politically biased search results (2018)
25. McCown, F., Nelson, M.L.: Agreeing to disagree: Search engines and their public interfaces. In: *Proceedings of the ACM International Conference on Digital Libraries*, pp. 309–318. ACM Press, New York, New York, USA (2007). <https://doi.org/10.1145/1255175.1255237>, <http://portal.acm.org/citation.cfm?doid=1255175.1255237>
26. Micarelli, A., Gasparetti, F., Sciarrone, F., Gauch, S.: Personalized search on the world wide web. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *The Adaptive Web. LNCS*, vol. 4321, pp. 195–230. Springer, Berlin, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_6
27. Paramita, M.L., Orphanou, K., Christoforou, E., Otterbacher, J., Hopfgartner, F.: Do you see what i see? Images of the COVID-19 pandemic through the lens of google. *Inf. Process. Manag.* **58**(5), 102654 (2021). <https://doi.org/10.1016/j.ipm.2021.102654>, <https://www.sciencedirect.com/science/article/pii/S0306457321001424>
28. Pariser, E.: *The Filter Bubble: What the Internet Is Hiding from You*. The Penguin Group, London (2011)
29. Pitoura, E., et al.: On measuring bias in online information. *SIGMOD Rec.* **46**(4), 16–21 (2018). <https://doi.org/10.1145/3186549.3186553>
30. ProPublica.: Google Has Quietly Dropped Ban on Personally Identifiable Web Tracking (2016). <http://bit.ly/2eAjC9w>
31. Puschmann, C.: Beyond the bubble: assessing the diversity of political search results. *Digit. Journal.* **7**(6), 824–843 (2019). <https://doi.org/10.1080/21670811.2018.1539626>

32. Robertson, R.E., Lazer, D., Wilson, C.: Auditing the personalization and composition of politically-related search engine results pages. In: The Web Conference 2018–Proceedings of the World Wide Web Conference, WWW 2018, pp. 955–965. Association for Computing Machinery Inc, New York, New York, USA, April 2018. <https://doi.org/10.1145/3178876.3186143>, <http://dl.acm.org/citation.cfm?doid=3178876.3186143>
33. Sakai, T., Kando, N., Macdonald, C., Soboroff, I.: Introduction to the special issue on search intents and diversification. *Inf. Retr.* **16**(4), 427–428 (2013). <https://doi.org/10.1007/s10791-013-9223-6>, <http://link.springer.com/10.1007/s10791-013-9223-6>
34. Sales, A., Balby, L., Veloso, A.: Media bias characterization in Brazilian presidential elections. In: HT 2019 - Proceedings of the 30th ACM Conference on Hypertext and Social Media, pp. 231–240. Association for Computing Machinery, Inc., September 2019. <https://doi.org/10.1145/3342220.3343656>
35. Santos, R.L.T., Macdonald, C., Ounis, I.: Search result diversification. *Found. Trends® Inf. Retr.* **9**(1), 1–90 (2015). <https://doi.org/10.1561/1500000040>
36. Selberg, E., Etzioni, O.: Multi-service search and comparison using the MetaCrawler. In: 4th International Conference on World Wide Web (1995)
37. Spink, A., Jansen, B.J., Wang, C.: Comparison of major web search engine overlap: 2005 and 2007. In: 14th Australasian World Wide Web Conference
38. Spink, A., Jansen, B.J., Blakely, C., Koshman, S.: A study of results overlap and uniqueness among major web search engines. *Inf. Process. Manag.* **42**(5), 1379–1391 (2006). <https://doi.org/10.1016/j.ipm.2005.11.001>, <https://linkinghub.elsevier.com/retrieve/pii/S0306457305001500>
39. Urman, A., Makhortykh, M., Ulloa, R.: The matter of chance: auditing web search results related to the 2020 U.S. presidential primary elections across six search engines. *Soc. Sci. Comput. Rev.* 08944393211006863. <https://doi.org/10.1177/08944393211006863>, <https://doi.org/10.1177/08944393211006863>
40. Vaughan, L., Thelwall, M.: Search engine coverage bias: evidence and possible causes. *Inf. Process. Manag.* **40**(4), 693–707 (2004). [https://doi.org/10.1016/S0306-4573\(03\)00063-3](https://doi.org/10.1016/S0306-4573(03)00063-3), <https://linkinghub.elsevier.com/retrieve/pii/S0306457303000633>
41. Weber, I., Garimella, V.R.K., Borra, E.: Mining web query logs to analyze political issues. In: Proceedings of the 4th Annual ACM Web Science Conference, pp. 330–334. WebSci 2012, Association for Computing Machinery, New York, NY, USA (2012). <https://doi.org/10.1145/2380718.2380761>
42. White, R.W.: Beliefs and biases in web search. In: SIGIR 2013–Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 3–12. ACM Press, New York, New York, USA (2013). <https://doi.org/10.1145/2484028.2484053>, <http://dl.acm.org/citation.cfm?doid=2484028.2484053>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Entity Resolution for Multiple Sources with Extended Approach

Phuc Pham Huu¹(✉) , Dongyun Nie² , and Michael Scriney^{1,2}

¹ Insight Centre for Data Analytics, Dublin City University, Dublin, Ireland
phucphamhuu@insight-centre.org

² School of Computing, Dublin City University, Dublin, Ireland
{dongyun.nie,michael.scriney}@dcu.ie

Abstract. Entity Resolution is a technique to find similar records that may refer to the same entity from one or many resources. It is mainly used in data integration or data cleaning with the existence of Big Data. It not only helps organisations have clean data, but it also provides a unified view of their data for later analysis. However, there is no one solution fitting all duplication issues. Because of the fact that the data itself is heterogeneous and varied. This paper focuses on finding the answers to the usefulness of a combination of different matching approaches, token blocking versus standard blocking and how other domain runs by examining how well they perform in different scenarios. To achieve these answers, this paper outline details and setups for these experiments to execute. A detailed evaluation demonstrates the effectiveness of the approaches with multiple datasets.

Keywords: Entity resolution · Blocking · Clustering

1 Introduction

Modern enterprises require large datasets from multiple sources which are represented by heterogeneous schemas, i.e., each source contains a different feature-set. The data quality can be incomplete, redundant, inconsistent or incorrect when compared among these schemas. Additionally, within such datasets single entities may appear with multiple different representations. As such a large investment in data processing is required. If the data is not processed to a sufficient standard, standard enterprise operations are affected and, crucially, decision-making abilities are hampered. To overcome this and reduce complexity while processing large chunks of data, Entity Resolution (ER) is an approach to recognize records referring to the same entities, objects or persons, etc. within datasets where entity identifiers are unclear. It can be known as record linkage, data matching or record deduplication. This is different from entity linking in

The work reported in this paper was part-funded by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2 (Insight SFI Research Centre for Data Analytics).

© The Author(s) 2023

L. Longo and R. O'Reilly (Eds.): AICS 2022, CCIS 1662, pp. 514–526, 2023.

https://doi.org/10.1007/978-3-031-26438-2_40

Natural Language Processing which mainly groups relevant information about entities for analysis and news. The main goals of ER are to match records stored in the same, known as deduplication, or different tables, which is record linkage, for better data quality.

General processing tasks of ER problems involve four phases: blocking, block processing, entity matching and entity clustering [10]. Blocking is the first task which discards as many comparisons as possible by putting similar records into blocks. The blocking or Block Building phase generates block collections which are groups formed by similar descriptions [4]. It mostly focuses on recall. Blocking Processing which improves the reduction ratio of the blocks. The reduction ratio estimates the comparison space which is calculated by the total of matches and non-matches, and the total of true matches and true non-matches as defined in [3]. This step is done by reducing most superfluous and redundant comparisons in the Blocking phase [4]. The most important step in ER workflow is Matching [4]. This phase aims at deciding if each pair of these records coexisting in the same block collection refer to the same real-world entity by using some similarity function. The final task in the ER workflow is Clustering [4]. Its input is the similarity graph showing sets of entity pairs associated with the edge weights, the matching likelihood, resulting from Matching. The ending goal of this step is to a set of entity clusters in which each corresponds to one real-world entity [4]. Recent studies showed the benchmark of some entity clustering algorithms for several particular data linkages [8, 12, 14] with good results. However, there remains room for improvement during the blocking and matching phases. Hence, with the goal of getting a higher recall score, this work focuses on implementing other techniques. In addition, we implemented more matching algorithms as the survey states that there can be many matching algorithm layers for better results [4] by adding in the matching phase.

As such, in this work, we extend the work presented in [12] by examining the effect of various combinations of ER algorithms across a standard ER pipeline on datasets from multiple domains. This paper is structured as followed. Section 2 details related research regarding Entity Resolution. Section 3 outlines our methodology. Section 4 reports our results and analysis. Finally, in Sect. 5 we detail our conclusions and future work.

2 Related Research

As stated by [8], most of the ER approaches are static which they consider entity groups in a set time frame from a number of static data sources. This research proposed a new and scalable approach to handle ER problems for upcoming data from different sources. The main approaches implemented the usage of Standard Blocking during the blocking phase and Qgram in the matching phase while having average scores resulting from the Qgram process for classification. The authors developed a process that continuously updates the entity blocks for new entries and sources without re-computing the whole entity clustering while having a competitive running speed. What is impressive about this research is

that their approaches perform quite stable across different domains. A survey of different blocking methods was presented in [3]. The authors examine blocking mechanisms on datasets across a variety of domains focused on record linkage and deduplication tasks. The author mentioned twelve variations of indexing techniques with algorithm explanation and evaluation with experiment results. These results demonstrate the performance of each approach on different datasets.

This research provides an overview of blocking keys and it suggested to have better work for more efficiency and scalability of the new indexing techniques. There exist several surveys listing possible algorithms for each phase of ER [4,9]. They highlighted the basic concepts, processing steps and strategies for different types of data sources. They also compared algorithms to see which one is used for different scenarios. In addition, [4] provided a list of existing tools for different usages. That overview also stated there exists no paper comparing the performance of the majority of matching algorithms. Among mentioned techniques in that survey, supervised learning and unsupervised learning trained the data by using the combination of different matching algorithms. While from the survey of [9], Standard Blocking and Sorted Neighbourhood are the most popular ones. The authors in [6] provide an approach to unsupervised learning that combines ensemble learning and enhanced automatic self-learning. This kind of approach proves beneficial when there are not many labelled data and various semi-supervised learning techniques. Ensemble learning aims at training and combining different classification models to get better performance than any individual classifiers. The system went through six steps. Blocking methods are referred to as Canopy Clustering and an unsupervised blocking scheme. In the second step, it selected different similarity measures. Then they were converted to similarity vectors so that the automatic seed selection process was used. Seed Q statistics were proposed to measure the diversity between sets of seeds. Once done, the self-learning algorithm was applied while the sixth step involved the proposed contribution ratio of base classifiers. They did eight experiments with different methods on four different datasets in this paper: Restaurant, Cora and two bibliographic datasets. Their experiment results indicated that this proposal could not outperform the supervised one. However, it seems to perform better than other unsupervised learning though it did not work well if there exist missing data and it requires a large number of record pair comparisons. The GenLink Algorithm proposed by [5] is another machine learning algorithm for the matching phase. The main idea of this approach is to develop a set of specialised crossover operators including function operator, operators crossover, aggregation crossover, transformation crossover, and threshold crossover. Each of these operators was in charge of a part of the linkage rule. The rule was represented by a combination of different distance measures non-linearly. There were six different datasets from three domains used in this research. Apart from Restaurant and Cora, they used Sider-Drugbank, NYT, LinkedMDB and DBpediaDrugbank datasets. They performed experiments on each of these and their results showed that it outperformed the state-of-the-art genetic programming approach for record linkage. Another important research that this paper could be related to is the comparison between the implementation of with or without machine learning techniques for real-world

datasets [7]. It is interesting that results are closely similar among approaches. However, there is still a big difference in the performance of distinct datasets.

Summary. The improvement of different algorithms is still an active area of research. However, there still exists some gap in comparing different matching algorithms. It is understood that different datasets require differing approaches. Hence, it is difficult to decide which aspects to compare matching approaches to. In addition, the need for parallelization is emphasized in most of the papers as this is one of the key elements to improve execution time.

3 Methodology

Entity resolution is a key component of a data integration pipeline, especially when the data pool is large. To produce quality data for analysis, the inputs are cleaned, transformed, and compared to other entities to find if they are similar. However, the actual process is more complicated as there are many different requirements, priorities, and approaches for different data. One of the recent researches has proposed an interesting approach that details promising performance for incremental entity resolution which assists the continuous addition of new records from new sources [8]. However, what seems to be an improvement for this research lies in the methods utilized. With the help of a standard blocking and trigram similarity function for the matching phase, the performance for the approaches was measured by precision, recall and F1 respectively. In this work, we evaluate the performance of multiple combinations of algorithms in addition to the Token blocking method presented in [4].

While there exist many datasets which can help us do this research, we had chosen to use the two datasets implemented by [8] and a real-world dataset. These datasets are from three different domains which are music, persons and products. The first dataset is the MusicBrainz dataset (DSM), a sample of this data can be seen in Table 1 generated by DAPO data generator¹ while the person dataset (DSP) in Table 2 was collected from North-Carolina voter registry generated by the tool GeCo². The third dataset contains the details of products (DSPro) on Amazon and Google³. These datasets come along with labeled values. The sample of these datasets can be seen in Table 1, 2 and 3.

Table 1. Music sample dataset

Title	Length	Artist	Album	Year
Just You, Just	296000	Sonny Stitt & Barry Harris	The Complete Late Quartets	2010
One Step Closer	193000	Oceans Divide	Oceans Divide EP	2011
All the Cocaine in the World (Maroon)	01:39	Webb Brothers	Maroon	2000

¹ <https://www.informatik.uni-leipzig.de/~saeedi/musicbrainz-2000-A01.csv.dapo>.

² <https://www.informatik.uni-leipzig.de/~saeedi/10Party-ocp20.tar.gz>.

³ <https://dbs.uni-leipzig.de/file/Amazon-GoogleProducts.zip>.

Table 2. Person sample dataset

Name	Surname	Suburb	Postcode
annie	johnson	warrenron	2758g
erib	pugh	asheboro	27202
lauren	toledo	chapeo hill	2751q
keorgetta	atkin5on	goldsboro	27530
lvnn	kitchenz	greensboro	27408

Table 3. Products sample dataset

Title	Description	Manufacturer	Price
monarch v9.0 1u	monarch v9.0 1u -	datawatch	699
midway arcade treasures (jewel case)	midway arcade treasures (win 98 me 2000 xp)	encore software	9.99
kids power fun for girls		topics entertainment	19.99
microspot interiors		microspot ltd	99.95

All experiments were completed on a laptop of 6 cores equipped with AMD Ryzen 5 3.00 GHz, 16 GB RAM and 1TB SSD Drive. All experiments are carried out several times to get the average score.

3.1 Approaches

This study performs dissimilar runs for each type of dataset, each has distinct algorithms for different phases of the ER. We have Standard and Token Blocking methods in the blocking phase; Qgram and String Exact techniques for matching; Average Score and Gradient for classification. String Exact is an umbrella term for standard string matching, the actual method depends on the dataset being evaluated.

Blocking Methods. The blocking methods evaluated for this research are Standard Blocking and Token Blocking. We considered Standard Blocking because it is the one used by [8] and it is easier to compare among experiments while Token Blocking is stated to yield high recall by [4] which is also one of the findings for our research questions. To be more specific, this Standard Blocking uses Prefix of k letters [2] to group entities while Token Blocking uses Q-Gram-Based Indexing [3,4] of n letters. Prefix clustering grouped all records whose first n letters of every record in a particular feature share the same values. For the provided datasets, [8] demonstrated that using Prefix of length 3 results in an impressive performance. For example, the authors extracted the first three letters of “givenname” and “surname” of the person dataset as seen in Fig. 1. All records having “ann” in “name” and “joh” in “surname” are placed in the

same block including both “ann, joh” while entities having “lau” in “givenname” and “toledo” in “surname” in the block of “lau, tol”. This results in all records belonging to a specific block. Many of them will share the same blocks while there exist blocks having only one record as seen in Fig. 1.

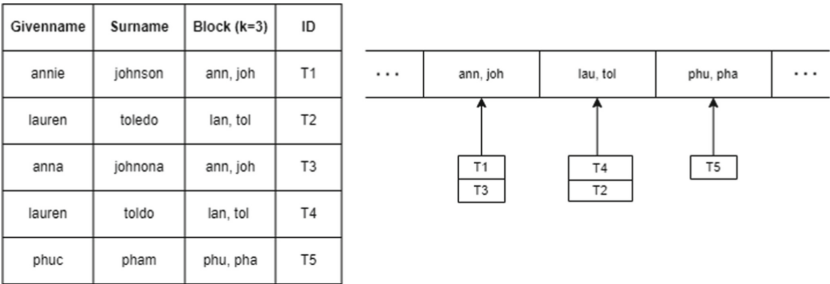


Fig. 1. Standard blocking (Prefix blocking)

Similar to Prefix blocking, Q-Gram-Based Indexing aims to index the records so that all similar data is put in the same block. However, instead of selecting the first three letters, in this research, each record is transformed into 3-gram tokens. Then sub-lists combinations of these 3-gram values are generated based on a selected threshold t , a minimum length l calculated also from this threshold and the number of 3-gram values. Then all created sub-lists are transformed back into strings. Figure 2 illustrates how this process has been done based on $q = 3$ and a threshold $t = 0.8$. This results in the issue that one record can exist in many blocks while a block can have at least one record. The process could be seen in Fig. 2. In addition, the number of comparisons is many more compared to Standard Blocking. For the person dataset, the combination of “givenname” and “surname” is assessed to perform better and the music dataset uses the combination of “artist”, “title” and “album”. While the product dataset, the selected feature is “title” and “manufacturer”.

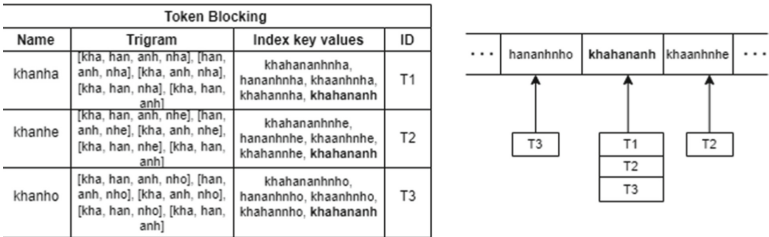


Fig. 2. Q-gram-based indexing (Token Blocking) with samples, trigram, threshold $t = 0.8$

Matching Methods. The matching phase is in charge of finding matches by using similarity functions. There are two types of approaches used in this

research: Qgram and String Exact. What makes these 2 solutions different is that String Exact combines matching or similarity functions which include Qgram. However, they are all categorised as preliminaries. This means that there exists a threshold while comparing 2 records. If the result of the comparison is greater or equal to the threshold, they are considered as matched. If not, they are not matched. As stated by [4], seeking a perfect similarity metric is impossible as it is too restrictive to identify nearly similar matches. Hence, it is suggested that taking a good similarity function is acceptable. In this paper, apart from Qgram (with $q = 3$), we introduce other distance measurements such as Levenshtein, Damerau Levenshtein, Smith-Waterman and Longest Common Subsequence (LCS) as suggested by [11]. These matching functions are implemented on every single feature. If we decide to do Qgram, matching algorithms are implemented on a column. However, if String Exact is picked then we use all five matching functions on that column. Smith-Waterman algorithm [13] is used mostly to tackle local sequence alignment coming from dynamic problems which divide problems into smaller ones and solve these sub-problems prior to combining them to form a complete solution. It finds the optimal local alignment by implementing operators like match, substitution, insertion or deletion. This approach has four main steps: determine the substitution matrix and gap penalty scheme, initialise the score matrix, score and trace back. This algorithm is considered to be used when accuracy is more important than execution time. Longest Common Subsequence (LCS) is a technique used to find the length of the longest subsequence presented in two sequences as long as the characters of the subsequence do not necessarily have consecutive positions in the parent sequences. There exist several approaches to implementing this technique, however, the most common one is dynamic programming [1]. Qgram matching algorithm is one of the most suitable fuzzy string matching for relational databases. It converts a text into a set of q characters. Then they are transformed into vectors for comparison among other converted lists of q -gram values. The result shows how similar the records are. A q -gram of size 1 is referred to as unigram, size 2 is a bigram, and size 3 is a trigram. It is one of the simplest algorithms to be used and provides efficient scalability.

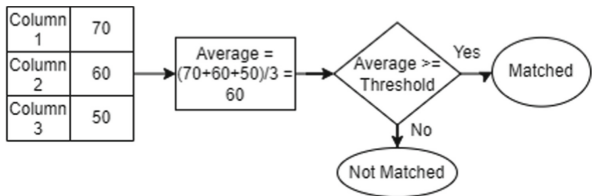


Fig. 3. Aggregate similarity steps using the similarity scores

Entity Clustering Methods. The entity clustering methods used in this research include the aggregate similarity and a machine learning technique called gradient descent back-propagation algorithm as suggested by [11]. The average score is determined by the total values of matching scores divided by the number

of scoring features. For example, after the matching phase of Qgram approach, the DSP has four scoring columns responding to “givenname”, “surname”, “sur-burb”, “postcode” then the average score is the total score of these four columns and divided by four. While for String Exact, each column has five matching scores so the average score is the total of twenty scoring columns and divided by twenty. However, an entity is considered as matched to another only if that average similarity value exceeds or equals a given threshold. The higher the threshold is, the more likely the compared entities are similar. In this research, the threshold value was 70%. Figure 3 illustrate how aggregate similarity works

Gradient-Based Matching (as shown in Fig. 4) requires a probability model whose dimensions are attribute similarities. As stated in [4], this supervised method uses an aggregate score of chosen features from its structure and parameter to train. After the model is split into training and test sets, the gradient descent back-propagation is implemented with the help of labeled data. At this stage, we considered the sigmoid function as the activation function for the nodes as recommended by [11].

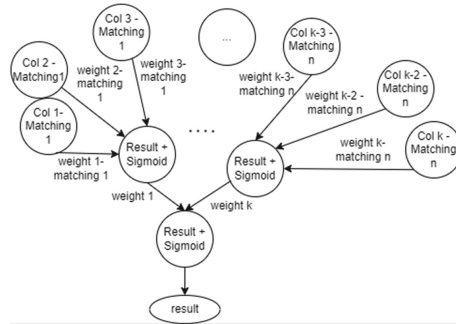


Fig. 4. Gradient-based matching steps

Evaluation. To compare the results from different experiments, we carefully followed the processing steps of ER in [8] with modifications of blocking methods in [3], matching methods and classification guided in [11]. We used the music, person and product datasets as described in Problem Definition. However, we did not use all provided datasets apart from product one as our machine was not strong enough to run them. The music one had 20,000 records and only 5,000 records of people dataset experimented while all of the provided records of product dataset were examined which has 4,589 records. This leads to possible mismatch results between our work and the mentioned approach as guided in [8] in this research. There were twenty different experiments for these three datasets. The music and person datasets ran through eight approaches which are the combination of different algorithms in each phase of ER as seen in Table 4 and 5 while the product data had four experiments.

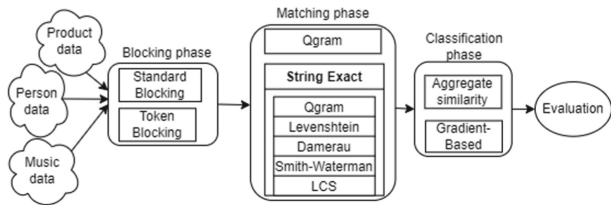


Fig. 5. Work flow

Workflow. As shown in Fig. 5, the process first takes in datasets which are Person, Music and Product. Each of these goes through the process including Blocking, Matching, Classification and Evaluation. At each phase, for one experiment, the data goes on either of the paths. To be more specific, if the input is DSP, in the Blocking phase, it first chooses Standard Blocking, then in the next phase, it chooses Qgram. Lastly, Aggregate similarity is its next choice during the classification phase. Alternatively, the dataset runs through Token Blocking, String Exact and Machine learning or Token Blocking, String Exact and Aggregate similarity. Hence, in this research, we performed twenty different experiments on three datasets. Once done, they are all put into the Evaluation phase to examine how good they are. The metrics are F1, recall and precision, in addition the runtime (duration) of each approach is reported in seconds.

4 Results and Analysis

Table 4. Results of standard methods

Identifier	Dataset	Methods	F1	Recall	Precision	Duration
P-SQA	Person	Standard Blocking+ Qgram+ Aggregate Similarity	0.839	0.952	0.750	0.284
P-SQG	Person	Standard Blocking+ Qgram+ Gradient-Based	0.329	<i>0.751</i>	0.211	0.883
P-SSA	Person	Standard Blocking+ String Exact+ Aggregate Similarity	0.918	0.949	0.890	2.127
P-SSG	Person	Standard Blocking+ String Exact+ Gradient-Based	0.329	<i>0.751</i>	0.211	2.585
M-SQA	Music	Standard Blocking+ Qgram+ Aggregate Similarity	0.280	0.998	0.163	0.856
M-SQG	Music	Standard Blocking+ Qgram+ Gradient-Based	<i>0.268</i>	0.993	<i>0.155</i>	0.806
M-SSA	Music	Standard Blocking+ String Exact+ Aggregate Similarity	0.596	0.999	0.425	16.317
M-SSG	Music	Standard Blocking+ String Exact+ Gradient-Based	<i>0.268</i>	0.993	<i>0.155</i>	9.142
Po-SQA	Products	Standard Blocking+ Qgram+ Aggregate Similarity	0.468	0.853	0.322	0.734
Po-SQG	Products	Standard Blocking+ Qgram+ Gradient-Based	0.519	0.844	0.519	1.163
Po-SSA	Products	Standard Blocking+ String Exact+ Aggregate Similarity	0.534	0.862	0.387	36.723
Po-SSG	Products	Standard Blocking+ String Exact+ Gradient-Based	0.519	0.844	0.519	36.164

To compare the results presented in [8, 11], we carefully followed the processing steps. The initial algorithm is a combination of Standard Blocking, Qgram in the matching phase and Aggregate similarity in the classification case (SQA). All the identifiers and results for the experiments are listed in Table 5 and 4 respectively for different domains and with details. The identifiers were created based on the first few letters of the dataset and its following methods.

Table 5. Results of extended methods

Identifier	Dataset	Methods	F1	Recall	Precision	Duration
P-TQA	Person	Token Blocking+ Qgram+ Aggregate Similarity	0.840	0.957	0.750	289.90
P-TQG	Person	Token Blocking+ Qgram+ Gradient-Based	<i>0.283</i>	<i>0.781</i>	0.173	287.79
P-TSA	Person	Token Blocking+ String Exact+ Aggregate Similarity	0.896	0.948	0.850	284.25
P-TSG	Person	Token Blocking+ String Exact+ Gradient-Based	0.284	<i>0.781</i>	0.173	291.10
M-TQA	Music	Token Blocking+ Qgram+ Aggregate Similarity	0.284	0.998	<i>0.166</i>	36.458
M-TQG	Music	Token Blocking+ Qgram+ Gradient-Based	0.284	0.979	0.166	36.760
M-TSA	Music	Token Blocking+ String Exact+ Aggregate Similarity	0.605	0.999	0.433	51.830
M-TSG	Music	Token Blocking+ String Exact+ Gradient-Based	0.284	0.979	<i>0.166</i>	50.570

The benefit of Standard Blocking over Token Blocking is fewer comparisons while maintaining high recall. In most of the experiments involving Standard Blocking for person dataset, three out of four experiment results including P-SQA, P-SSA, P-SSG of 83%, 91% and 32% for F1 score respectively performed better than those of Token Blocking which are P-TQA, P-TSA, P-TSG - 84%, 89% and 28%. However, for the music dataset, no method using standard blocking had a higher F1 score than those using token blocking. Due to the fact that the person data is much cleaner, formatted, unified and readable so it does not need to have many comparisons to see which one is better. While for music dataset, it is dirty and messy so having more comparisons to cover more possibilities of comparing is recommended by perhaps lowering the threshold of distance measure or putting entities in more comparing collections. When implemented in the product dataset which resembles the real-world dataset, all of the runs regarding the standard blocking phase performed better than the music one. The approach Po-SSA once more demonstrates that it is better than other standard blocking methods with 53% of F1 score.

When looking at matching algorithms, the configurations using String Exact always perform even better or more than Qgram. This trend can be seen in three datasets. For example, for person dataset, P-SSA, P-SSG, P-TSA, P-TSG have better F1 than P-SQA, P-SQG, P-TQA and P-TQG respectively. The reason for this is that having more perspectives and conditions involving calculating

distance measure, finding local sequence alignment and longest subsequence to judge how a pair can be a match is effective as String Exact uses five different matching algorithms even including Q-gram. In addition, in terms of the classification phase, most of the experiments having Gradient-Based outperform Aggregate Similarity, apart from Po-SQG and Po-SQA. In this stage, it is seen easily that the Gradient-Based approach is not necessary for three datasets. It is reasoned that there are not enough records to cover all possibilities to find acceptable weights. Hence, it is demonstrated that the combination of Standard Blocking, String Exact and Aggregate Similarity (P-SSA) achieved the best result for person and product dataset with 92% of F1 and 53% respectively. While the combination of Token, String Exact and Aggregate Similarity (M-TSA) is best for music one with 61% of F1. This points out that for ER, methods do not matter, what matters is the data itself. It is noticeable that the Music domain performed poorly. Only what seemed to be interesting is that the recall values in all eight experiments were really high (around 98%) while the precision ones were really low. This means that there is a large number of false positives, the matching and classification methods are too eager to confirm matches. The results of Music dataset show that blocking methods which are in charge of recall probably need to be more stringent, only matching and classification phase matter due to noticeable differences in precision. It means that there are many potential results but their predicted labels are incorrect when compared to the training ones. This could be answered by the fact that the Music dataset combined different languages and each of them may need different ways to convert to readable formats. For example, records have Japanese characters in one feature while another feature is written in English. This situation is hard to handle as the encoding parts need to identify different languages to get the correct interpretation. Moreover, non-alphabet languages like Chinese and Japanese may need to have different string comparisons or alphabet conversions. In addition, when compared to the person data, none of the combinations of methods produced better results than the person data and M-TSA performed worst compared to others. This statement is also supported by [4] that different data needs different ER approaches as there is no one solution that fits all.

As far as we observed, most of the experiments in person and product domains showed that the token blocking performs better recall than the standard blocking method. The difference is quite small, around 3% or less. It also performed better than the original one in the case of P-TQA. Token blocking results in higher recall in comparison to standard blocking methods though there exist some special cases like a pair P-TSA & P-SSA and M-TQG & M-SQG. The reason why Token Blocking performed better is that it has more computations than Standard Blocking because there are many entities existing in multiple blocks so they are compared to different collections. Lastly, by combining more matching algorithms, especially in the matching phase, we had better results as seen in experiment P-SSA. With the same approach of blocking and classification, the only different method was String Exact which combines different distance string measure methods, the results of the three metrics were higher than the

proposed one. In addition, though String Exact was used in other combinations, it sometimes did not perform well, in the experiment of P-TSA, it outperformed the P-SQA. Overall, all methods are efficient. However, when implementing the Token Blocking, the running time is increased rapidly for the dataset of person and music.

5 Conclusions

In this study, we have experimented with and compared different cases of implementation of several approaches of ER to evaluate their performance in different scenarios. By doing so, we managed to find the answers to our questions regarding suitable combinations of algorithms. However, more work is required to fully evaluate the effects of method combinations. Our future work focuses on the incorporation of more ML approaches into the ER pipeline.

References

1. Bergroth, L., Hakonen, H., Raita, T.: A survey of longest common subsequence algorithms. In: *Proceedings Seventh International Symposium on String Processing and Information Retrieval (SPIRE 2000)*, pp. 39–48 (2000)
2. Christen, P.: The data matching process. In: *Data Matching. Data-Centric Systems and Applications*, pp. 23–35. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31164-2_2
3. Christen, P.: A survey of indexing techniques for scalable record linkage and deduplication. *IEEE Trans. Knowl. Data Eng.* **24**(9), 1537–1555 (2012)
4. Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., Stefanidis, K.: An overview of end-to-end entity resolution for big data. *ACM Comput. Surv.* **53**(6), 1–42 (2020)
5. Isele, R., Bizer, C.: Learning expressive linkage rules using genetic programming. *Proc. VLDB Endow.* **5**(11) (2012)
6. Jurek, A., Hong, J., Chi, Y., Liu, W.: A novel ensemble learning approach to unsupervised record linkage. *Inf. Syst.* **71**, 40–54 (2017)
7. Köpcke, H., Thor, A., Rahm, E.: Evaluation of entity resolution approaches on real-world match problems. *Proc. VLDB Endow.* **3**(1–2), 484–493 (2010)
8. Nentwig, M., Rahm, E.: Incremental clustering on linked data. In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 531–538 (2018)
9. Papadakis, G., Skoutas, D., Thanos, E., Palpanas, T.: Blocking and filtering techniques for entity resolution: a survey. *ACM Comput. Surv.* **53**(2), 1–42 (2020)
10. Papadakis, G., Tsekouras, L., Thanos, E., Giannakopoulos, G., Palpanas, T., Koubarakis, M.: JedAI: the force behind entity resolution. In: Blomqvist, E., Hose, K., Paulheim, H., Lawrynowicz, A., Ciravegna, F., Hartig, O. (eds.) *ESWC 2017. LNCS*, vol. 10577, pp. 161–166. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70407-4_30
11. Reyes-Galaviz, O.F., Pedrycz, W., He, Z., Pizzi, N.J.: A supervised gradient-based learning algorithm for optimized entity resolution. *Data Knowl. Eng.* **112**, 106–129 (2017)

12. Saeedi, A., Peukert, E., Rahm, E.: Using link features for entity clustering in knowledge graphs. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 576–592. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_37
13. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. *J. Mol. Biol.* **147**(1), 195–197 (1981)
14. Vatsalan, D., Christen, P., Rahm, E.: Incremental clustering techniques for multi-party privacy-preserving record linkage. *Data Knowl. Eng.* **128**, 101809 (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Safe Lane-Changing in CAVs Using External Safety Supervisors: A Review

Lalu Prasad Lenka^(✉) and Mélanie Bouroche^{}

Trinity College Dublin, Dublin, Ireland
{lenkal,melanie.bouroche}@tcd.ie

Abstract. Connected autonomous vehicles (CAVs) can exploit information received from other vehicles in addition to their sensor information to make decisions. For this reason, their deployment is expected to improve traffic safety and efficiency. Safe lane-changing is a significant challenge for CAVs, particularly in mixed traffic, i.e. with human-driven vehicles (HDVs) on the road, as the set of vehicles around them varies very quickly, and they can only communicate with a fraction of them. Many approaches have been proposed, with most recent work adopting a multi-agent reinforcement learning (MARL) approach, but those do not provide safety guarantees making them unsuitable for such a safety-critical application. A number of external safety techniques for reinforcement learning have been proposed, such as shielding, control barrier functions, model predictive control and recovery RL, but those have not been applied to CAV lane changing.

This paper investigates whether external safety supervisors could be used to provide safety guarantees for MARL-based CAV lane changing (LC-CAV). For this purpose, a MARL approach to CAV lane changing (MARL-CAV) is designed, using parameter sharing and a replay buffer to motivate cooperative behaviour and collaboration among CAVs. This is then used as a baseline to discuss the applicability of the state-of-the-art external safety techniques for reinforcement learning to MARL-CAV. Comprehensive analysis shows that integrating an external safety technique to MARL for lane changing in CAVs is challenging, and none of the existing external safety techniques can be directly applied to MARL-CAV as these safety techniques require prior knowledge of unsafe states and recovery policies.

Keywords: Connected autonomous vehicles · Lane changing · Multi-agent reinforcement learning · Safe reinforcement learning

1 Introduction

Improvements in traffic mobility, eco-friendly vehicles, fewer fossil fuels and safer roads are among the benefits promised by the autonomous vehicle industry. An autonomous vehicle can be defined as a vehicle capable of travelling unassisted anywhere and at any time, with no restrictions, and without the help or even the

presence of a driver [1]. A connected autonomous vehicle (CAV) is an autonomous vehicle that can communicate with other vehicles or roadside units [1].

Though there has been great advancement in AV technology over the past decade, the number of traffic accidents involving autonomous vehicles has increased in recent years [2]. A major problem with automated driving at its current stage of development is that it is not yet reliable and safe [3].

As it is such a complex problem, a lot of the recent approaches to (connected) autonomous driving rely on artificial intelligence, and reinforcement learning (RL) in particular. RL agents, however, might visit unsafe states during the exploration phase, and there is no guarantee that they will not explore them during the exploitation phase.

This paper first reviews existing work in RL-based lane changing, safety approaches for lane changing and safety approaches for MARL more generally (Sect. 2). Recent papers addressing safe lane-changing of connected autonomous vehicles and safe multi-agent reinforcement learning were selected for review. It then presents a MARL design of lane changing for CAVs, building on the state of the art (Sect. 3). This design is used as a basis to discuss open challenges in safe MARL CAV lane changing (Sect. 4). Finally, the Sect. 5 concludes the paper.

2 Background

This section discusses the state-of-the-art lane changing approaches for CAVs (Sect. 2.1), before presenting the existing approaches to introduce safety for CAV lane changes (Sect. 2.2). Finally, it analyses the safety approaches for MARL (Sect. 2.3) applied in recent research.

2.1 RL-Based Lane Changing Approaches for CAV

Lane changing remains a challenging task where an autonomous vehicle has to predict the actions of neighbouring vehicles while changing lanes to avoid a collision. Furthermore, an autonomous vehicle is supposed to undertake the lane change manoeuvring without aggressive acceleration or deceleration to ensure the comfort of the passengers. The complexity of this problem has led many recent approaches to adopt a RL approach to lane changing. These RL-based autonomous vehicles controllers are dominated by the actor-critic methods, and the Deep Q Networks (DQN) and Deep Deterministic Policy Gradient (DDPG) algorithms [4]. Deep RL approaches are favoured over normal RL methods for high-dimensional state spaces. Policy-based methods are preferred in multi-agent setting. Ye et al. [7] proposed an automated mandatory lane change strategy using proximal policy optimization-based deep reinforcement learning, showing great advantages in learning efficiency and performance compared to Q-learning-based RL approaches.

One way of enhancing safety in multi-agent RL is through introducing cooperative behaviour and collaboration among agents. Fu et al. [5] modelled the lane-changing problem as a deep reinforcement learning process to learn the

optimal lane-changing strategy through a deep deterministic policy gradient (DDPG) algorithm. They also proposed a collective learning framework to use the collective intelligence of CAVs to improve the performance of autonomous lane-changing strategies. Zhou et al. [6] proposed a decentralized cooperative multi-agent reinforcement learning algorithm with an actor-critic policy for lane changing among CAVs, where they used parameter-sharing scheme to foster inter-agent collaborations.

Learning an optimal RL policy for lane changing can be challenging in the presence of mixed traffic where human driven vehicles (HDVs) are present along with controlled agents. Chen et al. used parameter sharing and local rewards to foster inter-agent cooperation and achieved high scalability in mixed traffic scenarios [8].

2.2 Safety Approaches in RL-Based Lane Changing for CAV

In the recent literature, most RL-based lane changing approaches introduced safety into the algorithm by modifying reward function, to include some parameters related to headway distance, collision avoidance, and passenger comfort.

Fu et al. discussed a blockchain-based collective learning framework for lane-changing in CAVs and introduced the headway distance (i.e. the distance between ego-vehicle and leading vehicle) into the reward function to make the lane changes safer [5]. Ye et al. introduced collision reward to penalize unsafe actions, and used a safety intervention module to label the output action from the algorithm as “catastrophic” or “safe” [7]. However, the details of the safety intervention module were not discussed.

Zhou et al. proposed the use of multi-agent actor-critic RL for cooperative lane changing among autonomous vehicles using the headway distance and a collision penalty in the reward function to add safety into their design [6]. Chen et al. [8] also used the headway distance and a collision penalty in the reward function to add safety, but also proposed a novel priority-based safety supervisor, which predicts the action of neighbouring vehicles to enable safer decisions. This is the first use of an external safety technique for CAV lane changing. However, though their approach is quite intuitive they do not discuss non-cooperative human driven vehicles.

Most papers discussed in this section focused on optimizing policies based on rewards and adding safety constraints to the reward function. But none truly guarantees safety, i.e., that no unsafe state is ever visited during the training and execution process.

2.3 Safety Approaches in MARL

Deep reinforcement learning techniques are able to maximize the intended reward, but they may not always ensure safety throughout the learning or execution stages. Safe Reinforcement Learning can be defined as the process of learning policies that maximize the expectation of the return, in problems in which it is

important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes [10]. In the exploration phase the agent might encounter some unsafe states which makes Reinforcement Learning approach unsuitable for safety-critical systems as in this case failure can be costly [11].

According to García & Fernández [10] generally safety can be introduced into RL algorithm through two ways. The first way is to modify the reward function to include parameters for safety, so the safety is improved while the algorithm learns to optimize the reward function. The second approach is where the RL algorithm is modified to account for an external safety supervisor (such as teacher advice or demonstrations).

External safety approaches can be classified into two groups; the first type of technique uses some prior information about unsafe states to develop a safety critic that can provide the probability of agents being in unsafe situations in future states, and uses a separate recovery policy to bring the agents back to safe states. Model predictive control predicts the next states of each agent when the learnt policy is followed. If the predicted next states are recoverable for all agents, it uses the learnt policy. Otherwise it uses a recovery policy for the agents who will move into irrecoverable states after the current action [12, 13]. Thananjeyan et al. proposed using a composite policy π which selects between a task-driven policy and a recovery policy at each time step, based on whether the agent will violate safety constraints in the near future [11].

The Second type of techniques creates a set of safe states by estimating the dynamics of the system. They learn about unsafe states, project the actions of the RL agents into the safe set, and constrain unsafe actions. Control Barrier Functions (CBF) is a model-based safety framework that learns about safe states and prevents the exploration of dangerous states by projecting the RL agents' actions onto a safe set of actions. It has been used by a number of recent works [14–17] to introduce external safety in reinforcement learning tasks. ElSayed-Aly et al. specified safe states using finite state machines and used Linear Temporal Logic (LTL) as modal temporal logic to formally verify whether the visited state is a safe state [19].

Overall, according to our research there are four key external safety supervisor techniques used in the RL literature, i.e., shielding, control barrier functions, model predictive control, and recovery RL. This paper aims to study recent papers using these techniques and evaluate whether these approaches can be applied to improve the safety of lane changing in CAVs.

3 Modeling MARL-CAV

In this section, a decentralized MARL-based approach [20] for highway lane-changing of multiple CAVs is discussed. This approach is a **customized and adapted** version of the MARL ramp merging algorithm in Chen et al. [8]. It was modified for the lane changing scenario and a custom reward function was added (discussed below).

The mixed-traffic lane-changing environment is modelled as a multi-agent network: $\mathcal{G} = \{v, \varepsilon\}$, where each agent $i \in v$ communicates with neighbours \mathcal{N}_i using the communication link $\epsilon_{ij} \in \varepsilon$. Here \mathcal{N}_i represents the set of agents in close proximity to agent i . The overall dynamic system can be considered a partially-observable Markov decision process (POMDP) which can be represented by the tuple $(\{\mathcal{A}_i, \mathcal{S}_i, \mathcal{R}_i\}_{i \subseteq v}, \mathcal{T})$, where \mathcal{A}_i is the local action space, \mathcal{R}_i is the reward space, $\mathcal{O}_i \in \mathcal{S}_i$ is the partial observation of the environment state [6].

In partially observable Markov games (multi-agent POMDP), every agent follows a decentralized policy $\pi_i : \mathcal{O}_i \times \mathcal{A}_i \rightarrow [0, 1]$ to choose its own action $a_{i,t} \sim \pi_i(\cdot | s_{i,t})$ at time step t . The action space \mathcal{A}_i of agent i is defined as a set of high-level **discrete control decisions (actions)**, including: cruising, turn left, turn right, speed up and slow down. The state space \mathcal{O}_i contains the longitudinal and latitudinal features (speed and position). The reward function is designed to achieve the goal of safe lane changing. Our reward function is composedly designed using multiple metrics like safety, headway distance, driving speed, right lane driving, lane changing:

$$r_{i,t} = w_s r_s + w_h r_h + w_d r_d + w_{rl} r_{rl} + w_{lc} r_{lc} \quad (1)$$

where w 's are weighing coefficients and r 's are cost evaluation.

In the proposed approach a deep neural network is used to approximate the stochastic decentralized policy π of the RL agents. This network is shared between all agents. In addition, a shared replay buffer is also maintained to store the experiences from all agents. A copy of the state information i.e. observations, actions and rewards, is held by the individual agents. No agent has access to the state information of any other agent. However, as the data in the Replay Buffer is shared and identical, each agent benefits from the collective experiences of all agents. Finally, each agent updates the policy network asynchronously at each step [9]. In this design, the MARL algorithm does not have access to information about unsafe states and recovery policies.

4 Using an External Safety Technique for MARL-CAV

The design of MARL-CAV discussed in Sect. 3 only uses the reward function to introduce some level of safety. This is achieved by penalising actions which lead to a collision and rewarding the agents when they maintain a suitable headway distance from the front vehicle. However, this is not sufficient to prevent the agents from visiting unsafe states especially during the exploration phase and subsequently in the exploitation phase. Hence, an external safety supervisor might be useful to stop the agent from visiting unsafe states.

An external safety supervisor can be applied to enhance the agents' safety. The safety approaches discussed in Sect. 2.3 namely shielding, control barrier functions, model predictive control, and recovery RL, should integrate with MARL-CAV and then we can examine whether these techniques can improve safety.

This section first discusses the safety requirements for lane changing in CAV (Sect. 4.1), and then uses them to analyse the suitability of the external safety techniques for MARL CAV. Finally, it discusses open challenges.

4.1 Safety Requirements for Lane Changing in CAV

AI approaches are often validated on simple scenarios. In contrast, lane changing is a challenging scenario. Not only it contains an unspecified and unbounded number of controlled agents, but also (an unspecified and unbounded) number of uncontrolled agents (e.g., human-driven vehicles). Also, to fit with our approach (Sect. 3), the safety approach should be compatible with discrete action space as this library only supports discrete actions: move to the left lane, move to the right lane, forward, idle.

The Table 1 shows the requirements specific to the MARL-CAV that the safety approach should satisfy. The next section analyses the applicability of the safety techniques to MARL-CAVs and examines whether they satisfy the requirements mentioned in the Table 1.

Table 1. Characteristics of Lane Changing in CAVs

Characteristic	Safety approach requirement
Number of agents	Should support a multi-agent scenario. It should be scalable to a large number of agents
Mixed traffic scenario	Should support the presence of both controlled and uncontrolled agents in the environment
Action Space	Should support discrete actions as the current implementation of MARL-CAV has discrete actions
Unsafe states information	Should not require prior information about unsafe states as the simulation is very dynamic
Recovery/Backup Policies	Should not require a recovery or backup policy developed to get the agent from unsafe to safe states

4.2 Analysis of External Safety Techniques for MARL

Section 2.3 showed that, generally, control-theoretic or formal methods like shielding, control barrier functions, model predictive control and recovery policies are used to introduce external safety in MARL implementation for different domains. This section presents a more detailed analysis of these external safety approaches.

Shielding. In the shielding method, finite state machines are used to model all possible states an agent can visit, and linear temporal logic (LTL) safety specifications are used to design shields that restrict the agent from visiting unsafe states. ElSayed-Aly et al. [19] claim that the shielding approach not only guarantees safety but also learns more optimal policies with better returns than non-shield MARL, as unsafe actions which may destabilize learning are removed. However, shielding requires prior knowledge of safe states in the environment so that they can be used to design shields. Safety is specified using Linear Temporal Logic (LTL). This might be possible for a simpler environment like navigation tasks in ElSayed-Aly et al. [19] but could be difficult for complex environments like lane changing in connected autonomous vehicles. As discussed in the Table 1, there is no prior information about unsafe states in the MARL-CAV design.

Control Barrier Functions. Control Barrier Functions (CBF) is a model-based safety framework that prevents the exploration of dangerous states by projecting the RL agent’s actions onto a safe set of actions. With CBFs safety is specified by defining a forward invariant set in the state space in which the system is **required to stay**. Given a time-varying set $\mathcal{C} \subset \mathbb{R}^n$ defined as zero superlevel set of a continuously differentiable function $h : \mathbb{R}^n \times \mathbb{R}_+ \rightarrow \mathbb{R}$

$$\mathcal{C} \triangleq \{x \in \mathcal{X} : h(x, t) \geq 0\} \quad (2)$$

where \mathcal{C} is the safe set, h is the CBF. The idea of forward invariance is that if we have a state, we want to make sure that the state of the system would stay inside the set for long time. Control Barrier Functions can be used as a formulation tool to achieve forward invariance and, therefore the safety of a set. This is a very promising idea and hence control barrier function is the most widely used safety technique out of the four techniques discussed.

As discussed in Sect. 2.3, several approaches to MARL have introduced external safety through CBFs. These papers have simulated different tasks such as navigation tasks, i.e. an agent aims to move from a source to destination like in [14, 15], or any OpenAI tasks like car following environment [17], unicycle environment [17].

These approaches [15, 18] used continuous action spaces as CBF would not work for a discrete action space. Control barrier functions do not work for discrete action space because with discrete action space the RL model cannot be represented using differential dynamic programming and the lie derivatives would not work. This is a major drawback as per the safety requirements discussion in Sect. 4.1.

A key problem for this approach is figuring out how to combine knowledge of the model (environment) dynamics with model-based safety, as control barrier functions are model-based i.e. they require information about model dynamics. Hence most of these papers either use model-based RL like in [17] or use a statistical model to learn model dynamics like in [15, 18]. However, in POMDP discussed in Sect. 3 the system dynamics can be very uncertain. There is no guarantee that a complex setting of multi-agent RL for lane changing with mixed

traffic scenario can be described with some statistical assumptions of dynamics. Complex environment dynamics of MARL-CAV makes use of CBF challenging for lane changing in CAVs.

Model Predictive Control. Model predictive control can be defined as single or multi step ahead estimation of states that can be reached by an agent over multiple next time steps under a sequence of control inputs. Generally, statistical models are used to approximate the system dynamics, which helps to predict the successive states of the environment when an action is taken by the agent. The predicted states are then verified if they violate safety constraints and if an agent has a high probability of reaching unsafe states, it uses a recovery (or backup) policy that brings it back to safe states.

MPC can also be applied in a multi-agent setting, like in [12]. In their proposed method, they are incrementally checking whether each agent is in a set of stable states χ_{stable} , and if any agent is predicted to be going into an irrecoverable or unsafe state, then recovery policy $\pi_{recovery}$ is used to bring the agent back to safe states χ_{stable} . The agents predicted to be in safe set after current action use the respective learned policy π^i . Unlike our implementation of parameter sharing discussed in Sect. 3 they have assumed different learned policy for each agent which is not scalable to higher number of agents.

This approach looks promising and able to act as a safety supervisor for the RL agent. However, these approaches generally have two underlying assumptions. First, availability of a recovery policy, also called safe policy π_{safe} is used to return the agent to safe state when it is about to visit the unsafe states. Second, some prior information of either the irrecoverable (unsafe) states or safe states.

The challenge is that the recovery policy and data of unsafe states for an environment might not be available for all use cases. For example, in a multi-agent lane changing the setting, it is difficult to know which all states are unsafe and how one trains a recovery policy. These challenges are limitations to using this safety technique for multi-agent lane-changing scenarios.

Recovery RL. The Recovery RL approach is relatively very recent compared to the previous three approaches. Thananjeyan et al. [11] proposed a unique approach to create an external supervisor for RL algorithms. However, it has some similarities to model predictive control methods, especially the one in [12].

According to Thananjeyan et al. [11], an offline dataset $D_{offline}$ was generated which contains set of trajectories where the agent violated safety constraints. This was either generated manually using human knowledge or through a naive RL policy. This $D_{offline}$ data was used to train a reinforcement learning-based safety-critic policy. The safety critic was later used to estimate the agent's probability of future constraint violations. If the safety-critic predicts the agent's action to be unsafe, then recovery policy $\pi_{recovery}$ is used to bring it back to safe states or else a learn policy π_{task} is used.

This approach requires us to design the $D_{offline}$ either manually using human knowledge or through a separate reinforcement learning policy. For complex

systems like the multi-agent RL model for lane changing in CAVs, it would require the first training an RL policy and extracting the $D_{offline}$. Though the $D_{offline}$ would be updated during the training of the recovery RL it does require to run the RL model without the safety critic to collect $D_{offline}$. This in-turn defeats the purpose of designing an external safety supervisor, as the main aim of external safety is to stop the agent from exploring unsafe states.

The Table 2 provides the analysis of safety techniques concerning the safety requirements. The green color cell means the safety method satisfies safety requirements. From the table, it can be inferred that none of the safety techniques satisfies all the safety requirements; hence, there are still some open challenges in using them for MARL-CAV. These challenges are discussed in more detail in the next section.

Table 2. Analysis of the External Safety Supervisors for MARL in terms of the scenario requirements. Green color cell means scenario requirement is satisfied

Characteristic	Shielding	Control Barrier Functions	Model Predictive Control	Recovery RL
Number of agents	Supports multi-agent	Supports multi-agent	Supports multi-agent	Supports multi-agent
Mixed agents (controlled & uncontrolled)	No support yet	No support yet	No support yet	No support yet
Action Space	Continuous	Continuous	Continuous	Continuous and Discrete
Model dynamics information	Required apriori	Modelled using Gaussian Processes or Assumed	Modelled using Gaussian Processes, or Assumed	Learned using RL
Unsafe states information	Required	Not required	Required	Required
Recovery/Backup Policies	Not required	Not required	Required	Required

4.3 Discussion

As discussed in Sect. 2.2, most MARL methods for CAV modelling mostly focus on achieving optimal policies based on their reward function but this does not guarantee safety, i.e., that no unsafe state is ever visited during the learning process. This is because penalizing agents through negative rewards does not stop them from going to unsafe states as they have to explore different (both safe and unsafe) actions to learn about the states and rewards from the environment.

The approaches discussed in Sect. 4.2, namely shielding, model predictive control, control barrier functions and recovery RL, are ways to guarantee safety during the exploration process as they act like an external supervisor, thereby monitoring the agents’ actions and preventing the exploration of unsafe states.

These approaches have been used to provide external safety for MARL for other scenarios and might be applicable to the multi-agent reinforcement learning algorithms for lane changing in CAVs. The multi-agent proximal policy optimization algorithm has to be modified to include one of these safety approaches.

However, this implementation will be a significant challenge. Based on the data in Table 2 it can be inferred that most of these approaches are applied to simpler tasks, such as one or more agents trying to reach a goal [14, 15], navigation tasks [11] or simple OpenAI gym environments [17]. These approaches use statistical models to estimate the dynamics of the system, but this might not work for a complex scenario like lane-changing in CAVs, where other human-driven vehicles, i.e., uncontrolled agents, are present. Compared to the static obstacles present in simpler environments, these represent dynamic obstacles. Furthermore, as discussed in Sect. 4.2, these approaches come with several assumptions such as:

- Availability of partial or complete prior knowledge of unsafe or irrecoverable states in shielding, model predictive control and recovery RL.
- Availability of recovery policies in the case of model predictive control and recovery RL.
- Bounded system dynamics for control-theoretic methods like control barrier functions.
- Having only a single or more controlled agents in the system. Absence of uncontrolled agents (like human-driven vehicles in the lane-changing scenarios).

As per the safety requirements in Sect. 4.1, these assumptions will be violated in the MARL-CAV setting discussed in Sect. 3 and this poses a significant challenge in adapting these approaches to MARL-CAV. Based on the assumptions and limitations discussed in this chapter, the potential introduction of these safety approaches to MARL-CAV comes with theoretical and practical challenges.

5 Conclusion

The investigation of recent literature showed that most of the state-of-the-art implementations of multi-agent reinforcement learning (MARL) for lane changing in CAVs use a custom reward function to introduce safety, but this is not completely safe as the agents still visit unsafe states during the exploration phase of the RL algorithm. The characteristics of safe MARL for lane changing scenarios were discussed (summarised in Table 1) and state-of-the-art external safety techniques for MARL were researched and analysed. It was inferred that an ideal safety supervisor approach should support both continuous and discrete actions, should operate in a mixed traffic scenario, should not require a recovery policy or any prior information about unsafe states. Based on these, the functioning and limitations of different external safety supervisors for multi-agent RL were analysed and summarised in Table 2. It was observed that these safety techniques

come with many assumptions like availability of recovery policies, prior information about unsafe states, the requirement of continuous action space etc. and are often not designed for mixed agents (controlled and uncontrolled) scenarios.

These assumptions mean these techniques cannot be directly applied to MARL for CAVs. Integration of these safety approaches to MARL-CAV is not straightforward and requires significant work, which involves modifying either the MARL-CAV implementation or the safety techniques. In future work we aim to modify the design of MARL-CAV and make tweaks in safety supervisor techniques to enable successful integration of the safety techniques with MARL-CAV.

References

1. Paret, D., Rebaine, H., Engel, B.A.: The buzz about autonomous and connected vehicles, pp. 3–22. Wiley (2022)
2. Dixit, V.V., Chand, S., Nair, D.J.: Autonomous vehicles: disengagements, accidents and reaction times. *PLOS One* **11**(12), e0168054 (2016)
3. Martens, M., van den Beukel, A.: The road to automated driving: dual mode and human factors considerations. In: ITSC 2013, pp. 2262–2267 (2013)
4. Haydari, A., Yılmaz, Y.: Deep reinforcement learning for intelligent transportation systems: a survey. *IEEE Trans. Intell. Transp. Syst.* **23**(1), 11–32 (2022)
5. Fu, Y., Li, C., Yu, F.R., Luan, T.H., Zhang, Y.: An autonomous lane-changing system with knowledge accumulation and transfer assisted by vehicular blockchain. *IEEE Internet Things J.* **7**(11), 11123–11136 (2020)
6. Zhou, W., Chen, D., Yan, J., Li, Z., Yin, H., Ge, W.: Multi-agent reinforcement learning for cooperative lane changing of connected and autonomous vehicles in mixed traffic. *Auton. Intell. Syst.* **2**(1), 5 (2022)
7. Ye, F., Cheng, X., Wang, P., Chan, C.-Y., Zhang, J.: Automated lane change strategy using proximal policy optimization-based deep reinforcement learning. In: 2020 IEEE Intelligent Vehicles Symposium (IV), pp. 1746–1752 (2020)
8. Chen, D., et al.: Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic (2022). <https://doi.org/10.48550/arXiv.2105.05701>
9. Kaushik, M., Singhanian, N., Krishna, K.M.: Parameter sharing reinforcement learning architecture for multi agent driving. In: AIR 2019, Association for Computing Machinery, New York (2019). <https://doi.org/10.1145/3352593.3352625>
10. García, J., Fernández, F.: A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.* **16**(1), 1437–1480 (2015)
11. Thananjeyan, B., et al.: Recovery RL: safe reinforcement learning with learned recovery zones (2020). <https://arxiv.org/abs/2010.15920>
12. Zhang, W., Bastani, O., Kumar, V.: MAMPS: safe multi-agent reinforcement learning via model predictive shielding (2019). <https://arxiv.org/abs/1910.12639>
13. Zanon, M., Gros, S.: Safe reinforcement learning using robust MPC. *IEEE Trans. Autom. Control* **66**(8), 3638–3652 (2021)
14. Cai, Z., Cao, H., Lu, W., Zhang, L., Xiong, H.: Safe multi-agent reinforcement learning through decentralized multiple control barrier functions (2021)
15. Qin, Z., Zhang, K., Chen, Y., Chen, J., Fan, C.: Learning safe multi-agent control with decentralized neural barrier certificates (2021)

16. Ames, A.D., Xu, X., Grizzle, J.W., Tabuada, P.: Control barrier function based quadratic programs for safety critical systems. *IEEE Trans. Autom. Control* **62**(8), 3861–3876 (2017)
17. Emam, Y., Glotfelter, P., Kira, Z., Egerstedt, M.: Safe model-based reinforcement learning using robust control barrier functions (2021). <https://arxiv.org/abs/2110.05415>
18. Zhao, H., Zeng, X., Chen, T., Liu, Z., Woodcock, J.: Learning safe neural network controllers with barrier certificates. *Formal Aspects Comput.* **33**(3), 437–455 (2021). <https://doi.org/10.1007/s00165-021-00544-5>
19. ElSayed-Aly, I., Bharadwaj, S., Amato, C., Ehlers, R., Topcu, U., Feng, L.: Safe multi-agent reinforcement learning via shielding. In: *AAMAS 2021, International Foundation for Autonomous Agents and Multiagent Systems*, Richland, SC, pp. 483–491 (2021)
20. Leurent, E.: An environment for autonomous driving decision-making (2018). <https://github.com/eleurent/highway-env>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Author Index

A

Aftab, Danyal 147
Agrahari, Rahul 134
Akbar, Awais 502
Alam, Tanvir 147
Alfano, Marco 175
Ali, Hazrat 32, 147
Asadi, Hamzeh 106
Awasthi, Anshul 488

B

Becker, Brett A. 201
Bezbradica, Marija 355
Bierig, Ralf 502
Bouroche, Mélanie 527
Bowden, David 332
Brennan, Rob 251
Bridge, Derek 279
Brown, Kenneth N. 106, 305

C

Campbell, Mark J. 95
Carraro, Diego 305
Caton, Simon 502
Cavadas, Joana 450
Cepeda Zapata, Karla Anielá 163
Cooney, Ciaran 450
Coscrato, Victor 279
Crane, Martin 355
Cullen, Gary 40

D

Dair, Zachary 3
Delany, Sarah Jane 214, 226
Dev, Soumyabrata 134
Dockray, Samantha 3
Dogan, Vedat 409
Dolphin, Rian 81
Dong, Ruihai 81

E

Elkelany, Amany 121

G

Ghanadbashi, Saeedeh 462
Golpayegani, Delaram 189
Golpayegani, Fatemeh 462
Griffith, Josephine 318
Grimes, Diarmuid 332, 397, 437

H

Hayes, Conor 292
Hazman, Muzhaffar 318
Helfert, Markus 175
Henna, Shagufta 40
Heyburn, Rachel 450
Hogan, Ciarán 264
Hojaji, Fazilat 95
Hossari, Murhaf 134
Huu, Phuc Pham 514

I

Ibrahim, Mohamed 40
Isakowitsch, Clara 239

J

Jain, Nishtha 134
Jeyaraj, Manuela Nayantara 214
Jones, Gareth J. F. 381
Jouda, Ahmed 368

K

Kelleher, John D. 68
Kellett, John 175

L

Lambert, Julie 106
Lenka, Lalu Prasad 527
Lenzitti, Biagio 175
Lewis, Dave 189
Loughran, Róisín 163

M

Madigan, Liam 450
Mahon, Joyce 201

Marzo, Stefano 251
 McCaffery, Fergal 163
 McKeever, Susan 121
 McKeever, Susan 318
 McKenna, Lucy 251
 McKeon, Carla 381
 Muehlhausen, Willie 475
 Mukherjee, Ruben 475
 Mulcahy, Eoghan 344
 Murad, Shafaq 32

N

Namee, Brian Mac 201
 Nelson, John 344
 Nguyen, An Pham Ngoc 355
 Nie, Dongyun 514

O

O' Mahony, Niall 488
 O'Cuinn, Mairead 450
 O'Mahony, Tom 106
 O'Reilly, Ruairi 3, 18
 O'Sullivan, Barry 397, 437
 O'Sullivan, Declan 68
 Orlandi, Fabrizio 134

P

Pandit, Harshvardhan J. 189
 Patil, Ankur 134
 Pinto, Royston 251
 Prestwich, Steven 409

Q

Qureshi, Rizwan 147

R

Ramo, Mirco 55
 Rehmani, Mubashir Husain 18

Riad, Maha 462
 Riordan, Daniel 488
 Rocha, Claudio 381
 Ross, Robert 121

S

Saad, Muhammad Muneeb 18
 Sardina, Callie 68
 Sardina, Jeffrey 68
 Scriney, Michael 514
 Shah, Zubair 32, 147
 Silvestre, Guénolé C. M. 55
 Sistu, Ganesh 264
 Smyth, Barry 81
 Sobhani, Nasim 226
 Souza, Filipe 397, 437

T

Tariq, Usama 147
 Thompson, Chloe 450
 Toth, Adam J. 95

V

Van de Ven, Pepijn 344

W

Wadhai, Piyush 475
 Wallace, Richard J. 423
 Walsh, Joseph 488
 Ward, Tomás 163
 Ward, Tomas 475
 Wu, Jia 147

Z

Zafar, Anas 147
 Zhang, Lili 475
 Zhou, Yifei 292