4-2022

FAKE NEWS DETECTION ON THE WEB: A DEEP LEARNING BASED APPROACH

Piyush Vyas

DAKOTA STATE UNIVERSITY

FAKE NEWS DETECTION ON THE WEB: A DEEP LEARNING BASED APPROACH

A doctoral dissertation submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Information Systems

April 2022

By
Piyush Vyas

Dissertation Committee:

Dr. Jun Liu (Chair)
Dr. Cherie Noteboom
Dr. Shengjie Xu

DISSERTATION APPROVAL FORM

DAKOTA STATE UNIVERSITY

DISSERTATION APPROVAL FORM

This dissertation is approved as a credible and independent investigation by a candidate for the Doctor of Philosophy degree and is acceptable for meeting the dissertation requirements for this degree. Acceptance of this dissertation does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department or university.

Student Name: Piyush Vyas

Dissertation Title: Fake News Detection on the Web: A Deep Learning Based Approach

Dissertation Chair/Co-Chair: Jun Liu Date: April 22, 2022

Name: Jun Liu

Dissertation Chair/Co-Chair: Date:

Name:

Committee member: Cherie Noteboom Date: April 22, 2022

Name: Cherie Noteboom

Committee member: Shengjie Xu Date: April 22, 2022

Name: Shengjie Xu

Committee member: Date:

Name:

Committee member: Date:

Name:

Original to Office of Graduate Studies and Research
Acid-free copies with written reports to library

ACKNOWLEDGMENT

I would like to acknowledge all those eminent personalities here who have helped/supported/guided/mentored me to accomplish my goal of earning a Doctoral degree and finishing this dissertation.

I owe my deepest gratitude to my dissertation chair, Dr. Jun Liu for dedicating his time and knowledge, and for continuous support and guidance throughout this doctoral study. It is an honor to have such an understanding and inspiring guide. Also, a big thank you to my dissertation committee members – Dr. Cherie Noteboom and Dr. Shengjie Xu, for their feedback and suggestions to improve this dissertation work. I acknowledge my academic advisor and co-author Dr. Omar El-Gayar, for always advising me to be on track and helping me in my research work. I pay my sincere gratitude to my supervisors, Dr. Insu Park and Dr. Ronghua Shan for their support during my assistantship work.

Thank you to all academic and non-academic staff of DSU for providing an exceptional learning experience during the course of my Ph.D. Thank you to my family and friends for their direct/indirect support. A special thanks to my wife for her emotional support and patience.

ABSTRACT

The acceptance and popularity of social media platforms for the dispersion and proliferation of news articles have led to the spread of questionable and untrusted information (in part) due to the ease by which misleading content can be created and shared among the communities. While prior research has attempted to automatically classify news articles and tweets as credible and non-credible. This work complements such research by proposing an approach that utilizes the amalgamation of Natural Language Processing (NLP), and Deep Learning techniques such as Long Short-Term Memory (LSTM).

Moreover, in Information System's paradigm, design science research methodology (DSRM) has become the major stream that focuses on building and evaluating an artifact to solve emerging problems. Hence, DSRM can accommodate deep learning-based models with the availability of adequate datasets. Two publicly available datasets that contain labeled news articles and tweets have been used to validate the proposed model's effectiveness. This work presents two distinct experiments, and the results demonstrate that the proposed model works well for both long sequence news articles and short-sequence texts such as tweets. Finally, the findings suggest that the sentiments, tagging, linguistics, syntactic, and text embeddings are the features that have the potential to foster fake news detection through training the proposed model on various dimensionality to learn the contextual meaning of the news content.

DECLARATION

I hereby certify that this dissertation constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the dissertation describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Piyush Vyas

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background of the problem

In all vocations, including marketing, journalism, and public relations, social media has become a powerful factor of massive information exchange and communication (Zafarani et al., 2014). The reason for preferring social media is due to mobility, accessibility, and interactivity. Social media's low cost, simple access, and quick transmission of information attract a considerable audience (Shu & Liu, 2019). Thus, with an increasing reliance on social media as a major source of news, people are witnessing a speedy and rampant proliferation of misinformation and fake news.

A recent study shows that the dispersion of fake news on social media platforms propagates six times faster than the truth (Villafranca & Peters, 2019). Dissemination of misinformation on Twitter at the time of the 2016 U.S. presidential election impacted people's judgment, and 25% of the tweets related to the election were identified as fake in a period of five months before the election (Bovet & Makse, 2019). During the 2020 presidential election, 56% of American adults have cast doubt on the credibility of published information by trusted news sources, and 37% found that the news sources have reported misleading news (Shearer, 2020).

A recent survey by Pew research center indicates that 48% of American citizens have encountered misinformation regarding the COVID-19 pandemic (Jurkowitz & Mitchell, 2020). Misinformation and fake news on social media platforms should be treated as a severe emerging and growing problem that can significantly impact social and political lives as they create an obstacle for disseminating credible news and forming informed decisions. Despite the ubiquity of misinformation on social media and its swift proliferation, many Americans still have faith that they are not a part of fake news dissemination and believe that they have the cognizance to identify misleading information (Ghosh & Shah, 2018). However, existing research found that many Americans struggle to distinguish fact from fiction, with many believing false claims and even more failing to believe factual information. In a recent survey, over 50% of the respondents utilized social media as a major or minor source of news, but 75% of them could not identify fake news (Janze & Risius, 2017). In a recent study, given some fake news headlines about the COVID-19 pandemic, 40% of participants judged them as credible news, while the remaining 60% were uncertain (Kreps & Kriner, 2020). Hence, it is critical for information systems researchers to develop methods to help users distinguish misinformation and fake news from truthful ones. Such methods would help mitigate the adverse effects caused by fake news – both to benefit the public and the news ecosystem.

1.2 Statement of the problem

Fake news on social media poses a significant set of problems. First, fake news is purposefully crafted to deceive users, making it difficult to detect. Second, social media data is vast, primarily user-generated, and disruptive in nature. Third, social media users come from diverse backgrounds, have varying opinions or requirements, and utilize the platform for a variety of objectives (Shu & Liu, 2019). Therefore, fake news research often faces the problem of defining fake news definitively, and so far, there is no universal definition (Allcott & Gentzkow, 2017). However, the acceptable narrower definition would be "fake news is news articles that are intentionally and verifiably false and could mislead readers" (Allcott & Gentzkow, 2017; Shu et al., 2017). The definition portrays the two aspects of fake news; first, fake news includes false information, and second, it is created illicitly.

To alleviate the effects of fake news, researchers have been developing approaches to detect fake news. Fake news contents are often in the form of online news articles published on websites such as Reddit.com or in short texts such as Twitter feeds. The current research focuses on detecting fake news articles and tweets. Shu & Liu (2019) have defined the Fake News detection task as, "Given the social news engagements E among n users for news article a, the task of fake news detection is to predict whether the news article a is fake or not, i.e.,

$$\mathcal{F}(a) = \begin{cases} 1, & if\ a\ is\ piece\ of\ Fake\ or\ non\ credible\ News \\ 0, & Otherwise \end{cases}$$

Where $\mathcal{F}$ is the prediction function, that one wants to learn." Websites such as Snopus.com and Politifact.com (Popat et al., 2016) were developed to analyze the truthfulness of news contents, primarily relying on manual fact-checking and validation. There is also research such as (Alrubaian et al., 2018; Boididou et al., 2018; Castillo et al., 2011; Janze &

Risius, 2017; Kumar et al., 2016) that proposed automatic approaches for detecting fake news articles using various machine learning techniques such as Decision Tree (DT-rank) and Support Vector Machine (SVM). However, researchers (Alrubaian et al., 2018; Castillo et al., 2011; Kochkina et al., 2018; Qazvinian et al., 2011) found that it is highly challenging to label short sequence data (140 to 280 characters) as fake or truthful, mainly due to the semantic sparsity of short texts, thus calling for significant research that develops more effective methods for detecting fake news in short texts such as tweets.

With the development of deep learning models such as recurrent neural networks (RNNs), scientists have achieved remarkable results in natural language processing and text mining. The improved performance of these algorithms could be attributed to the following benefits provided by deep learning neural networks. First, in deep learning, word embeddings can be employed, where individual words are represented as real-valued vectors. This allows words that are used in similar ways to result in having similar vector representations, naturally capturing their similar semantics. Second, RNNs, in particular, Long Short Term Memory (LSTM) networks, take words in sequential order and learn the long-term dependencies of texts rather than local features (X. Wang et al., 2016). These benefits of deep learning techniques such as RNN, have the potential to enhance the effectiveness of text classification tasks (Ma et al., 2016). Moreover, according to Wang et al.(2016), RNNs with word embeddings provide a semantic-rich representation for individual words and can better solve the aforementioned semantic sparsity issue of short texts, which makes them a feasible method for detecting fake news in short sequence texts such as tweets. Furthermore, a recent study by Ezen-Can (2020) showed that LSTM-based models have the potential to significantly outperform other advanced models such as Bidirectional Encoder Representations from Transformers (BERT) models and take less training time.

Moreover, such deep learning techniques have hyperparameters that must be selected and optimized beforehand to attain the optimum model performance (Feurer & Hutter, 2019), and choosing the hyperparameter tuning technique for best results is not easy, especially when data comes in streams (i.e., sequence of text) (Bakhashwain & Sagheer, 2021). Traditional hyperparameter tuning techniques such as randomized and grid search cross-validation may get local optimum parameter values (values from early result's convergence). In contrast, the Genetic Algorithm (GA) can avoid this problem by providing global maxima parameter values

(Koza & Rice, 1991). Thus, this work has incorporated GA as an optimization technique with the proposed LSTM based model to utilize this advantage. Moreover, existing studies such as (Liashchynskyi & Liashchynskyi, 2019; Wicaksono & Supianto, 2018) have demonstrated that the GA can outperform other grid-based and random search parameters tuning techniques.

1.3 Objectives of the dissertation

Considering the aforementioned benefits provided by deep learning techniques in text classification, this dissertation proposed the development of a model using GA-based optimization and LSTM, a special kind of RNN, that is capable of learning long-term dependencies in textual sequences. This behavior of LSTM helps process sequential representations of both long news articles and short tweets on different newsworthy events and classify them as credible or non-credible. Therefore, the specific objectives of this work are:

1) To complement existing research by proposing an automatic fake news detection model for both long sequence textual data such as news articles and short sequence textual data such as tweets.

2) To identify and explore the potential of the textual feature set for fake news detection.

3) To evaluate the proposed approach by utilizing distinct news datasets and compare its performance with existing ML techniques and approaches in the literature.

4) To analyze the model's performance improvement during GA-based hyperparameter tuning vs traditional grid search parameter tuning.

1.4 Outline of the dissertation

The presented dissertation is structured as follows: first, the introduction section comprises a background and statement of the problem, potential solution, and objectives that showcase the motivation and necessity behind fake news detection on the web. Second, a comprehensive literature review is present in chapter 2 that discusses the existing findings and research gaps. Third, chapter 3 presents the adopted methodology, proposed approach and interleaved components of it, and various evaluation criteria. Fourth, chapter 4 explains the results and discussions. Wherein, the results of the conducted experiment are present, and potential theoretical, methodological, and practical contributions are discussed. Fifth, chapter 5 presents the concluding remarks of findings and discuss the future scope of this research in the direction of online machine learning (OL) (See Chapter 6)

CHAPTER 2

LITERATURE REVIEW

This chapter represents the existing literature studies in the area of fake news detection on social media and microblogging platforms. Initially, the chapter introduces misinformation as a problem and further shed a light upon the emergence of research interest to detect the misinformation. Afterward, the chapter unravels the existing investigations in fake news detection considering news articles on social media and news tweets on the Twitter platform. Moreover, utilized datasets, techniques, and features have been identified from existing literature and presented in this chapter. Among the relevant studies, a significant research gap has been observed and identified to pursue.

2.1 Misinformation on social media

The extensive propagation of distorted social media information has become the utter matter of consideration among academicians (Wu et al., 2019). The drift of interest towards social media (Facebook, Twitter, Reddit, PolitiFact, Instagram, etc.) arose, especially after the campaigning of the 2016 US Precedential election. Disinformation is usually an intentionally created erroneous information, unlike misinformation which can be unintended.

The type of content a user creates, and shares showcase the attributes of the utilized social media platform. Concentrating over user-centric parameters and sculpting a prototype of the content used, can be an absolute key to detecting the misinformation disseminators. In a previous study to do the direct identification of false information spreader, content extracted from the posts of a user account has been explored (Lee et al., 2021). To identify the political misinformation and its commencement, sentiment data rooted in the content has been utilized too (Bollen et al., 2010). This can be a trailblazer in identifying the association of false news dissemination with the political campaign (Ratkiewicz et al., 2011). It is concluded that the users' accounts that are centrally handled and controlled tend to incline in favor of particular political figures and campaign and thus ends up eventually supporting them (Kwon & Cha, 2014).

Figure 1. Misinformation as an umbrella term

The relevant information that hovers on social media networks consists of time and geolocations. These features with the combination of others are the vital parameters to accelerate misinformation detection. Studies such as (Kwon & Cha, 2014) reveal the usage of contextual information in form of bursty misinformation patterns for fake news detection. According to Kwon & Cha (2014), the genuine posts are posted over a lap of time, and they are widespread over a certain duration while unauthenticated posts explode frequently in a burst. As misinformation is mostly erupted by the accounts of specific groups, therefore the attached notion is that they have distinct posting models. Also, a previous study found that rumors are prevalent in bursts (Friggeri et al., 2014). According to recent research, the fake news issue remained intact in Facebook's news feed algorithm even after the modifications were made in 2018. Facebook's algorithm has unnoticed various stories evaluated as fraudulent by the foremost fact-checking firms. Moreover, two fake news-spreading websites have been actively involved and were not even barred from Facebook since starting of 2016. Therefore, recently Facebook stall the flagging that may be due to lower accuracy of flagging disputed headlines. According to the social media reviewers, the steps taken to alleviate the misinformation via fact-checking are "not working" and due to its reachability, on the whole, it is "becoming unstoppable" (Allcott et al., 2019). As shown in Figure 1 (inspired by (Wu et al., 2019)), according to Wu et al., (2019) misinformation should be treated as an umbrella term because it can be defined in multiple contexts such as fake news, hoaxes, rumors, and spams. Wherein, fake news implies fabricated information on real-life events that circulates inadvertently, rumor implies the information that has the probability of either being accurate or forged, and lastly, the unverified information forwarded to a bulk of recipients together describes the spam. This research focuses on the dispersion of fake news on the web; therefore, the following literature review highlights the findings and gaps from the relevant studies in the context of fake news detection on social media platforms in the form of news articles and tweets. Following Table 1 shows the existing literature that considered the misinformation as a generalized problem and applied various ML/DL techniques to detect misinformation on distinct social media platforms.

Table 1. Existing literature with a focus on misinformation detection

Authors	**Focus**	**Data Sources**	**Techniques**
(Niknam et al., 2020)	Misinformation Detection	Instagram	Customized input-output quantitative (IPO)
(Abul-Fottouh et al., 2020)		YouTube	Latent Order Logistic Model (LOLOG)
(Ghenai & Mejova, 2018)		Twitter	Logistic Regression (LogReg)
(Ghenai, 2017)		Twitter	Statistical analysis/ Tools – botornot & metamap
(Hou et al., 2019)		YouTube	Support Vector Classifier (SVC)
(Waszak et al., 2018)		Twitter, Facebook, Pinterest, LinkedIn	Buzzsumo
(Smaldone et al., 2020)		Facebook, Twitter, Instagram, YouTube, Pinterest	Structure Equation Modeling (SEM)
(Sear et al., 2020)		Facebook	Latent Dirichlet allocation (LDA)
(Porat et al., 2019)		Twitter	Pearson correlation
(Kouzy et al., 2020)		Twitter	Chi-square statistic

2.2 Fake news article credibility assessment

Most of the existing research related to assessing the credibility of fake news has utilized news articles from social media platforms for their analysis (e.g., Agrawal et al., 2019; Bovet & Makse, 2019; Grinberg et al., 2019; Ma et al., 2016; Popat et al., 2016, 2018a). Social media platforms work as a mechanism to establish the interaction between individuals, and users' actions influence each other's opinions (Candogan & Drakopoulos, 2020). This phenomenon works well in the case of fake news article dissemination. The fake news articles are usually shared in a sequential manner irrespective of any illicit intentions because, according to Papanastasiou (2020), users are inclined to share the news article that is shared by their peers, which leads to the proliferation of viral fake news content.

In the study conducted by Dennis (2019), they assert that users' level of belief affects by encouraging them to think about who produced the article; thus, believability is crucial irrespective of the news source's credibility; therefore, users are more inclined to trust news articles that matched with their belief rather than highlighting the source name. The warning attached to the news headlines and people's comments on news articles plays an important role in sharing further or disseminating fake news. Pennycook et al. (2020) have found that the accuracy of fake news article's headlines is often spared by getting tagged as fake and has attracted the users to share more. Similarly, users are also inclined to align their replies on news items with the majority's sentiments for an article's legitimacy which can lead to the dispersion of fake news (Wijenayake et al., 2021). Understandably, at the same time, the proliferation can be stall by showcasing the majority of critical user comments for probable fake news (Wijenayake et al., 2021).

Websites such as Politifact.com and Snopes.com rely on investigating journalists and groups of experts, distinguishing unreliable news articles from reliable ones. Ma et al.(2016) and Popat et al. (2016, 2018a) have questioned the human intervention for bifurcation of news articles as fake and real and urged to assess the credibility of expert's judgments. The study by Mukherjee and Weikum (2015) shows that renowned news communities like Reddit.com, Dig.com, and Newstrust.net gave privileges to users for rating and reviewing news articles. Moreover, they highlighted the necessity for joint assessment of credibility and trustworthiness of users, articles, and news sources because they believed that user sentiments and popularity also affect the dissemination of news.

Manual work of feature extraction and news content encoding/labeling is sometimes impossible or leads to a cumbersome and complex approach (Ebadi et al., 2021). Thereby causes a necessity for automatizing and featureless methods. Since fake news detection has been evolved as a potentially strong research area and urges the implementation of various detection techniques (Shrivastava et al., 2020), RNN based deep learning models are recommended because of their ability to seize the sequential information of news content (Ebadi et al., 2021). The manual assessment of fake news is not scalable because of the speed of misinformation dispersion on social media platforms and the sheer volume of such content. Hence, researchers used machine learning and deep learning techniques to automatically detect fake news articles (Janze & Risius, 2017; Kumar et al., 2016; Popat et al., 2016, 2018a).

As an example, Janze and Risius (2017) used convolution neural network (CNN), SVM, and DT-rank to differentiate true and false news postings by mainstream media pages on Facebook through cognitive (message and comments), visual (images), affective (various emojis) and behavioral cues (sharing and tagging) of the news posts. Popat et al. (2016) implemented a classification model for the credibility analysis of news claims through various features such as the language of articles, sources of articles, subjectivity, and implicative verbs. Also, they used Amazon Mechanical Turk to validate their approach. Another study by Popat et al. (2019) presents an end-to-end neural network model incorporating bidirectional LSTM to take advantage of generated past and new features to assess news articles' truthfulness. The authors extracted features such as the language style of articles, stance towards a claim, and trustworthiness of the sources. Kumar et al.(2016) used machine learning techniques including logistic regression, SVM, and random forest to detect false information on Wikipedia. They focused on different article structures and content characteristics such as text length, markup ratio, and link density.

2.3 Fake news tweets credibility assessment

Microblog has become a widespread news reporting and dissemination medium. Fake news propagating on microblogs, on the other hand, would significantly undermine its public confidence (Jin et al., 2017). Twitter has emerged to share news in limited words and has become a prevalent broadcasting medium. Taking into account the nature of microblogs, the formal definition for a news event would be, “a news event is made up of a series of tweets that contain specific keywords over a period of time” (Jin et al., 2017). The contents of fake news may have several domains, and they may develop and mutate throughout online distribution. Users are more inclined to forward the news that deviates significantly from common sense, is more contentious, or more sensational than routine news. Thus, information like the number of retweets, comments, and likes is vital (Li et al., 2021). When identifying fake news, common information across news domains and semantic connections between real (fake) news are critical (Yuan et al., 2021). However, to excel the fake news detection, Liao et al. (2021) have argued that news propagator info is helpful because news on microblogs or in short textual content would serve limited information to attain apt representations (i.e., different content-based features) during the detection task. On a positive note, automated fake news detection helps detect information that is more likely to be fraudulent (Reis et al., 2019). Li et al. (2021) have suggested that the news text, contextual information, dissemination information are the primary characteristics that can be used to detect fake news.

With the evolution of Artificial intelligence, ML and DL techniques have been applied to detect fake news with remarkable success (Li et al., 2021). Researchers such as (Alrubaian et al., 2018; Boididou et al., 2018; Castillo et al., 2011; Kochkina et al., 2018) have developed approaches for detecting fake news tweets using machine and deep learning techniques. Boididou et al. (2018) used machine learning techniques, including logistic regression (LogReg) and Random forests (RF), to detect misleading information on Twitter through various extracted features such as tweet length, account age, number of followers, number of tweets, number of hashtags and retweets, and so on. Qazvinian et al. (2011) focused on identifying tweets that endorsed the rumors using Bayes classification based on content and network-based features such as lexical patterns, part-of-speech, retweet, hashtags, and unified resource locators (URLs). Castillo (2011) used decision trees to automatically classify the trending news tweets and validated their approach using 3-fold cross-validation. They considered various features such as the number of followees, followers, retweets, URLs, and hashtags, along with manual labeling of data through human assessors. Similarly, Alrubaian et al. (2018) used a combination of random forest, naïve Bayes, and decision trees to detect tweets containing malicious information. Kochkina et al. (2018) considered the issue of fake news tweets, consisting of four subproblems, including rumor detection, rumor tracking, stance classification, and rumor verification. They used the branch LSTM technique to solve the problems. Agrawal et al. (2019) developed a fake news detection method using logistic regression and studied how such a method fares when applied to the news being shared on Twitter in a period of several months. Clues encircled around the fake news content and dissemination sequence (i.e., the pattern of spread) can be identified by utilizing attention-based neural networks, such as Ni et al. (2021), have applied Multi-View Attention Networks (MVAN) to detect fake news on Twitter wherein the model extracts the clue words for news tweets sources and dissemination structure. Table 2 highlights the focus and adopted technique category of existing literatures.

Table 2. Existing study's focus and techniques

Authors	Focus	Data source	Techniques
(Ali et al., 2021)	Stance Detection	Snopes.com	Deep learning-memory neural network (MemNN) model
(Liao et al., 2021)	Fake News Detection	LIAR	Multi-task learning (FDML) model
(Huu Do et al., 2021)	Identification of Fake News	Weibo, and PHEME	Generic Deep Markov Random Fields Neural Network (GDMFN) Model
(Qi et al., 2019)	Identify Fake News	Weibo	Multi-domain Visual Neural Network (MVNN)
(Bhutani et al., 2019)	Fake News Detection	PolitiFact & LIAR	Naive Bayes (NB) and Random Forest (RF)
(Yuan et al., 2021)	Identify Fake News	Twitter and Weibo	Domain-adversarial and graph-attention neural network (DAGA-NN) model
(Ni et al., 2021)	Fake News Detection	Twitter	Multi-View Attention Networks (MVAN)
(Jin et al., 2017)	News Verification	Weibo	Support vector machine (SVM), Logistic Regression (LogReg), and Random Forest
(Ajao et al., 2019)	Fake News Detection	PHEME Twitter	ML stack: LogReg, SVM, Decision Trees (DT), RF, and extreme gradient boosting (XG-Boost)
(Reis et al., 2019)	Fake News Detection	BuzzFeed	ML Stack (k-Nearest Neighbors (KNN), NB, RF, SVM, and XGB.
(Li et al., 2021)	Fake News Detection	Twitter & Weibo	Unsupervised autoencoder
(Janze & Risius, 2017)	Identify Fake News	Facebook & BuzzFeed	ML Stack: LR, SVM, DT, RF, and XGB
(Alrubaian et al., 2018)	Credibility Analysis of Fake News	Twitter	RF, NB, DT, and feature-rank naïve Bayes (FR_NB)

(Castillo et al., 2011)	Credibility Analysis of Fake News	Twitter	SVM, DT, NB.
(Ma et al., 2016)	Identifying Rumors.	Twitter and Weibo	Recurrent neural networks (RNN)
(Ma et al., 2015)	Identify Rumors	Twitter and Weibo	Dynamic Series-Time Structure (DSTS)—based SVM model
(Qazvinian et al., 2011)	Identify misinformation	Twitter	Bayes Classifiers
(Grinberg et al., 2019)	Fake News Detection	Twitter	The harmonic algorithm presented by (De Alfaro et al., 2015)

2.4 Research gap

Similar to the existing literature, this research is intended to develop a model for the automatic detection of fake news articles and tweets. The proposed model is designed to detect fake news in both long-sequence and short-sequence texts. Similar to the researchers such as Ma et al. (2016) and Popat et al. (2018a, 2019), this work has used deep learning, more especially an LSTM neural network, to leverage its advantages in learning continuous representations of textual data. Complementing the aforementioned approaches, this research has used various textual-based (i.e., word count, text length), Sentiments (Positive and Negative), tagging (i.e., # hash tags, @ mentions) based, and Syntactic features (i.e., Ngrams). The presented approach focused on the textual information presented in news tweets and articles. The proposed model automatically extracts all mentioned features to train the LSTM based model to understand the different contextual meanings of a sentence (e.g., the word Bank refers to the place where someone keep money or a riverside). The inclusion of word embedding maps each word with different associated meanings by creating distinct vectors. Thus, not rely on the extraction of handcrafted features (i.e., manually extracted) as used by Boidiou et al. (2018) and Castillo et al. (2011). Moreover, the presented approach may detect the proliferation of Fake news at its early stages. Furthermore, the approach can work proficiently on any news-related textual content, either long articles or short tweets.

CHAPTER 3

RESEARCH METHODOLOGY

This chapter first, briefly presents the rationale for adopting design science research methodology (DSRM) to design the proposed artifact which is an LSTM based fake news detection model. Moreover, a detailed description of the artifact's interleaved components extracted features from the datasets, and evaluation criteria are presented.

According to (Hevner, 2004), majorly the behavioral and design science, both paradigms recline under the domain of information systems (IS). Wherein, behavioral science focuses on theory building and testing and design science emphasize artifact design and evaluation. This research has accommodated the design science research methodology (DSRM) provided by (Hevner, 2004) to build the proposed approach as an artifact. DSRM comprises components such as artifact design, research relevance, design evaluation, contribution, rigor, and communication of research. Prior studies such as Liu et al. (2020) have demonstrated the utilization of DSRM to build deep learning-based artifacts. Thereby DSRM suitably guides the presented approach to creating an efficient artifact. The research relevance and rigor have been established in the prior sections, such as the introduction and literature review of this work. Herein, a detailed illustration of the proposed artifact design is presented. In DSRM, the design of the artifact can be seen as a search process involving an iterative evaluation and refinement of the artifact. Thus, the proposed artifact is an amalgamation of NLP and LSTM-based models to classify the credible and non-credible news content. The artifact evaluation and contribution have been discussed in the subsequent sections. The implementation of the artifact was done by utilizing evolving DL, NLP, and GA-based techniques to enhance the generalizability of the solution. The efficiency demonstration of the proposed model has been done by using two distinct datasets for the assessment of the model on long and short text news. Former publications based on the proposed study including this dissertation may communicate the implications, findings, and contributions to the IS research knowledge body. With this, the presented work assures the utilization and meeting of the guidelines provided by Hevener's (2004) DSRM.

3.1 Proposed approach

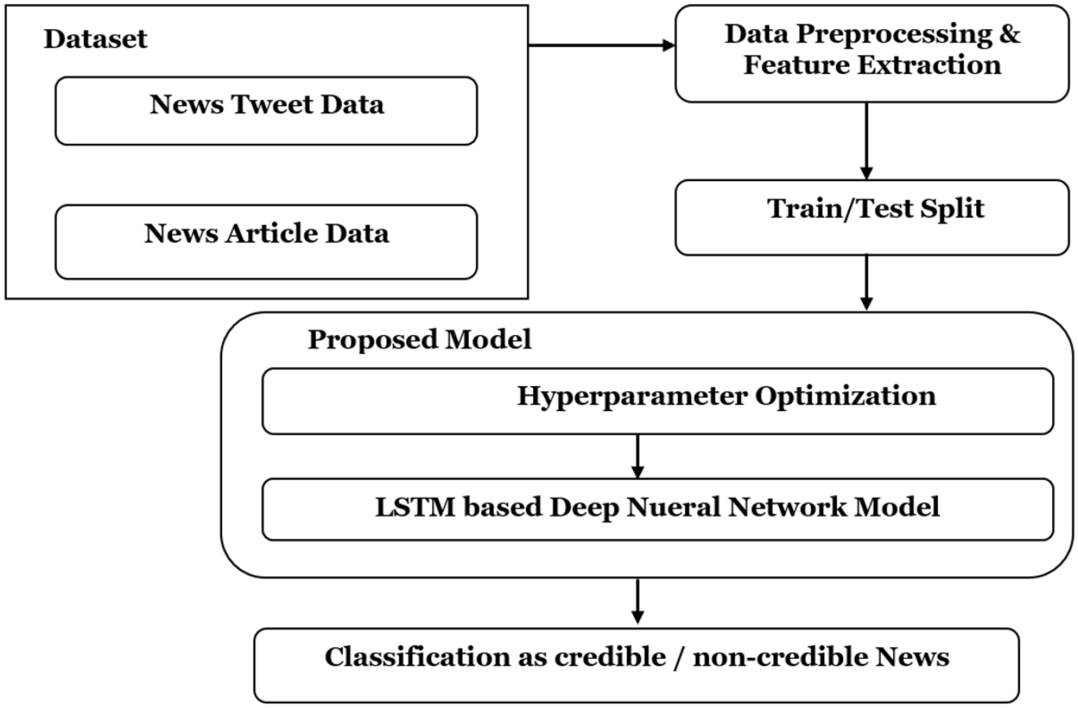


Figure 2. High-level representation of the proposed approach

Figure 2 shows the high-level representation of the proposed classification model for classifying credible and not-credible news content. To assess the effectiveness of the model on long and short textual posts, two different datasets are used. The first dataset consists of many online news articles, and the second includes tweets related to real-world news events. First, text data preprocessing is performed by utilizing python's natural language toolkit (NLTK) library to clean and filter out irregularities and anomalies in the datasets. Then the data was split into training and test datasets.

Sklearn-deap and Keras - open-source python-based libraries, have been used to implement GA and LSTM based model for text classification. Finally, the performance of the model has been evaluated based on commonly used measures such as accuracy, precision, recall, and F1 scores. The following sections describe the methodology in detail.

3.1.1 Datasets

For any classification task, there is a necessity for labeled datasets from authorized sources. When it comes to news content classification, there is a paucity of standard benchmark datasets (Ghosh & Shah, 2018). Two particularly relevant datasets have been identified and used in this research. The first dataset contains relatively long news articles, and the second one developed by PHEME (Zubiaga et al., 2016) includes short news tweets. These two datasets are used to evaluate the effectiveness of proposed model for classifying news contents represented in long-sequence and short-sequence texts. First, News Article Dataset was developed by Ahmed et al. (2018). This dataset includes 12,600 fake news articles published on Kaggle.com and 12,600 truthful news articles from Reuters.com. The fake news articles were originally identified by Politifact.com, a fact-checking website that rates the accuracy of claims by governmental leaders and politicians. Articles in both the fake and truthful categories occurred in the same timeline, specifically in 2016.

(a)

(b)

Figure 3. (a) The non-credible tweet from a leading news agency; (b) The tweet showcasing actual news

Binary numbers are used to label the articles. 1 is assigned to the fake news articles and 0 to the truthful news articles, as the focus is on detecting fake news. Second, News Tweet

Dataset provided by PHEME consists of tweets related to real-world news events including Ferguson protests, shooting in Ottawa, the Boston attack, the hostage situation in Sydney, and so on (Zubiaga et al., 2016). In this dataset, the tweets were labeled as “rumor” vs. “non-rumor”. There are 1,972 tweets labeled as rumors and 2,830 labeled as non-rumors. This data is publicly available in JavaScript Object Notation (JSON) format with several directories, each of which includes several tweets. All tweets are collected in a single file, and again 1 is used to represent rumors and 0 to represent non-rumors. From the dataset as an example, figure 3a shows the fake news tweet snippet, wherein a Boston-based news agency tweeted the fake news that proliferated misleading information that connects the JFK library blast with the chain of ongoing Boston marathon terrorist attacks. The tweet in figure 3b cast out this fake news by mentioning the reason behind the JFK library blast by citing a different tweet by one of the leading news agencies in the Boston area.

3.1.2 Data preprocessing & feature extraction

For textual data, preprocessing is essential before supplying the text to the model learning process, as textual data usually contain redundancies and irregularities. Natural language processing techniques (NLP) have been used to preprocess the textual data and prepare them for further analysis. The stop words, special characters such as, ‘!’, ‘&’, ‘$’, symbols such as emojis, repetitive period signs (e.g., .. or …), white spaces, line breaks, blank rows, and extra variables such as tweet ids and user ids, have been removed. The removed stop words include articles (a, an, the, etc.), pronouns (me, you, etc.), and prepositions (in, on, to, etc.) that have importance in English grammar for communication but do not have semantic importance in the learning process of the model. Then stemming is performed using the Porter Stemmer developed by Martin Porter (1980). Stemming reduces infected or derived words to their word stems and helps to increase the performance and reduce the size of data (Porter, 1980). All the words in the datasets have been lowercased because the same word in different cases would be treated differently during the encoding process. Textual features have been extracted that are used in the model building. Features include N-grams (i.e., a chain/set of co-occurring words in tweets) such as unigrams, bigrams, and trigrams, trending hashtags (Ex., #pray4boston), and trending entity mention (Ex., @BarackObama). Additionally, the number of characters and number of words were also included as features for modeling building. Following are the details of extracted features:

3.1.2.1 Linguistic feature extraction

Part of speeches (POS) helps to determine the linguistics of the text data by assigning verbs, adverbs, adjectives, and nouns tags to each word in the news content. Such POS is helpful to understand the contextual meaning of the news sentences because often, the same word might have two entirely distinct meanings (Kouloumpis et al., 2011). For example, the phrases like “who was involved behind the Boston attacks” and “understanding reason behind Boston casualty is an involved matter” both have distinct contextual meanings; hence it is crucial to define the meaning of “involved” to determine whether or not these phrases are meaningfully related. Therefore, the POS has been extracted for news articles and tweets by utilizing Natural Language Toolkit (NLTK) (Bird et al., 2009) python library. Table 3 shows the sample extracted POS wherein JJ stands for adjectives, RB for an adverb, NN for a singular noun, NNS for a plural noun, VB for the verb.

Table 3. Sample of Extracted POS

POS tags	**Sample Tweet**
(‘media’, ‘NN’) (‘rush’, ‘NN’) (‘truths’, ‘NN’) (‘are’, ‘VB’), (‘missed’, ‘VB’) (‘mistakes’, ‘NN’), (‘are’, ‘VB’), (‘made’, ‘VB’) (‘pain’, ‘NN’), (‘is’, ‘VB’), (‘increased’, ‘VB’) (‘reality’, ‘NN’) (‘rushed’, ‘VB’) (‘legitimate’, ‘JJ’) (‘answers’, ‘NN’) (‘boston’, ‘NN’)	“In the media rush, truths are missed, mistakes are made, pain is increased. Reality can't be rushed to legitimate answers. #Boston”

3.1.2.2 Syntactic feature extraction

Primarily, this work utilizes news written in the English language; hence, unraveling the syntactic or syntax-related structure is crucial. N-grams are helpful to extract the relationship between words because, according to Weaver (1955), the word relationships in a sentence are complex, but it is possible to capture all relevant information by analyzing the order of words or which words tend to come together. Such co-occurring words in the corpora are known as N-grams, wherein N represents the number of words that co-occurred. Herein, Unigram shows the one word, Bigram offers the combination of two words, Trigram shows the three words, and so forth.

Table 4. Top five N-Grams

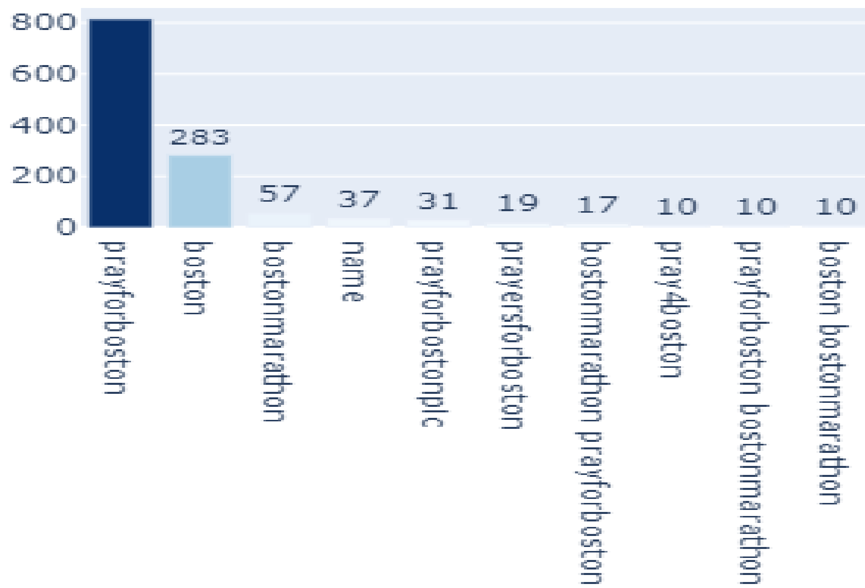
Unigram	**Bigram**	**Trigram**
Marathon	Year old	Year old boy
Year	Baptist church	Sandy hook kids
Died	Sandy hook	Old boy killed
Today	Died boston	Explosion jfk library
killed	Boy died	Explosive devices boston

Table 4 shows the top 5 unigrams, bigrams, and trigrams from extracted N-grams among the used datasets. Overall, most news content possesses information about the killing, disgust, Boston, explosion, and death-related news.

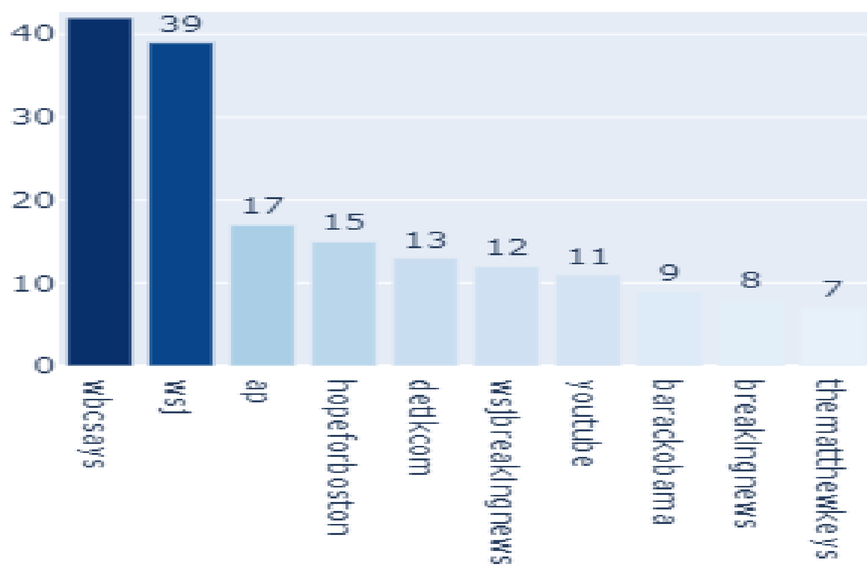
3.1.2.3 Tagging features extraction

As the tagging features, hashtags and user mentions are used. Hashtags are created by combining # character and topic of the information (e.g., #bostonkilling). Such hashtags can be made by users the moment they share any social media post or tweets, and these hashtags lead to all relevant tweets or posts that may have used the same news topic. Generally, trending news is the result of hashtag usage in an ample amount of news articles or tweets (Giachanou & Crestani, 2016). User mentions in the news articles and tweets address the other user's social media handle or account name. Such reference to users can be created by using @ special symbol and the specific account name (e.g., @barakobama). These can be cited anywhere in the news text body (Giachanou & Crestani, 2016).

Figures 4a and 4b show the top trending hashtags among tweets and the most frequent shout-outs (i.e., @entity mentions). #pray4boston is the most trending hashtag presented in news tweet dataset, and @wbcsays is the most used shout-out entity.



(a)



(b)

Figure 4. (a) Top trending hashtags; (b) Top trending user mentions

3.1.2.4 Sentiments Extraction

Sentiment extraction is sometimes known as polarity-based analysis because polarity denotes positive, neutral, and negative sentiments. The Valance-Aware Dictionary for Sentiment Reasoning (VADER) (Hutto & Gilbert, 2014) widely adopted sentiment extractor. Because of its capacity to assess the sensitivity of textual information, VADER is a rule-based model mainly created for sentiment extraction of social media posts (Shihab Elbagir and Jing

Yang, 2019). Vader has outperformed similar techniques such as Linguistic Inquiry and Word Count (LIWC) and the General Inquirer (Hutto & Gilbert, 2014). Hence, the VADER has been employed for the extraction of sentiment. VADER calculated the polarity scores of positive, neutral, and negative sentiments, and the compound score was used as a threshold to divide news articles and tweets across three sentiment classes. Table 5 shows the sample of extracted positive, neutral, and negative sentiments tweets.

Table 5. Extracted sample sentiments of news tweets

Sentiments	**Example tweets**	**Scores**
Positive	Proudly wearing my Boston Celtics shirt to sleep tonight.	{'neg': 0.0, 'neu': 0.552, 'pos': 0.448, 'compound': 0.7579}
Negative	The 8-year-old boy who was killed by Boston Marathon blast was waiting to greet his runner father.	{'neg': 0.206, 'neu': 0.688, 'pos': 0.106, 'compound': -0.4939}
Neutral	#PrayForBoston Everyone wear purple tomorrow for Boston! Retweet to get the word around.	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}

3.1.2.5 Content-based feature extractions:

Figure 7a (Appendix B) shows the word cloud of the most frequently used words among news tweets in the corpus. Figures 7b and 7c (Appendix B) show the distribution of the number of words, and the distribution of the number of characters in each tweet respectively. Most of the tweets have a number of characters between 80 -140 and a number of words between 10 - 25.

Next, each dataset was split into two datasets: training and test. For the news tweets dataset, 70% of the examples were selected as the training dataset, and the test dataset includes the rest, 30%. Similarly, for the News article dataset, 65% of the examples as the training dataset was randomly selected, and the test dataset includes the rest 35%. The test datasets remained untouched in the model fitting process and were later used to estimate the prediction performance of the proposed model when applied to unseen news contents.

3.1.3 Proposed LSTM model

3.1.3.1 Hyperparameter optimization

Deep learning methods often contain numerous variables, known as hyperparameters, that should be chosen ahead of time before training the models (Feurer & Hutter, 2019). Hyperparameters are frequently fine-tuned manually or via automatic algorithms. The automatic selection of appropriate hyperparameters is critical for achieving optimum and quicker outcomes, especially when working with big datasets. Understandably, an essential aspect of choosing the best performing model. Alternatively, if the model's accuracy is high, it aids in making correct judgments; conversely, when accuracy is poor, it hinders accurate decision-making. Thereby, it is required to adopt appropriate hyperparameter tuning techniques, especially when the data is in sequential nature like text sentences (Bakhashwain & Sagheer, 2021).

To optimize the hyperparameters of the ML/DL models, there is a variety of techniques are available such as randomized and grid-based k-fold cross-validation, evolutionary algorithms such as particle swarm optimization, and genetic algorithm. It has been identified that many of such techniques have the problem of converging in the local minima of the solution space (i.e., iteratively, providing the lower end or minimum values of the hyperparameters) (Ritchie et al., 2003). GA has the potential to stall the problem of local minima and converge at the global maxima (i.e., providing the maximum hyperparameter values) (Koza & Rice, 1991). Thus GA has advantages over traditional hyperparameter tuning techniques such as escaping the local minima/optima and handling large hyperparameter values and complex problems (Bakhashwain & Sagheer, 2021). Moreover, existing studies (Liashchynskyi & Liashchynskyi, 2019; Wicaksono & Supianto, 2018) have demonstrated that the GA has the potential to outperform other grid-based and random search parameter tuning techniques. Therefore, the GA was incorporated as a part of the proposed approach.

Figure 5 represents the detailed view of the GA components and their work for optimizing the hyperparameters. Holland (1992) invented the Genetic Algorithm (GA) in 1970. This combines Darwin's evolutionary theory with the concept of human reproduction. GA is a stochastic search method based on the natural selection process and is the crux of evolution. Many optimum solution searching, and parameter optimization issues have been solved with GA. Iteratively, GA creates new populations of chromosomes from old ones.

In the context of hyperparameter tuning, a chromosome can be the parameter value that defines a potential solution to the problem (i.e., optimum values of hyperparameters) targeted by the GA to solve. Every chromosome represents a binary encoded (in the form of 0 and 1 bits) candidate solution. Every chromosome is assigned a fitness measure via a fitness function (i.e., a mathematical function to generate desired fitness score), indicating its adequacy for the task. To calculate an entire generation of new chromosomes, standard GA applies genetic operations such as crossover and mutation to an initially randomly selected population. Randomly picked chromosomes are then crossed over based on their fitness value, i.e., which chromosome has the highest fitness value engaging in crossover to generate a new generation of strong solutions. Similarly, the mutation will be performed to change the bits arrangements of the chromosomes to produce new unique and robust generations. GA comes to a halt when the maximum number of generations has occurred, or the number of mutation-crossovers has been finished, or if the desired fitness values are reiterating (Vyas & Dubey, 2013).

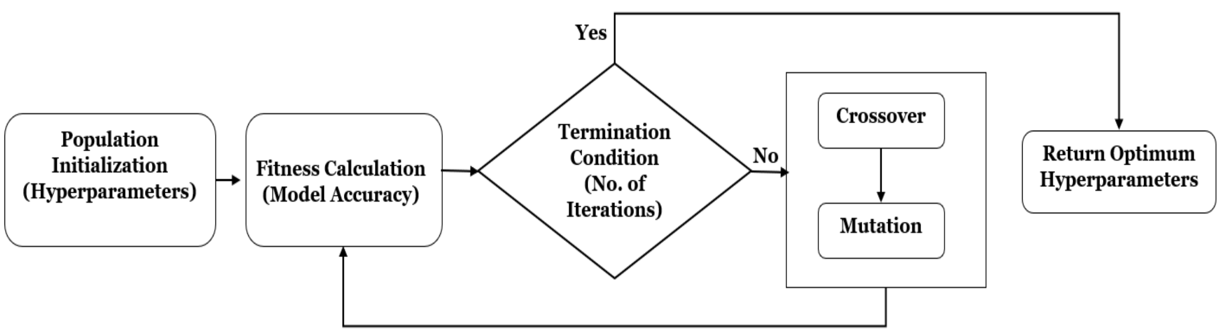


Figure 5. Detailed view of GA process

3.1.3.2 Word embedding

According to Brownlee (2017), word embedding is a class of approaches for representing words and documents using a dense vector representation. It is an improvement over the traditional bag-of-word (BOW) model encoding schemes, where large sparse vectors were used to represent each word or to score each word within a vector to represent an entire vocabulary. In an embedding, words are represented by dense vectors where a vector represents the projection of the word into a continuous vector space. The position of a word within the vector space is learned from the text and is based on the words that surround the word when it is used. The position of a word in the learned vector space is referred to as its embedding. An embedding layer offered by Keras was used to perform word embeddings for neural networks on text data. It requires that the input data be integer encoded so that each word is represented by a unique integer. The embedding layer was first initialized with random weights and learned an embedding for all words in training data sets.

3.1.3.3 Long short-term memory (LSTM)

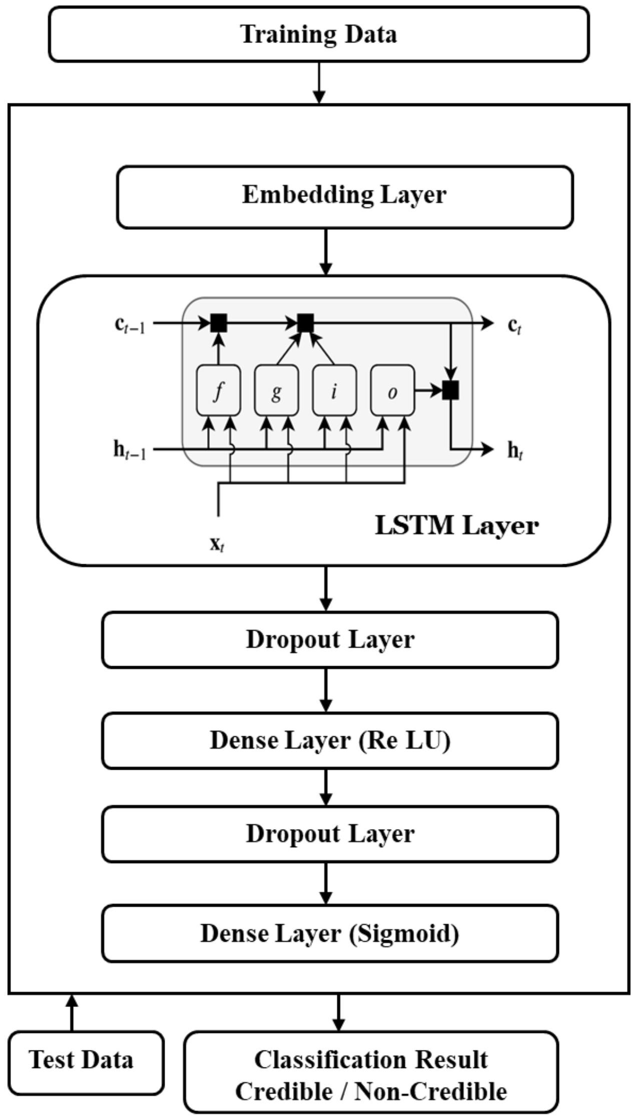


Figure 6. Detailed view of LSTM based approach

Figure 6 represents the detailed view of proposed LSTM model. Keras, an open-source deep-learning python-based library was used to implement the model. LSTM is a specific recurrent neural network (RNN) architecture. RNNs are efficient in the processing of sequential data and have been widely used for speech and text recognition (Heinrich et al., 2019), but they have limitations in learning long-term dependencies. LSTM has been introduced to overcome the shortcoming of the standard RNNs and gain optimum performance (C. Zhou et al., 2015). LSTM utilizes gradient-based optimization for learning long-term contextual dependencies, which has the potential to outperform standard RNNs for textual data of different lengths (Karmiani et al., 2019; Ma et al., 2016). LSTM is used as the centerpiece of the model for classifying news articles and tweets into credible (represented as 0 in the datasets) vs. not credible (represented as 1). As shown in Figure 6, the LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. These blocks can be thought of as a differentiable version of the memory chips in a digital computer. Each one contains one or more recurrently connected memory cells and three multiplicative units – the input, output, and forget gates – that provide continuous analogs of write, read and reset operations for the cells (Graves & Schmidhuber, 2005). The key to LSTM is the cell state, represented as the horizontal line running through the top of the LSTM layer in Figure 6. The cell state runs straight down the entire chain from c_{t-1} (old memory) to c_t (new memory). The LSTM can remove or add information to the cell state, carefully regulated by structures called gates. In Figure 6, *i* and *g* represent the input gates, *o* represents the output gate, and *f* represents the forget gate. For example, if a fake news tweet such as “former president Barak Obama was not born in the United States” goes into LSTM, the input gate holds this information in the memory. Next, suppose some similar fake news tweets such as “Barak Obama who was the president of united states was not born in the USA” are input to the LSTM, since both tweets have the same contextual meaning, the forget gate will eliminate unnecessary information such as “who was”, and the output gate will generate a new combined sequence “Barak Obama was not born in the USA” and keep this information as fake news for a long time in the memory. The whole process is known as memorization. The included additional memory units in LSTM including input, output and forget gates make LSTM different than the standard RNNs. Hence, LSTM is capable of keeping the memory long term (Hochreiter & Schmidhuber, 1997). After the word embedding and LSTM training, a couple of dropout and dense layers is used. Dropout layers prevent model overfitting by dropping some units (i.e., neurons). The dense layer is a fully connected layer used for outputting the predictions. As the existing research such as Altche et al. (2017) recommends using two dense layers rather than one; therefore, multiple dense layers were used. As activation functions, the Rectified Linear Units (ReLU) and the logistic Sigmoid function were used for dense layers. The logistic function was used to provide binary outputs representing credible vs. not credible.

3.1.4 Evaluation

The proposed model was applied to the test datasets and evaluated the classification performance of the model using the classical precision, recall, F1 score, and accuracy metrics. The formulas for computing recall, precision, F1 score, and accuracy are shown above. Recall refers to the rate of correctly classified positives among all positives and is equal to TP divided by the sum of TP and FN. Precision refers to the rate of correctly classified positives among all examples classified as positive and is equal to the ratio of TP to the sum of TP and FP. F1 score represents the harmonic mean of recall and precision. Accuracy represents the percentage of correct predictions. Where TP represents the number of True Positives, i.e., positive samples that were correctly classified. TN is the number of True Negatives, i.e., negative samples that were correctly classified, FP the number of False Positive, i.e., negative samples that were incorrectly classified as positive, and FN the number of False Negatives, i.e., positive samples incorrectly classified as negatives. These numbers were then used to calculate classification measures such as precision, recall, F1 score, and accuracy.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1 score} = \frac{2(\text{precision x recall})}{\text{precision} + \text{recall}} \qquad \text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

CHAPTER 4

RESULTS AND DISCUSSION

Table 6. Hyperparameter description of models (Buitinck et al., 2013; Pedregosa et al., 2011)

Model	Tuned Parameters	Description
Decision Tree (DT)	Criterion = "entropy", max_depth = 8	Criterion is a predefined function that processes the given value to produce gini index or entropies of information gain. Moreover, a suitable integer type value can be supplied in the max_depth variable to halt the model overfitting.
Gaussian Naïve Bayes	var_smoothing = 0.4	To attain high model performance and to normalize the datapoint variance, a float type value is used to pass in var_smoothing parameter.
Logistic Regression	C = 0.01	C is the strong parameter and required to be tuned to halt the model overfitting; thus, it is the regularization parameter that takes positive float values. The smaller value is the strong value.
Random Forest	n_estimators = 100, Criterion = "gini", max_depth = 8	Max_depth and criterion work like CART-DT parameters. Additionally, to select the no. of trees in the random forest, a n_estimator parameter can be used.
SVC	C=0.01, kernel=rbf	The parameter C has the same significance as in LR, whereas kernel parameter helps to deal with Non-linear data or high dimensional data during training, and possible values can be a mathematical function such as linear', 'poly', 'rbf', 'sigmoid', 'precomputed'.
LSTM	Batch_size = 32-128, Dropout_rate = 0.1-0.5, No. of Epochs = 10-100	Neural networks (NN) have various hyperparameters to tune; hence Batch_size, droput_rate, and num_of_epochs are used. Wherein, to provide the number of samples for NN, a batch size integer type parameter was used. To stop overfitting, dropout_rate was used which takes float values between 0 and 1. The no_of_ephochs represent the number of iterations an NN-based model can run to get the best results.

Every machine learning technique has a critical component known as hyperparameters that are often responsible for profound performance changes. Table 6 shows the hyperparameters for the implemented various ML and DL-based techniques on collected datasets. These models include Support Vector Classifier (SVC), Logistic Regression (LogReg), Classification and Regression decision Trees (CART DT), Gaussian Naïve Bayes (GBN), Random Forest (RF), and bidirectional LSTM. Moreover, two experiments were performed to assess and compare the performance of such models after optimizing the hyperparameters via traditional grid search and GA methods. All the data and features have been used during both experiments. As an experiment setup, the python-based opensource libraries, jupyter notebook environment of google colab (i.e., cloud-based platform for performing highly computational tasks such as significant news text classification), and NVIDIA Tesla K80 one graphics processing unit (GPU) were used.

4.1 Experiment 1 – fake news classification with grid search optimization

In the realm of Machine learning and deep learning, cross-validation is a powerful concept that can be utilized to adjust the hyperparameters of ML techniques and DL neural networks. Mostly, the K-fold cross-validation method has been adopted in the practices wherein K demonstrates the number of folds in the supplied dataset. The provided dataset can be the whole part of the data or the training part, thus depending on the data size, the number of folds would be selected i.e., 5-fold or 10-fold. Such folds denote the split of datasets into distinct sections; for example, 10-fold denotes the partition of a dataset into ten parts. All parts are used simultaneously as a training and testing set during the K-fold cross-validation. Finally, as result, the optimal set of parameter values, as well as the accuracy of the model's performance were generated. After examining the performance of various hyperparameter settings, the appropriate values can be chosen and used while training the final ML model.

In practice, tracing the appropriate hyperparameter values for a specific dataset has usually been done manually. To train ML/DL models with apt hyperparameter values, researchers rely on their previous experience of performing similar tasks. This could be led to an issue of not getting optimum results because the hyperparameter values for one problem may not be the best option for another one, as such values can differ throughout the datasets. As a result, defining hyperparameter values based on prior experience is challenging. Hence, Grid Search (GS) is the hyperparameter tuning technique that can be used as an automated guided method to get the best set of hyperparameters for ML/DL models. GS performs the Cartesian product-based mathematical operation to produce the possible blend of hyperparameter values.

During GS the ML/DL models usually train on all hyperparameter choices. To assess the training, a performance metric was used, which is commonly utilized in the training set incorporating the cross-validation (CV) approach. This method of validation guarantees that the trained model can perform the task efficiently. In GS, first, a grid (usually a dictionary) is used to create with every conceivable combo of all hyperparameter values supplied, Second, the model's accuracy score is calculated to assess it, and then the model is chosen that gets the highest results. Finally, the best set of hyperparameter values is used for the actual model training. Although GS provides the best set of parameter values, it often suffers from the dimensionality issue that further leads to local minima values of hyperparameters (Elgeldawi et al., 2021).

This work used Grid Search 5-fold cross-validation to tune the main hyperparameters (see Table 6). The work used scikit -learn (skLearn), a python-based library to implement GS-based CV on a training labeled dataset. Wherein, the GridSearchCV() method is responsible for searching optimized hyperparameter values by invoking another supporting fit() method, that would use to pass the ML classifier along with the datasets, labels, cross-validation folds (e.g., 2,5,10,.., i), and scoring measure(e.g., Accuracy). As a result, the best (optimum) set of parameter values corresponding to each ML / DL technique would be generated.

From the 5-fold GS cross-validation, the optimized resulting hyperparameter values are: CART DT (criterion = entropy, max_depth = 6), SVC (C=0.01, kernel=rbf), LR (C=0.01), RF (n_estimator=50), GBN (var_smoothing=0.4), LSTM (batch size = 64, dropout rate =0.3, and epochs=19).

Table 12a (Appendix B) shows the confusion matrix obtained when applying the grid search optimized hyperparameters to the proposed LSTM model for the news articles test dataset, and Table 12b (Appendix B) depicts the confusion matrix for the news tweets dataset. When computing the confusion matrix, the examples labeled as 1 (i.e., non-credible news) were considered as the negative examples and those labeled as 0 (i.e., credible news) were considered as the positive examples. All the performance measures of proposed models have been evaluated upon these confusion matrix values.

4.2 Experiment 2 – fake news classification with GA optimization

GA is the popular metaheuristic algorithm (MA) that is primarily a biologically inspired technique. Such techniques are known for the proofs to resolve non-convex and non-continuous optimization problems. During each iteration, MA usually starts by generating a population (i.e., a generation of solutions), individuals (i.e., each possible solution value), and chromosomes (i.e., the binary representation of a solution). Then for every generation, each possible individual (potential solution value) is examined until a global optimum value is discovered (Man et al., 1996).

According to Elgeldawi et al. (2021), in GA-based hyperparameter tuning each parameter is represented as a chromosome that is a decimal value for the hyperparameter. A chromosome consists of various binary encoded genes, then, these genes are subjected to chromosomal selection, crossover, and mutation processes in order to determine the best parameters. The chromosomes that have high fitness values are more likely to be chosen and handed down to the next iteration. The cycle of evolution among the best offspring from previous generations keep on going by carrying the best qualities of their parents. The random selection of initial population value is one of the advantages of GA that makes it easy to implement and foster the process of crossover, mutation, and selection to attain the optimum parameter values (Elgeldawi et al., 2021).

Lately, GA as an Evolutionary Algorithms is being used for Hyperparameter tuning (Rajan, 2021; Wirsansky, 2020) and also being adopted by or implemented by tech companies such as Elon Musk-backed OpenAI (Albanesius, 2017; Suryansh S., 2018). Moreover, python-based libraries such as TPOT (AutoML) (Le et al., 2020; Olson et al., 2016) are using GA to automatize the implementation of ML techniques to eliminate the model training and testing. This work has used model accuracy as a measure to check the fitness of optimum parameters because it shows the model's performance and how efficiently the model has classified the fake news vs real news. To implement the GA-based optimization, this work utilized sklearn-deap (sklearn-deap, 2021) – a python based library that provides all GA-based operations to tune the hyperparameters. Sklearn-deap is an amalgamation of scikit-learn and Distributed Evolutionary

Algorithms in Python (DEAP) libraries. Sklearn-deap is a time-effective library to get the best parameters fast (sklearn-deap, 2021). In the background, sklearn-deap uses the functionality of DEAP library (Fortin et al., 2012). DEAP involves the implementation of a highly complex framework of techniques belonging to the evolutionary computation family. The goal of DEAP is to provide an easy-to-use, understand, and feasible tool to build evolutionary algorithms such as GA.

This work utilized EvolutionaryAlgorithmSearchCV() method of skelarn-deap library that simulates the GA implementation and behavior. Wherein, estimator (the model), params (i.e., dictionary of or grid of all hyperparameters), scoring (i.e., evaluation measure e.g., accuracy), CV (k-fold cross validation value, e.g., 5,10), population_size (i.e.,), gene_mutation_prob (i.e., Probability of child mutation), gene_crossver_prob (i.e., Probability of crossover operation between two individuals), tournament_size(i.e., Number of individuals to perform tournament selection) and generation_num (i.e., Number of generations or iterations to run the evolutionary algorithm) used to pass as arguments (arenas, 2021; GASearchCV, 2021; sklearn-deap, 2021).

Observably, the utilization of sklearn-deap based GA implantation is required the wrapper library to provide the link between keras based deep learning models and GA based tuning. Therefore, this work has used KerasClassifier() method (i.e., build a link between keras model and sklearn methods) from keras.wrappers library. The dropout_rate = [0.2, 0.3, 0.5], batch_size = [32, 64, 128] and epochs = [5, 10, 15, 25] hyperparameters were passed to tune. And for EvolutionaryAlgorithmSearchCV() method, the parameters such as cv=5, population_size=50, gene_mutation_prob=0.10, gene_crossover_prob=0.5, tournament_size=3, and generations_number=5 along with the proposed model as estimator, were passed .

The GA optimized resulting hyperparameter values for LSTM are - batch size = 32, dropout rate =0.5, and epochs=25. Table 13a (Appendix B) shows the confusion matrix obtained after applying GA optimized hyperparameters to LSTM model for the news articles test dataset, and Table 13b (Appendix B) depicts the confusion matrix for the news tweets dataset.

Table 7. Comparing evaluation measures from both experiments

Experiment 1: Grid Search Hyperparameter optimization				
Datasets - News Articles				
	Precision	Recall	F1 Score	Accuracy
Proposed LSTM + Grid Search	93.08	86.72	89.79	**89.95**
Datasets - News Tweets				
Proposed LSTM + Grid Search	99.10	98.39	98.75	**98.12**
Experiment 2: GA Based Hyperparameter Optimization				
Datasets - News Articles				
Proposed GA + LSTM	95.31	91.04	93.13	**93.32**
Datasets - News Tweets				
	98.02	97.41	97.71	**96.57**

Table 7 shows the performance measure's values for the proposed model included in both the experiments; these measures help to compare the hyperparameter optimization effect of GridSearch and GA upon the proposed model. The test data has been supplied to all optimized models. Table 14 (Appendix B) shows the performance measure's values for all models including the proposed model for both the experiments; these measures help to compare the hyperparameter optimization effect of GridSearch and GA upon the proposed and all other models.

In experiment 1, LSTM based model achieved high precision scores, 93.08% for the news article test dataset and 99.10% for the news tweet dataset. The recall scores are higher: 86.72% for the news article test dataset and 98.39% for the tweet test dataset. The F1 scores are 89.79% for the news article test dataset and 98.75% for the news tweet test dataset. The proposed model also achieved a high accuracy of 89.95% in classifying the news articles in the test dataset and 98.12% in classifying the news tweets.

In experiment 2, LSTM based model achieved high precision scores, 95.31% for the news article test dataset and 98.02% for the news tweet dataset. The recall scores are higher: 91.04% for the news article test dataset and 97.41% for the tweet test dataset. Combining precision and recall, the obtained high F1 scores are: 93.13% for the news article test dataset and 97.71% for the news tweet test dataset. The model also achieved a high accuracy of 93.32% in classifying the news articles in the test dataset and 96.57% in classifying the news tweets.

For the news articles, the performance accuracy of LSTM has been increased from 89.95% to 93.32%, and for the news tweet data, the accuracy of LSTM is quite comparable among performed experiments, i.e., 98.12% and 96.57%, respectively. Hence, the GA-based optimization for the proposed LSTM performed well on the large textual news article data along with the short textual news tweets.

Then the performance of the proposed GA+LSTM model was compared with some of the existing models presented in (Castillo et al., 2011; Huu Do et al., 2021; Ma et al., 2015; Ni et al., 2021) that used the news tweet datasets. Accuracy as a measure was used for the comparison because not all the existing papers provided the measures such as recall, precision, and F1 score. As shown in Table 8, the proposed model can potentially outperform existing models for short-sequence news tweets.

Table 8. Compare GA+LSTM method with existing literature's techniques for fake news tweet detection

Models	GA + LSTM	GS + LSTM	SVM-DSTS (Ma et al., 2015)	DT-Rank (Castillo et al., 2011)	(Huu Do et al., 2021)	(Ni et al., 2021)
Accuracy	**96.57%**	98.12%	85%	86%	79.2%	92.34%

4.3 Discussion

The neural networks, such as LSTM, have built upon the working of the human brain, and evolutionary algorithms such as GA have built upon the concept of human evolvement; thus, obtained results have demonstrated that NN and GA can complement each other to solve real-time problems such as fake news detection. The reason behind gaining the improved performance measure values and the results is that the GA iteratively keeps on searching for the optimum deals and stalls the problem of local minima. Moreover, GA works well on large search space, i.e., with many hyperparameter values (Liashchynskyi & Liashchynskyi, 2019) and large-scale data such as news articles. Therefore, GA supports the LSTM network to detect fake news with finesse and implied findings align with the assumption of (Bakhashwain & Sagheer, 2021).

Table 9. Test accuracies for different feature sets

Feature Set	**Test Accuracies %**
Linguistic (POS) + Text Embeddings	98.00%
Tagging + Text Embeddings	98.00%
Tagging (@, #) + Sentiments + Text Embeddings	97.07%
Syntactic (Ngram) + Text Embeddings	95.00%
Tagging + Text Embeddings + Content-based	74.70

To identify the potent of extracted features, distinct combinations have been tried during the training of the proposed model. As shown in Table 9, all feature sets performed satisfactorily except the combination of tagging, embeddings, and content-based (length and word count). It has been observed that sentiments, tagging, linguistics, syntactic, and text embeddings are the features that have the potential to foster fake news detection through training the proposed model on various dimensionality to learn the contextual meaning of the news content.

Although, existing literature has a paucity of fake news detection theories because the area has been evolving in recent years. Hence, theoretically, presented features extraction has tied with the rationale of (X. Zhou et al., 2020). Wherein, X. Zhou et al. (2020) suggested concentrating on news content to detect fake news by incorporating lexicon-level, syntax-level, and semantic-level features. Moreover, X. Zhou et al. (2020) motivate the need for techniques that deeply mine news content. Thus, the incorporation of the aforementioned features with sentiments extraction and the combination of GA and LSTM fosters the ongoing research of fake news detection.

Practically, this research contributes to the ongoing efforts aimed at curtailing the spread of fake news on the web by proposing a deep learning model that is scalable and automates the process of identifying fake news. Complementing prior research that focuses on detecting fake news either in articles or in tweets (short text) that contain misinformation, the proposed model is equally effective in detecting fake news represented in both long-sequence texts such as fake news articles posted on websites such as Reddit.com and short-sequence texts such as tweets and in.

Methodologically, this work proposed a deep learning model that relies on Long Short-term Memory (LSTM), a special kind of Recurrent neural network (RNN). Researchers have found that it is challenging to label short-sequence data such as tweets as fake or not due to the semantic sparsity of short texts (Alrubaian et al., 2018; Boididou et al., 2018; Castillo et al., 2011; Kumar et al., 2016). LSTM is different from standard RNNs in that it is capable of learning long-term dependencies. This feature of LSTM, together with the word embedding method that is used in the proposed model, helps enrich the semantics of the short sequence texts resulting in improved performance compared to existing methods.

CHAPTER 5

CONCLUSIONS

5.1 General summary

The news may span across many online platforms, and the text of fake news may change and evolve as it spreads on the internet. Most of the techniques seem less efficient in recognizing fake news in real-world settings. This paper suggests that in fake news detection tasks, semantic, syntactic, sentimental, tagging, and contextual connections between real (fake) news are essential. This dissertation proposed the GA-optimized LSTM-based model for automatically detecting fake news on social media. The results have shown that GA has the potential to optimize the LSTM's hyperparameters to gain optimum performance accuracy. The proposed model complements prior approaches with demonstrated efficacy for both long sequences of text, e.g., news articles as well as a short sequence of text such as tweets. This is particularly important as fake news often propagates in a multitude of forms that exhibit distinctive semantic features that in prior work had to be accommodated using separate models. It is now possible to have one model to optimize, which can also handle such diversity in content. It has been demonstrated that the proposed approach works for both long sequence news articles and short-sequence texts such as tweets.

5.2 Limitations

This dissertation fosters the thoughts of existing literature that finding a benchmark labeled data set for fake news detection is challenging. Although there are quite a few publicly available datasets, but the authenticity of such sources is always in doubt. The presented work utilized the authenticated fake news datasets that have been used by existing researchers to detect fake news on social media platforms. Deep learning-based techniques often required large datasets and utilized data is limited in size. However, when textual data is converted into vectors (numerical forms), it acts as big data but to enhance the scalability of DL-based models, it is essential to provide a variety of data. Therefore, the scarcity of multidimensional datasets limits the scope of this work.

Moreover, the proposed model is able to detect fake news in an offline fashion whereas the propagation of news on the web happened in an online way. Therefore, the proposed approach has the limitation of training on real-time fake news data. However, the proposed model can be deployed online once an offline training is done but may require re-training to learn on new propagated news data.

CHAPTER 6

FUTURE SCOPE: REAL-TIME FAKE NEWS DETECTION

The aforementioned limitations lead to the future expansion of this work. Therefore, to eliminate the cited limitations, in the future, this work will utilize the Online Machine learning (OL) techniques to train the model on real-time generated news feeds. As follows, the preliminary study has been presented in this chapter that demonstrates the possibility of implementing OL techniques to detect real-time online fake news.

6.1 Introduction

Dissemination of misleading information on the internet has serious consequences for individuals, corporations, and society. As a result, academicians have turned their attention to the identification of misinformation. Detecting misinformation is inextricably linked to the classification task. Online machine learning (OL) has grown in prominence for text classification due to the creation of large and dynamic unstructured textual data on the web (Barve & Mulay, 2020). Such data contains enormous misleading information due to the ease of internet usage. However, researchers are incorporating machine learning techniques to eliminate the dispersed misinformation, but such algorithms are ineffective for handling newly incoming data over time and performing misinformation detection categorization to detect falsehoods. Supervised Machine Learning (ML) and Deep Learning (DL) techniques do not support OL thereby limiting the scope of performing misinformation detection or fake news detection for real-time settings (Burdisso et al., 2019; D. Wang et al., 2017). Due to learning on small batches of streaming big data, OL has the potential to increase the performance for knowledge acquisition and this requires less time and memory space (Shan et al., 2020).

According to Hoi et al. (2021), online learning refers to a group of machine learning techniques in which a model attempts to resolve a predictive problem by learning from a series of data sequences one by one and in an on-the-go fashion. The sole purpose of online learning is to improve the accuracy of the series of predictions made by the online ML model. Conversely, traditional batch or offline ML/DL approaches, are frequently intended to create a model from the complete training data set all at once. Online learning has shown to be a promising approach to learning from continuous streams of data (Hoi et al., 2021). According to Halford et al. (2019), existing literature comprises a huge knowledge base for utilizing traditional ML/DL approaches and only a few researchers have considered developing such techniques that can update from time to time to deal with the new input data. Thereby, limiting the knowledge base for future researchers to pursue the investigation of available OL techniques to detect real-time fake news on the web. In the OL techniques, the data inputs come in the form of streams (i.e., continuous series of information) continuously and the old data is discarded immediately after the model updates, therefore there is no need to retrain the model again and again

The over-ranching goal of this work is to automatically detect the real-time fake news by utilizing OL via incorporating textual features. This work has used news articles and news tweet datasets to train and evaluate the OL techniques.

6.2 Literature review

Existing studies such as (Cauwenberghs & Poggio, 2000; Shalev-Shwartz et al., 2011) have explored the realm of online machine learning techniques. In (Han et al., 2021), the author asserts that online data is usually growing in nature, so does the fake news dispersion therefore, it requires retraining the deployed fake news detection model again and again on new data which would be expensive. Although, techniques such as Gradient Episodic Memory (GEM) and Elastic Weight Consolidation (EWC) make it feasible to deal with historic and new data for model training to detect fake news via minor computational overhead.

Moreover, relevant studies such as (Babu et al., 2021; Janakieva et al., 2021; Nikam & Dalvi, 2020; Shaikh & Patil, 2020) are highly focused on adopting the Passive-Aggressive (PA) algorithm developed by (Crammer et al., 2006), that is an online learning algorithm and usually used for classification tasks. Wherein, the algorithm works passively when the predictions are correct and become aggressive to adjust the prediction in case of incorrect predictions, thereby known as a passive-aggressive algorithm (Shaikh & Patil, 2020). On the internet data presents in the form of an infinite sequence of words that are often treated as streams, therefore it is required to train models continuously on new data, in one data at a time manner (Bifet et al., 2018). In (Nikam & Dalvi, 2020), authors have used real-time Twitter streamed data to detect fake news via the PA technique through extracting TF-IDF features. Similarly, (Babu et al.,

2021) have demonstrated the effectiveness of PA for detecting large-scale fake news on social media.

The following table 10 has been extracted from (Montiel et al., 2021), which indicates the available python based library to implement OL models. Moreover, the table compares the performance of distinct library's OL techniques such as gaussian naïve byes (GNB), linear regression (LR), and HoeffdingTreeClassifier (HT) on benchmark Elec2 data – provided by the Australian New South Wales Electricity Market (link: https://www.kaggle.com/yashsharan/the-elec2-dataset).

Table 10. Benchmark accuracy (%) for the Ele2 dataset (Montiel et al., 2021).

Models	Scikit-learn	Creme	Scikit-multiflow	River
GNB	73.22	72.87	73.30	72.87
LR	68.01	67.97	NA	67.97
HT	NA	74.48	75.82	75.55

6.3 Method

6.3.1 Data collection:

For any classification task, there is a necessity for labeled datasets from authorized sources. When it comes to news content classification, there is a paucity of standard benchmark datasets (Ghosh & Shah, 2018). Two particularly relevant datasets have been identified and used in this research. The first dataset contains relatively long news articles, and the second one developed by PHEME (Zubiaga et al., 2016) includes short news tweets. These two datasets are used to evaluate the effectiveness of the proposed model for classifying news contents represented in long-sequence and short-sequence texts. First, News Article Dataset was developed by Ahmed et al. (2018). This dataset includes 12,600 fake news articles published on Kaggle.com and 12,600 truthful news articles from Reuters.com. The fake news articles were originally identified by Politifact.com, a fact-checking website that rates the accuracy of claims by governmental leaders and politicians. Articles in both the fake and truthful categories occurred in the same timeline, specifically in 2016. Binary numbers are used to label the articles. 1 is assigned to the fake news articles and 0 to the truthful news articles, as the focus is on detecting fake news. Second, News Tweet Dataset provided by PHEME consists of tweets related to real-world news events including the Ferguson protests, the shooting in Ottawa, the

Boston attack, the hostage situation in Sydney, and so on (Zubiaga et al., 2016). In this dataset, the tweets were labeled as "rumor" vs. "non-rumor". There are 1,972 tweets labeled as rumors and 2,830 labeled as non-rumors. This data is publicly available in JavaScript Object Notation (JSON) format with several directories, each of which includes several tweets. All tweets are collected in a single file, and again 1 is used to represent rumors and 0 to represent non-rumors.

6.3.2 Data preprocessing

For textual data, preprocessing is essential before supplying the text to the model learning process, as textual data usually contain redundancies and irregularities. Natural language processing techniques (NLP) have been used to preprocess the textual data and prepare them for further analysis. The stop words, special characters such as, '!', '&', '$', symbols such as emojis, repetitive period signs (e.g., .. or …), white spaces, line breaks, blank rows, and extra variables such as tweet ids and user ids, have been removed. The removed stop words include articles (a, an, the, etc.), pronouns (me, you, etc.), and prepositions (in, on, to, etc.) that have importance in English grammar for communication but do not have semantic importance in the learning process of the model. All the words in the datasets have been lowercased because the same word in different cases would be treated differently during the encoding process. Tokenization has been used to break the sentences into the token (words).

6.3.3 Feature extraction

Textual features have been extracted that are used in the model building. Part of speeches (POS) helps to determine the linguistics of the text data by assigning verbs, adverbs, adjectives, and nouns tags to each word in the news content. Such POS is helpful to understand the contextual meaning of the news sentences because often, the same word might have two entirely distinct meanings (Kouloumpis et al., 2011). For example, the phrases like "who was involved behind the Boston attacks" and "understanding reason behind Boston casualty is an involved matter" both have distinct contextual meanings; hence it is crucial to define the meaning of "involved" to determine whether or not these phrases are meaningfully related. Therefore, the POS has been extracted for news articles and tweets by utilizing Natural Language Toolkit (NLTK) (Bird et al., 2009) python library.

Primarily, this work utilizes news written in the English language; hence, unraveling the syntactic or syntax-related structure is crucial. N-grams are helpful to extract the relationship between words because, according to Weaver (1955), the word relationships in a sentence are complex, but it is possible to capture all relevant information by analyzing the order of words or which words tend to come together. Such co-occurring words in the corpora are known as N-grams, wherein N represents the number of words that co-occurred. Herein, Unigram shows the one word, Bigram offers the combination of two words, Trigram shows the three words, and so forth.

6.3.4 OL techniques

During the training of any OL technique, the data comes in the form of streams, and working on the data streams for model learning is yet to be explored. Such a data stream learning process is distinct from batch learning and supports real-time text classification tasks such as fake news detection. An OL technique as a classifier can be trained on previously acquired labeled datasets and can be used to detect the label for the new upcoming data. OL classifiers discard the old information and continue learning about the new data, one after another fashion (Bifet et al., 2018). To select the best OL techniques to incorporate in any business setting for early fake news detection, this work has used various techniques from the family of supervised binary classifiers such as Linear - Approximate Large Margin Algorithm (ALMA) and Passive-Aggressive (PA), Naïve based - Multinominal Naïve Bayes (MNB), Ensemble - Adaptive Boosting (Adaboost), Bagging, and Tree-based - Hoeffding Tree Classifier (HDT).

6.3.5 Evaluation

The OL models were applied to the test datasets and evaluated the classification performance of the models using the classical precision, recall, F1 score, and accuracy metrics. The formulas for computing recall, precision, F1 score, and accuracy are shown above. Recall refers to the rate of correctly classified positives among all positives and is equal to TP divided by the sum of TP and FN. Precision refers to the rate of correctly classified positives among all examples classified as positive and is equal to the ratio of TP to the sum of TP and FP. F1 score represents the harmonic mean of recall and precision. Accuracy represents the percentage of correct predictions. Where TP represents the number of True Positives, i.e., positive samples that were correctly classified. TN is the number of True Negatives, i.e., negative samples that were correctly classified, FP is the number of False Positive, i.e., negative samples that were incorrectly classified as positive, and FN is the number of False Negatives, i.e., positive samples incorrectly classified as negatives. These numbers were then used to calculate classification measures such as precision, recall, F1 score, and accuracy.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1 score} = \frac{2(\text{precision x recall})}{\text{precision} + \text{recall}} \qquad \text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

6.4 Results and discussion

Table 11 Evaluation measure's scores for OL techniques for each dataset.

Dataset: News Articles				
Techniques	**Accuracy %**	**Precision %**	**Recall %**	**F1 Score %**
ALMA	96.30	96.33	96.30	96.31
PA	99.15	99.16	99.15	99.15
MNB	97.46	97.67	97.47	95.15
Adaboost	74.29	67.59	74.29	67.12
Bagging	75.00	56.25	75.00	64.29
HDT	74.84	61.60	74.84	64.32
Dataset: News Tweets				
Techniques	**Accuracy %**	**Precision %**	**Recall %**	**F1 Score %**
ALMA	98.21	98.13	98.30	98.12
PA	96.86	97.12	97.35	96.72
MNB	96.19	96.17	96.20	96.18
Adaboost	82.39	81.71	83.27	81.56
Bagging	81.26	81.24	82.62	80.52
HDT	79.71	78.44	80.45	78.67

By utilizing River (Montiel et al., 2021) python-based library the OL techniques were trained on fake news tweets and articles data. The datasets were supplied to techniques in the form of the data stream in a one-at-a-time fashion. Table 11 shows the evolutionary measure scores of the adopted OL technique's performance on both datasets. For the news tweets, the PA technique predicted the outcomes well than the other OL techniques with 99.15% testing accuracy. However, MNB and ALMA also performed with 97.47% and 96.30% test accuracy, respectively. For the long sequential news article datasets, ALMA has outperformed all other OL techniques with 98.21% test accuracy.

REFERENCES

Abul-Fottouh, D., Song, M. Y., & Gruzd, A. (2020). Examining algorithmic biases in {YouTube}'s recommendations of vaccine videos. *International Journal of Medical Informatics*, *140*, 104175. https://doi.org/https://doi.org/10.1016/j.ijmedinf.2020.104175

Agrawal, R., de Alfaro, L., Ballarin, G., Moret, S., Di Pierro, M., Tacchini, E., & Della Vedova, M. L. (2019). Identifying Fake News from Twitter Sharing Data: A Large-Scale Study. *ArXiv Preprint ArXiv:1902.07207*. http://arxiv.org/abs/1902.07207

Ahmed, H., Traore, I., & Saad, S. (2018). Detecting opinion spams and fake news using text classification. *Security and Privacy*, *1*(1), e9. https://doi.org/10.1002/spy2.9

Ajao, O., Bhowmik, D., & Zargari, S. (2019). Sentiment Aware Fake News Detection on Online Social Networks. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2507–2511. https://doi.org/10.1109/ICASSP.2019.8683170

Albanesius, C. (2017, August 14). *Elon Musk's OpenAI Bot Beats Pro Dota 2 Player*. https://sea.pcmag.com/news/16981/elon-musks-openai-bot-beats-pro-dota-2-player

Ali, H., Khan, M. S., Alghadhban, A., Alazmi, M., Alzamil, A., Al-Utaibi, K., & Qadir, J. (2021). All Your Fake Detector are Belong to Us: Evaluating Adversarial Robustness of Fake-News Detectors Under Black-Box Settings. *IEEE Access*, *9*, 81678–81692. https://doi.org/10.1109/ACCESS.2021.3085875

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, *31*(2), 211–236.

Allcott, H., Gentzkow, M., & Yu, C. (2019). Trends in the diffusion of misinformation on social media. *Research \& Politics*, *6*(2), 2053168019848554.

Alrubaian, M., Al-Qurishi, M., Hassan, M. M., & Alamri, A. (2018). A Credibility Analysis System for Assessing Information on Twitter. *IEEE Transactions on Dependable and Secure Computing*, *15*(4), 661–674. https://doi.org/10.1109/TDSC.2016.2602338

Altché, F., De, A., & Fortelle, L. (2017). An LSTM Network for Highway Trajectory Prediction. *20th International Conference on Intelligent Transportation Systems*, 353–359.

arenas, rodrigo. (2021). *GitHub - rodrigo-arenas/Sklearn-genetic-opt: Hyperparameters tuning and feature selection, using evolutionary algorithms.* https://github.com/rodrigo-arenas/Sklearn-genetic-opt/

Babu, A. M., Catharine, J. P., Divya, K. J., & Joffi, A. (2021). HAWKSEYE-A Machine Learning-Based Technique for Fake News Detection with IoT. *IJISRT*, *6*(6).

Bakhashwain, N., & Sagheer, A. (2021). Online tuning of hyperparameters in deep LSTM for time series applications. *International Journal of Intelligent Engineering and Systems*, *14*(1), 212–220.

Barve, Y. N., & Mulay, P. (2020). Bibliometric Survey on Incremental Learning in Text Classifification Algorithms for False Information Detection. *Library Philosophy and Practice*, 1–21.

Bhutani, B., Rastogi, N., Sehgal, P., & Purwar, A. (2019). Fake News Detection Using Sentiment Analysis. *2019 Twelfth International Conference on Contemporary Computing (IC3)*, 1–5. https://doi.org/10.1109/IC3.2019.8844880

Bifet, A., Gavalda, R., Holmes, G., & Pfahringer, B. (2018). *Machine learning for data streams: with practical examples in MOA*. MIT press.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc."

Boididou, C., Papadopoulos, S., Zampoglou, M., Apostolidis, L., Papadopoulou, O., & Kompatsiaris, Y. (2018). Detection and visualization of misleading content on Twitter. *International Journal of Multimedia Information Retrieval*, *7*(1), 71–86. https://doi.org/10.1007/s13735-017-0143-x

Bollen, J., Mao, H., & Pepe, A. (2010). Determining the Public Mood State by Analysis of Microblogging Posts. *ALIFE*, 667–668.

Bovet, A., & Makse, H. A. (2019). Influence of fake news in Twitter during the 2016 US presidential election. *Nature Communications*, *10*(1).

Brownlee, J. (2017). *Deep learning for natural language processing: develop deep learning models for your natural language problems*. Machine Learning Mastery.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122.

Burdisso, S. G., Errecalde, M., & Montes-y-Gómez, M. (2019). A text classification framework for simple and effective early depression detection over social media streams. *Expert Systems with Applications*, *133*, 182–197.

Candogan, O., & Drakopoulos, K. (2020). Optimal signaling of content accuracy: Engagement vs. misinformation. *Operations Research*, *68*(2), 497–515.

Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. *Proceedings of the 20th International Conference on World Wide Web - WWW '11*, 675. https://doi.org/10.1145/1963405.1963500

Cauwenberghs, G., & Poggio, T. (2000). Incremental and decremental support vector machine learning. *Advances in Neural Information Processing Systems*, *13*.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). *Online passive aggressive algorithms*.

De Alfaro, L., Polychronopoulos, V., & Shavlovsky, M. (2015). Reliable aggregation of boolean crowdsourced tasks. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, *3*, 42–51.

Dennis, A. K. and A. R. (2019). Says Who? The Effects of Presentation Format and Source Rating on Fake News in Social Media. *MIS Quarterly*, *43*(3), 1025–1039. https://doi.org/10.25300/MISQ/2019/15188

Ebadi, N., Jozani, M., Choo, K.-K. R., & Rad, P. (2021). A Memory Network Information Retrieval Model for Identification of News Misinformation. *IEEE Transactions on Big Data*, 1. https://doi.org/10.1109/TBDATA.2020.3048961

Elgeldawi, E., Sayed, A., Galal, A. R., & Zaki, A. M. (2021). Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. *Informatics*, *8*(4). https://doi.org/10.3390/informatics8040079

Ezen-Can, A. (2020). A Comparison of LSTM and BERT for Small Corpus. *ArXiv*, *abs/2009.0*.

Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. In *Automated machine learning* (pp. 3–33). Springer, Cham.

Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A. G., Parizeau, M., & Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *The Journal of Machine Learning Research*, *13*(1), 2171–2175.

Friggeri, A., Adamic, L., Eckles, D., & Cheng, J. (2014). Rumor cascades. *Proceedings of the International AAAI Conference on Web and Social Media*, *8*(1), 101–110.

GASearchCV. (2021). *GASearchCV — sklearn genetic opt 0.8.0 documentation*. https://sklearn-genetic-opt.readthedocs.io/en/stable/api/gasearchcv.html#sklearn_genetic.GASearchCV

Ghenai, A. (2017). Health {Misinformation} in {Search} and {Social} {Media}. *Proceedings of the 40th {International} {ACM} {SIGIR} {Conference} on {Research} and {Development} in {Information} {Retrieval}*, 1371. https://doi.org/10.1145/3077136.3084153

Ghenai, A., & Mejova, Y. (2018). Fake {Cures}: {User}-centric {Modeling} of {Health} {Misinformation} in {Social} {Media}. *Proceedings of the ACM on Human-Computer Interaction*, *2*(CSCW), 1–20. https://doi.org/10.1145/3274327

Ghosh, S., & Shah, C. (2018). Towards automatic fake news classification. *Proceedings of the Association for Information Science and Technology*, *55*(1), 805–807. https://doi.org/10.1002/pra2.2018.14505501125

Giachanou, A., & Crestani, F. (2016). Like It or Not. *ACM Computing Surveys*, *49*(2), 1–41. https://doi.org/10.1145/2938640

Graves, A., & Schmidhuber, J. (2005). Framewise Phoneme Classification with Bidirectional LSTM Networks. *IEEE International Joint Conference on Neural Networks*, 2047–2052.

Grinberg, N., Joseph, K., Friedland, L., Swire-Thompson, B., & Lazer, D. (2019). Fake news on Twitter during the 2016 U.S. presidential election. *Science*, *363*(6425), 374–378. https://doi.org/10.1126/science.aau2706

Halford, M., Bolmier, G., Sourty, R., Vaysse, R., & Zouitine, A. (2019). *creme, a Python library for online machine learning*.

Han, Y., Karunasekera, S., & Leckie, C. (2021). Continual Learning for Fake News Detection from Social Media. *International Conference on Artificial Neural Networks*, 372–384.

Heinrich, K., Zschech, P., Skouti, T., Griebenow, J., & Riechert, S. (2019). Demystifying the Black Box: A Classification Scheme for Interpretation and Visualization of Deep Intelligent Systems. *AMCIS*, 8.

Hevner, A. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 75–105. https://www.researchgate.net/publication/201168946

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hoi, S. C. H., Sahoo, D., Lu, J., & Zhao, P. (2021). Online learning: A comprehensive survey. *Neurocomputing*, *459*, 249–289.

Holland, J. H., & others. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Hou, R., Perez-Rosas, V., Loeb, S., & Mihalcea, R. (2019). Towards {Automatic} {Detection} of {Misinformation} in {Online} {Medical} {Videos}. *2019 {International} {Conference} on {Multimodal} {Interaction}*, 235–243. https://doi.org/10.1145/3340555.3353763

Hutto, C. J., & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media*, *8*(1). http://sentic.net/

Huu Do, T., Berneman, M., Patro, J., Bekoulis, G., & Deligiannis, N. (2021). Context-Aware Deep Markov Random Fields for Fake News Detection. *IEEE Access*, *9*, 130042–130054. https://doi.org/10.1109/ACCESS.2021.3113877

Janakieva, D., Mirceva, G., & Gievska, S. (2021). Fake News Detection by Using Doc2Vec Representation Model and Various Classification Algorithms. *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, 223–228.

Janze, C., & Risius, M. (2017). Automatic Detection of Fake News on Social Media Platforms. *PACIS 2017 Proceedings*. https://aisel.aisnet.org/pacis2017/261

Jin, Z., Cao, J., Zhang, Y., Zhou, J., & Tian, Q. (2017). Novel Visual and Statistical Image Features for Microblogs News Verification. *IEEE Transactions on Multimedia*, *19*(3), 598–608. https://doi.org/10.1109/TMM.2016.2617078

Jurkowitz, M., & Mitchell, A. (2020). *Early in outbreak, Americans cited claims about risk level and details of coronavirus as made-up news*. Pew Research Center Jornalism & Media. https://www.journalism.org/2020/04/15/early-in-outbreak-americans-cited-claims-about-risk-level-and-details-of-coronavirus-as-made-up-news/

Karmiani, D., Kazi, R., Nambisan, A., Shah, A., & Kamble, V. (2019). Comparison of Predictive Algorithms: Backpropagation, SVM, LSTM and Kalman Filter for Stock Market. *Proceedings - 2019 Amity International Conference on Artificial Intelligence, AICAI 2019*, 228–234. https://doi.org/10.1109/AICAI.2019.8701258

Kochkina, E., Liakata, M., & Zubiaga, A. (2018). All-in-one: Multi-task Learning for Rumour Verification Title and Abstract in Russian. *ArXiv Preprint ArXiv:1806.03713*, 3402–3413.

Kouloumpis, E., Wilson, T., & Moore, J. (2011). Twitter Sentiment Analysis: The Good the Bad and the OMG! *International AAAI Conference*, 538–541.

Kouzy, R., Abi Jaoude, J., Kraitem, A., El Alam, M. B., Karam, B., Adib, E., Zarka, J., Traboulsi, C., Akl, E., & Baddour, K. (2020). Coronavirus {Goes} {Viral}: {Quantifying} the {COVID}-19 {Misinformation} {Epidemic} on {Twitter}. *Cureus*. https://doi.org/10.7759/cureus.7255

Koza, J. R., & Rice, J. P. (1991). Genetic generation of both the weights and architecture for a neural network. *IJCNN-91-Seattle International Joint Conference on Neural Networks*, *ii*, 397–404 vol.2. https://doi.org/10.1109/IJCNN.1991.155366

Kreps, S., & Kriner, D. (2020, June). Medical Misinformation in the Covid-19 Pandemic. *SSRN*. osf.io/jbgk9

Kumar, S., West, R., & Leskovec, J. (2016). Disinformation on the Web. *Proceedings of the 25th International Conference on World Wide Web - WWW '16*, 591–602. https://doi.org/10.1145/2872427.2883085

Kwon, S., & Cha, M. (2014). Modeling bursty temporal pattern of rumors. *Eighth International AAAI Conference on Weblogs and Social Media*.

Le, T. T., Fu, W., & Moore, J. H. (2020). Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics*, *36*(1), 250–256.

Lee, K., Eoff, B., & Caverlee, J. (2021). Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter. *Proceedings of the International AAAI Conference on Web and Social Media*, *5*(1), 185–192. https://ojs.aaai.org/index.php/ICWSM/article/view/14106

Li, D., Guo, H., Wang, Z., & Zheng, Z. (2021). Unsupervised Fake News Detection Based on Autoencoder. *IEEE Access*, *9*, 29356–29365. https://doi.org/10.1109/ACCESS.2021.3058809

Liao, Q., Chai, H., Han, H., Zhang, X., Wang, X., Xia, W., & Ding, Y. (2021). An Integrated Multi-Task Model for Fake News Detection. *IEEE Transactions on Knowledge and Data Engineering*, 1. https://doi.org/10.1109/TKDE.2021.3054993

Liashchynskyi, P., & Liashchynskyi, P. (2019). *Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS*. https://arxiv.org/abs/1912.06059v1

Liu, X., Wang, G. A., Fan, W., & Zhang, Z. (2020). Finding useful solutions in online knowledge communities: A theory-driven design and multilevel analysis. *Information Systems Research*, *31*(3), 731–752.

Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K.-F., & Cha, M. (2016). Detecting Rumors from Microblogs with Recurrent Neural Networks. *AAAI Press*.

Ma, J., Gao, W., Wei, Z., Lu, Y., & Wong, K.-F. (2015). Detect Rumors Using Time Series of Social Context Information on Microblogging Websites. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM '15*, 1751–1754. https://doi.org/10.1145/2806416.2806607

Man, K.-F., Tang, K.-S., & Kwong, S. (1996). Genetic algorithms: concepts and applications [in engineering design]. *IEEE Transactions on Industrial Electronics*, *43*(5), 519–534.

Montiel, J., Halford, M., Mastelini, S. M., Bolmier, G., Sourty, R., Vaysse, R., Zouitine, A., Gomes, H. M., Read, J., Abdessalem, T., & others. (2021). *River: machine learning for* *streaming data in Python*.

Mukherjee, S., & Weikum, G. (2015). Leveraging Joint Interactions for Credibility Analysis in News Communities. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM '15*, 353–362. https://doi.org/10.1145/2806416.2806537

Ni, S., Li, J., & Kao, H.-Y. (2021). MVAN: Multi-View Attention Networks for Fake News Detection on Social Media. *IEEE Access*, *9*, 106907–106917. https://doi.org/10.1109/ACCESS.2021.3100245

Nikam, S. S., & Dalvi, R. (2020). Machine learning algorithm based model for classification of fake news on twitter. *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 1–4.

Niknam, F., Samadbeik, M., Fatehi, F., Shirdel, M., Rezazadeh, M., & Bastani, P. (2020). {COVID}-19 on {Instagram}: {A} content analysis of selected accounts. *Health Policy and Technology*. https://doi.org/https://doi.org/10.1016/j.hlpt.2020.10.016

Olson, R. S., Urbanowicz, R. J., Andrews, P. C., Lavender, N. A., Kidd, L. C., & Moore, J. H. (2016). *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 -- April 1, 2016, Proceedings, Part I* (G. Squillero & P. Burelli (eds.); pp. 123–137). Springer International Publishing. https://doi.org/10.1007/978-3-319-31204-0_9

Papanastasiou, Y. (2020). Fake news propagation and detection: A sequential model. *Management Science*, *66*(5), 1826–1846.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pennycook, G., Bear, A., Collins, E. T., & Rand, D. G. (2020). The implied truth effect: Attaching warnings to a subset of fake news headlines increases perceived accuracy of headlines without warnings. *Management Science*, *66*(11), 4944–4957.

Popat, K., Mukherjee, S., Strötgen, J., & Weikum, G. (2018a). CredEye: A credibility lens for analyzing and explaining misinformation. *Companion Proceedings of the The Web Conference 2018*, 155–158. https://doi.org/10.1145/3184558.3186967

Popat, K., Mukherjee, S., Strötgen, J., & Weikum, G. (2018b). CredEye. *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18*, 155–158. https://doi.org/10.1145/3184558.3186967

Popat, K., Mukherjee, S., Strötgen, J., & Weikum, G. (2016). Credibility Assessment of Textual Claims on the Web. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management - CIKM '16*, 2173–2178. https://doi.org/10.1145/2983323.2983661

Popat, K., Mukherjee, S., Yates, A., & Weikum, G. (2019). *DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning*. 22–32. https://doi.org/10.18653/v1/d18-1003

Porat, T., Garaizar, P., Ferrero, M., Jones, H., Ashworth, M., & Vadillo, M. A. (2019). Content and source analysis of popular tweets following a recent case of diphtheria in {Spain}. *European Journal of Public Health*, *29*(1), 117–122. http://www.ezproxy.dsu.edu:2048/login?url=https://www.ezproxy.dsu.edu:2155/scholarly-journals/content-source-analysis-popular-tweets-following/docview/2306531218/se-2?accountid=27073

Porter, M. F. (1980). An algorithm for suffix stripping. *MCB UP Ltd*. https://doi.org/10.1108/00330330610681286

Qazvinian, V., Rosengren, E., Radev, D. R., & Mei, Q. (2011). Rumor has it: identifying misinformation in microblogs. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1589–1599. https://dl.acm.org/citation.cfm?id=2145602

Qi, P., Cao, J., Yang, T., Guo, J., & Li, J. (2019). Exploiting Multi-domain Visual Information for Fake News Detection. *2019 IEEE International Conference on Data Mining (ICDM)*, 518–527. https://doi.org/10.1109/ICDM.2019.00062

Rajan, S. (2021, June 15). *Genetic Algorithms and its use-cases in Machine Learning*. https://www.analyticsvidhya.com/blog/2021/06/genetic-algorithms-and-its-use-cases-in-machine-learning/

Ratkiewicz, J., Conover, M., Meiss, M., Gonçalves, B., Patil, S., Flammini, A., & Menczer, F. (2011). Truthy: mapping the spread of astroturf in microblog streams. *Proceedings of the 20th International Conference Companion on World Wide Web*, 249–252.

Reis, J. C. S., Correia, A., Murai, F., Veloso, A., & Benevenuto, F. (2019). Supervised Learning for Fake News Detection. *IEEE Intelligent Systems*, *34*(2), 76–81. https://doi.org/10.1109/MIS.2019.2899143

Ritchie, M. D., White, B. C., Parker, J. S., Hahn, L. W., & Moore, J. H. (2003). Optimizationof neural network architecture using genetic programming improvesdetection and modeling of gene-gene interactions in studies of humandiseases. *BMC Bioinformatics*, *4*(1), 28. https://doi.org/10.1186/1471-2105-4-28

Sear, R. F., Velasquez, N., Leahy, R., Restrepo, N. J., Oud, S. El, Gabriel, N., Lupu, Y., & Johnson, N. F. (2020). Quantifying {COVID}-19 {Content} in the {Online} {Health} {Opinion} {War} {Using} {Machine} {Learning}. *IEEE Access*, *8*, 91886–91893. https://doi.org/10.1109/ACCESS.2020.2993967

Shaikh, J., & Patil, R. (2020). Fake News Detection using Machine Learning. *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (ISSSC)*, 1–5.

Shalev-Shwartz, S., Singer, Y., Srebro, N., & Cotter, A. (2011). Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical Programming*, *127*(1), 3–30.

Shan, G., Xu, S., Yang, L., Jia, S., & Xiang, Y. (2020). Learn\#: A Novel incremental learning method for text classification. *Expert Systems with Applications*, *147*, 113198.

Shearer, E. (2020, November 2). *Two-thirds of US adults say they've seen their own news sources reported facts mean to favor one side*. Pew Research Center. https://www.pewresearch.org/fact-tank/2020/11/02/two-thirds-of-u-s-adults-say-theyve-seen-their-own-news-sources-report-facts-meant-to-favor-one-side/

Shihab Elbagir and Jing Yang. (2019). *Twitter Sentiment Analysis Using Natural Language Toolkit and VADER Sentiment*. 16.

Shrivastava, G., Kumar, P., Ojha, R. P., Srivastava, P. K., Mohan, S., & Srivastava, G. (2020). Defensive Modeling of Fake News Through Online Social Networks. *IEEE Transactions* *on Computational Social Systems*, *7*(5), 1159–1167. https://doi.org/10.1109/TCSS.2020.3014135

Shu, K., & Liu, H. (2019). Detecting fake news on social media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, *11*(3), 1–129.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: a data mining perspective. CoRR abs/1708.01967 (2017). *ArXiv Preprint ArXiv:1708.01967*.

sklearn-deap. (2021). *GitHub - rsteca/sklearn-deap: Use evolutionary algorithms instead of gridsearch in scikit-learn*. https://github.com/rsteca/sklearn-deap

Smaldone, F., Ippolito, A., & Ruberto, M. (2020). The shadows know me: {Exploring} the dark side of social media in the healthcare field. *European Management Journal*, *38*(1), 19–32. https://doi.org/https://doi.org/10.1016/j.emj.2019.12.001

Suryansh S. (2018, March 26). *Genetic Algorithms + Neural Networks = Best of Both Worlds | by Suryansh S. | Towards Data Science*. https://towardsdatascience.com/gas-and-nns-6a41f1e8146d

Villafranca, E., & Peters, U. (2019). Smart and Blissful? Exploring the Characteristics of Individuals That Share Fake News on Social Networking Sites. *AMCIS 2019 Proceedings*. https://aisel.aisnet.org/amcis2019/virtual_communities/virtual_communities/10

Vyas, P., & Dubey, J. (2013). An Efficient Methodological Study for Optimization of Negative Association Rule Mining. *IJCA*, *ICRTITCS*(3), 27–31.

Wang, D., Al-Rubaie, A., Clarke, S. S., & Davies, J. (2017). Real-time traffic event detection from social media. *ACM Transactions on Internet Technology (TOIT)*, *18*(1), 1–23.

Wang, P., Xu, B., Xu, J., Tian, G., Liu, C. L., & Hao, H. (2016). Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing*, *174*, 806–814.

Wang, X., Jiang, W., & Luo, Z. (2016). Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts. *26th International Conference on Computational Linguistics: Technical Papers*, 2428–2437. https://github.com/ultimate010/crnn

Waszak, P. M., Kasprzycka-Waszak, W., & Kubanek, A. (2018). The spread of medical fake news in social media – {The} pilot quantitative study. *Health Policy and Technology*, *7*(2), 115–118. https://doi.org/https://doi.org/10.1016/j.hlpt.2018.03.002

Weaver, W., & others. (1955). Translation. *Machine Translation of Languages*, *14*(15–23), 10.

Wicaksono, A. S., & Supianto, A. A. (2018). Hyper parameter optimization using genetic algorithm on machine learning methods for online news popularity prediction. *International Journal of Advanced Computer Science and Applications*, *9*(12), 263–267.

Wijenayake, S., Hettiachchi, D., Hosio, S., Kostakos, V., & Goncalves, J. (2021). Effect of Conformity on Perceived Trustworthiness of News in Social Media. *IEEE Internet Computing*, *25*(1), 12–19. https://doi.org/10.1109/MIC.2020.3032410

Wirsansky, E. (2020). *Hands-on genetic algorithms with Python: applying genetic algorithms to solve real-world deep learning and artificial intelligence problems*. Packt Publishing Ltd.

Wu, L., Morstatter, F., Carley, K. M., & Liu, H. (2019). Misinformation in Social Media: Definition, Manipulation, and Detection. *SIGKDD Explor. Newsl.*, *21*(2), 80–90. https://doi.org/10.1145/3373464.3373475

Yuan, H., Zheng, J., Ye, Q., Qian, Y., & Zhang, Y. (2021). Improving fake news detection with domain-adversarial and graph-attention neural network. *Decision Support Systems*, 113633. https://doi.org/https://doi.org/10.1016/j.dss.2021.113633

Zafarani, R., Abbasi, M. A., & Liu, H. (2014). *Social media mining: an introduction*. Cambridge University Press.

Zhou, C., Sun, C., Liu, Z., & Lau, F. C. M. (2015). A C-LSTM Neural Network for Text Classification. *ArXiv Preprint ArXiv:1511.08630*. http://arxiv.org/abs/1511.08630

Zhou, X., Jain, A., Phoha, V. V, & Zafarani, R. (2020). Fake News Early Detection: A Theory-Driven Model. *Digital Threats: Research and Practice*, *1*(2). https://doi.org/10.1145/3377478

Zubiaga, A., Liakata, M., & Procter, R. (2016). Learning Reporting Dynamics during Breaking News for Rumour Detection in Social Media. *ArXiv Preprint ArXiv:1610.07363*.

APPENDICES

APPENDIX A: ABBREVIATIONS

AI – Artificial intelligence

ALMA – Approximate Large Margin Algorithm

CART DT – Classification and Regression Trees Decision Tree

CV – Cross-Validation

DL – Deep Learning

DEAP – Distributed Evolutionary Algorithms in Python

GA – Genetic Algorithm

GNB – Gaussian Naïve Bayes

GS – Grid Search

IL – Incremental Learning

LSTM – Long Short-Term Memory

LR – Logistic Regression

LogReg – Logistic Regression

ML – Machine Learning

NLP – Natural Language Processing

OL – Online Machine Learning

OCL – Out-of-core learning

RF – Random Forest

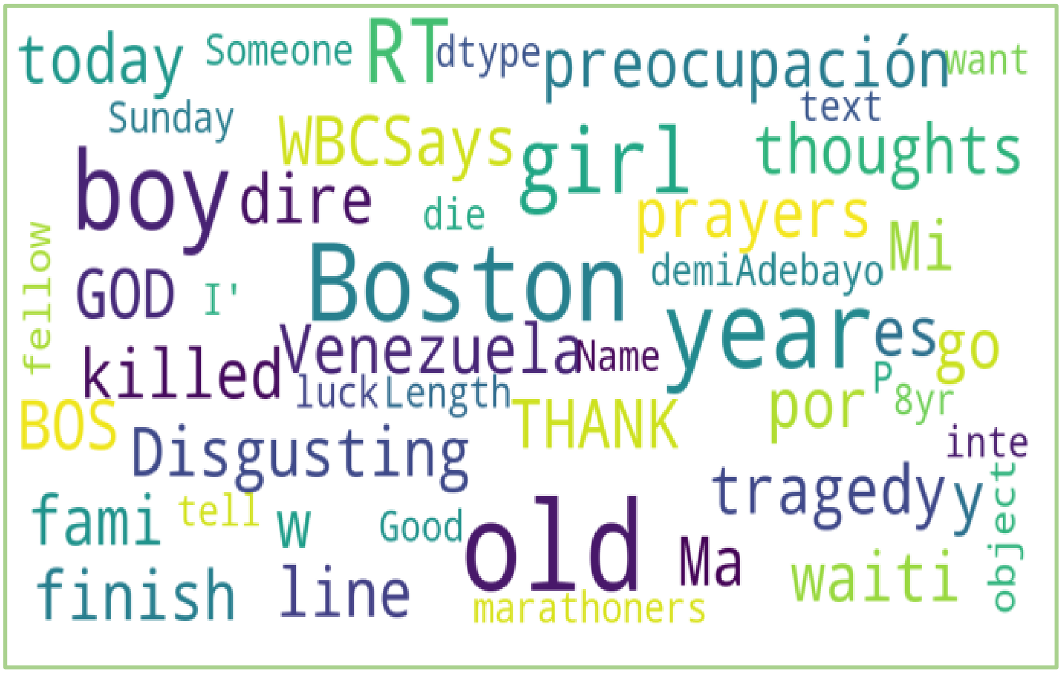
ReLU - Rectified Linear Units

RNN – Recurrent Neural Network

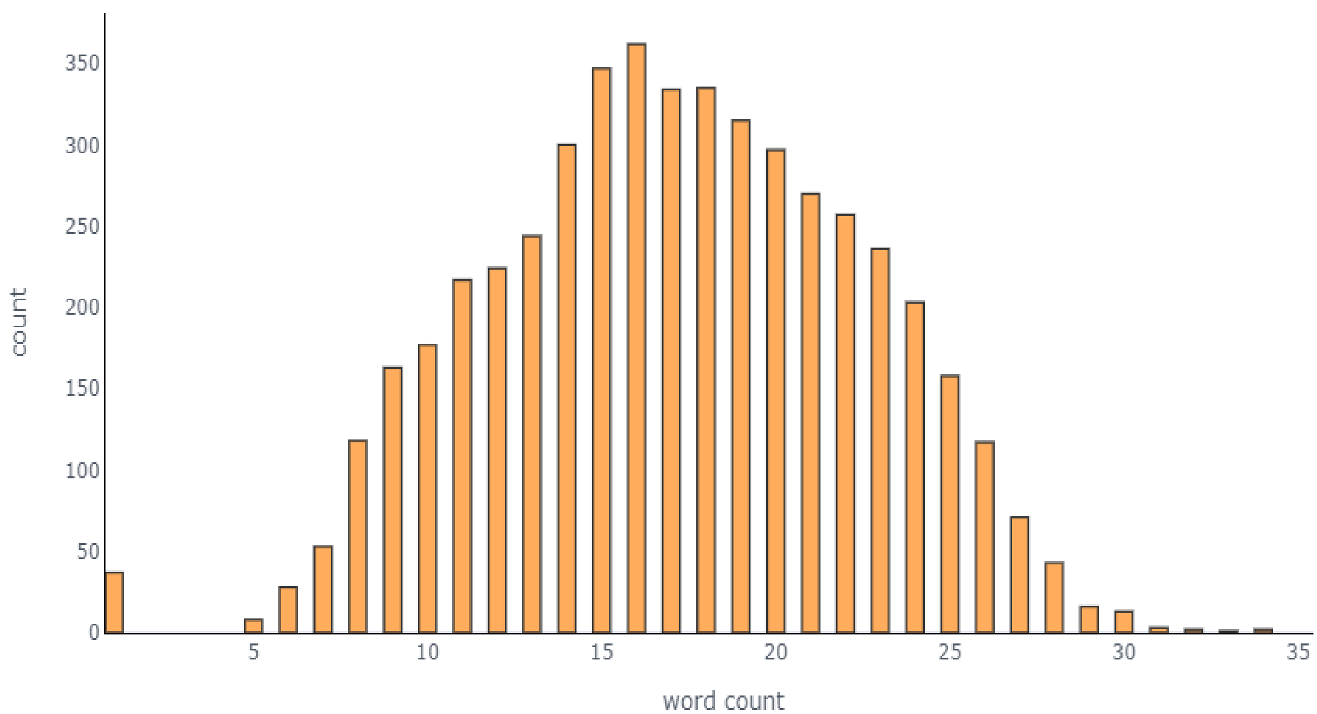
SVC – Support Vector Classifier

SkLearn – Sci-Kit Learn

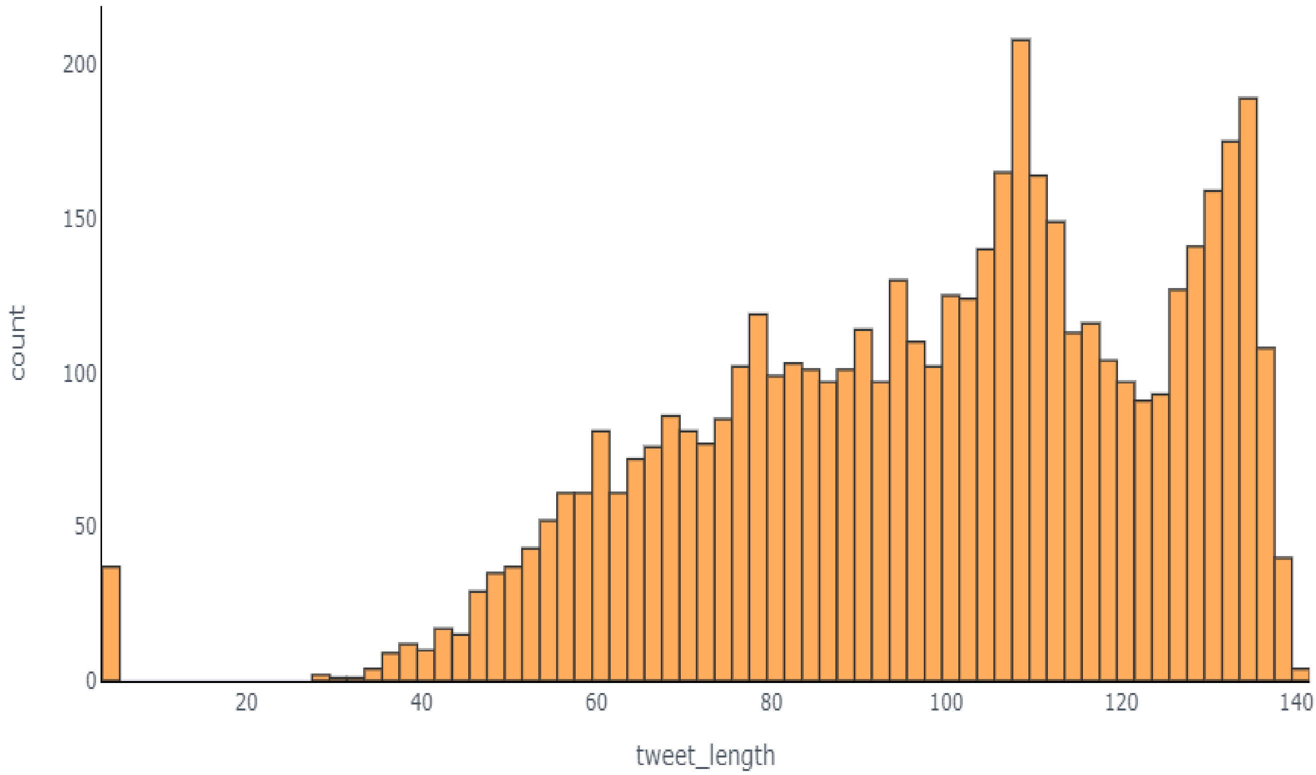
APPENDIX B: ADDITIONAL RESULTS



(a)



(b)



(c)

Figure 7. a) Word cloud for most frequent words; b) words count in tweets; c) characters count in tweets

Table 12. a) GS based Confusion matrix for the news article test set; b) GS based Confusion matrix for the news tweet test set

	Actual Positive	**Actual Negative**
Predicted Positive	3970 (TP)	295 (FP)
Predicted Negative	608 (FN)	4110 (TN)

a)

	Actual Positive	**Actual Negative**
Predicted Positive	1102 (TP)	10 (FP)
Predicted Negative	18 (FN)	356 (TN)

b)

Table 13. a) GA based Confusion matrix for the news article test set; b) GA based Confusion matrix for the news tweet test set

	Actual Positive	**Actual Negative**
Predicted Positive	4065(TP)	200 (FP)
Predicted Negative	400 (FN)	4318 (TN)

a)

	Actual Positive	**Actual Negative**
Predicted Positive	1090 (TP)	22 (FP)
Predicted Negative	29 (FN)	345 (TN)

b)

Table 14. Comparison of distinct ML/DL technique performances from both experiments

Experiment 1: Grid Search Hyperparameter optimization						
Datasets - **News Articles**						
	SVC	CART DT	GNB	LR	RF	**Proposed LSTM + Grid Search**
Precision	87.92	87.92	82.06	86.75	89.10	93.08
Recall	84.12	83.00	74.18	81.89	84.11	86.72
F1 Score	85.98	85.39	77.92	84.25	86.53	89.79
Accuracy	86.39	85.72	77.92	84.60	86.83	**89.95**
Datasets - **News Tweets**						
Precision	95.50	96.40	78.06	96.58	97.39	99.10
Recall	98.24	97.99	91.37	98.08	98.63	98.39
F1 Score	96.85	97.19	84.19	97.33	98.01	98.75
Accuracy	95.36	95.83	78.06	96.03	97.0%	**98.12**
Experiment 2: GA Based Hyperparameter Optimization						
Datasets - **News Articles**						
	SVC	CART DT	GNB	LR	RF	**Proposed GA + LSTM**
Precision	89.45	91.79	80.07	85.93	91.79	95.31
Recall	90.51	90.52	81.21	90.38	88.47	91.04
F1 Score	89.98	91.15	80.64	88.10	90.10	93.13
Accuracy	90.54	91.54	81.74	88.98	90.43	**93.32**
Datasets - **News Tweets**						
Precision	97.75	97.30	80.22	97.12	97.12	98.02
Recall	96.11	96.43	92.72	96.43	96.09	97.41
F1 Score	96.92	96.87	86.02	96.77	96.60	97.71
Accuracy	95.36	95.29	80.48	95.15	94.89	**96.57**