Explainable Machine Learning for Robust Modelling in Healthcare

WATSON, MATTHEW,STEVEN

Explainable Machine Learning for Robust Modelling in Healthcare

Matthew Watson

A thesis presented for the degree of
Doctor of Philosophy

Durham University

Department of Computer Science
Durham University
United Kingdom
May 2023

Abstract

Deep Learning (DL) has seen an unprecedented rise in popularity over the last decade, with applications ranging from machine translation to self-driving cars. This includes extensive work in sensitive domains such as healthcare and finance with, for example, models recently achieving better-than-human performance in tasks such as chest x-ray diagnosis. However, despite these impressive results there are relatively few real-world deployments of DL models in sensitive scenarios, with experts claiming this is due to a lack of model transparency, reproducibility, robustness and privacy; this is in spite of numerous techniques having been proposed to address these issues. Most notably is the development of Explainable Deep Learning techniques, which aim to compute feature importance values for a given input (i.e. which features does a model use to make its decision?) - such methods can greatly improve the transparency of a model, but have little impact on reproducibility, robustness and privacy. In this thesis, I explore how explainability techniques can be used to address these issues, by using feature attributions to improve our understanding of how model parameters change during training, and across different hyperparameter setups. Through the introduction of a novel model architecture and training technique that used model explanations to improve model consistency, I show how explanations can improve privacy, robustness and reproducibility. Extensive experimentation is carried out across a number of sensitive datasets from healthcare and bioinformatics in both traditional and federated learning settings show that these techniques have a significant impact on the quality of these models. I discuss the impact these results could have on real-world applications of deep learning, due to the issues addressed by the proposed techniques, and present some ideas for further research in this area.

Declaration

The work in this thesis is based on research carried out at the Department of Computer Science, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

Firstly, I would like to thank my supervisor Noura Al Moubayed for her guidance throughout this PhD; without this, you would not be reading this thesis. A special thanks must also go to Bashar Awwad Shiekh Hasan for his valued time and input across my time as a PhD student - his endless feedback has helped shape my writing and figure-producing style into what it is today. I must also thanks my colleagues in our research group for our insightful group meetings, as well as all the other academic and admin staff at the Department of Computer Science, along with those at Evergreen Life who supported me with this work.

It would be remiss of me to not mention my friends, both those in Durham and those not, who have helped make my time in Durham an enjoyable one. A particular shoutout must go to those who (somehow) put up with me during the many Covid lockdowns that plagued much of my postgraduate studies; you helped me though some tough times, and for that I will be forever grateful. How you ever put up with my incessant complaining I do not know, but if you hadn't this PhD would not have been as fun as it has been.

Of course, the biggest thanks must go to my family. To Steven, my father: thank you for your constant kindness throughout the past 24 years, even when times were tough; here's the proof that I actually have been working over the past 3 years! To my youngest brother, Christopher: turns out being decent at maths does get you places - who'd have thought it?

And to those who, sadly, are no longer with us. To Simon, my brother: your time with us was far too short, and I wish I would've had a chance to be your elder brother. I'm sure your support would have been invaluable during the completion of this thesis. To Lynne, my mother: you are sorely missed, and forever remembered. You helped shape the person I am today, and I hope you'd be proud of what I've achieved - hopefully you're looking down at me with a smile on your face, knowing that I've had you in my thoughts every step along this journey.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Over the past decade, Deep Learning (DL) has seen a meteoric rise in popularity. From advancing machine translation [3], to autonomous vehicles [4] and even creating new artwork [5] it is hard to find areas of our lives that have not been affected by the field. DL's popularity largely stems from its ability to be successfully applied to a wide-range of scenarios, with models beginning to outperform humans in some tasks [6]. However, the field is not without its problems: concerns around data privacy [7], biased decision making [8], lack of transparency [9] and model robustness [10, 11] plague DL topics and affect real-world uptake in the techniques. Not only do researchers have a moral obligation to address many of these issues but, as lawmakers catch up with the rapid rise of neural networks with the introduction of laws such as the European Union's General Data Protection Regulations [12], they must also be addressed to allow for further adoption of DL models in real-world applications.

This is even more imperative in sensitive domains such as healthcare and finance [13], where these issues are significant roadblocks to the implementation of DL models in clinical or financial practice. For example, whilst there has been significant process in the area of automatic Chest X-Ray (CXR) diagnosis [14] using neural networks, there are currently few-to-no applications of these models in day-to-day practice despite evidence showing they can provide real-world benefit to radiologists [15]. DL applications which have been approved by medical bodies such as the US' Food and Drug Administration (FDA) usually do so without explicitly labelling themselves as such - possibly due to the increased difficulties these technologies traditionally face when under the scrutiny of public bodies [16].

These issues are numerous, and must be addressed before we are to see widespread adoption of Machine Learning (ML) in many sensitive settings such as healthcare and finance. While it is simple to say that models must be more trustworthy for them to be used in these settings [13], this does not shed much light on *how* we can make ML models trustworthy. In fact, *trustworthiness* can be thought of as many constituent parts: explainability, transparency [17], quality (i.e. does it learn causal or correlated features?) [18], privacy [19], robustness [20] and more.

Each of these individual problems have been extensively studied, with many different approaches being suggested to combat them. For example, numerous different explainability techniques have been proposed [21–23], each with their own advantages and disadvantages, which claim to "open up" black-box deep learning models. Similarly, techniques such as Federated Learning [24] and Differential Privacy [19] have been proposed to improve the privacy provided by DL models; architectures such as hyperensembles [25] are suggested to create more robust models; and causal learning techniques have been proposed to improve the quality of learned features [26].

As these are all separate, independent strands of DL research it is not common to see them all applied in practice: in order to create an end-to-end model that addresses all of these issues, the amount of research and implementation one would have to do to apply all of these independent techniques would be immense. Instead, this thesis takes a more unified approach, utilising explainability techniques to uncover some of the issues with modern, deep neural networks. By using a unified approach, the techniques presented in this thesis are able to encompass all of the above problems by using explainability alone. Finally, these same explainability techniques are used to create a single approach that aims to address many of the problems (such as privacy, explainability, robustness and transparency) that plague modern deep learning models, and compare how this approach compares to each of the current state of the art techniques in each of the sub-fields mentioned above.

1.1 Motivation

As previous studies have shown, advances in Deep Learning mean that models can be successfully applied to increasingly complex tasks. Examples include applications such as chest x-ray diagnosis [27], diabetes risk prediction [28], money laundering detection [29] and criminal re-offending likelihood prediction [30]. All of these models have one thing in common: they operate in high risk settings, where the stakes are high and the data used is extremely confidential. In such highly sensitive environments, model explainability, trustworthiness and robustness is of paramount importance; for example, clinicians and patients alike are unlikely to trust a black-box DL model which cannot explain its decisions [13].

Recently, many new explainability techniques have been developed that attempt to address the problems of transparency and trustworthiness [31] with numerous methods being proposed that aim to provide explanations for a DL model's decision [21, 32, 33]. Similarly, there are approaches that improve model robustness [25, 34] and generalisability [34–36]. However, despite these advances, there are still comparatively few real-world deployments of DL in domains such as healthcare and finance when compared to other less sensitive domains. Though the exact reasons for this can differ between disciplines, many experts agree that there are four main overarching concerns: data privacy [12], transparency [13], bias [37] and robustness [9]. Without significant advances in these areas it is unlikely we will see widespread adoption of DL techniques, despite the many advantages that they could bring.

To date, all of these issues have been addressed independently of one another: explainability techniques are proposed for transparency, privacy techniques for data confidentiality and so on. However, many argue that many of these issues all stem from our lack of understanding of the mathematical foundations of modern machine learning [38] - essentially, they are all results of the black-box nature of deep neural networks. In this thesis, I explore how explainability techniques can be used to open up this black-box and, thus, how model explainability can improve not only model transparency but also model privacy, robustness and generalisability. I focus experiments on settings where these problems are of paramount importance, such as healthcare and bioinformatics, to explore how the techniques proposed throughout the thesis can be applied to (and improve) highly sensitive applications. The overall hope, then, is that the novel methods suggested can be used to allow DL to be applied to situations in healthcare, bioinformatics and beyond to have a positive impact on people's lives.

1.2 Publications

The work contained in this thesis is the result of a number of linked peer-reviewed publications I have produced throughout my PhD. In particular:

- Chapter 3 contains results from Watson, Matthew, and Al Moubayed, Noura. "**Attack-agnostic adversarial detection on medical data using explainable machine learning.**" *2020 25th International Conference on Pattern Recognition (ICPR).* IEEE, 2021.

- Chapter 4 contains results from Watson, Matthew and Awwad Shiekh Hasan, Bashar and Al Moubayed, Noura. "**Agree to disagree: When deep learning models with identical architectures produce distinct explanations**." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 2022.

- Chapter 5 consists of the method and results in Watson, Matthew and Awwad Shiekh Hasan, Bashar and Al Moubayed, Noura. "**Using model explanations to guide deep learning models towards consistent explanations for EHR data.**" *Scientific Reports* 12.1 (2022): 1-14.

- Chapter 6.1 consists of the work presented in Watson, Matthew and Awwad Shiekh Hasan, Bashar and Al Moubayed, Noura. "**Learning How to MIMIC:**

Using Model Explanations to Guide Deep Learning Training." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 2023.

- Chapter 6.2 Watson, Matthew and Awwad Shiekh Hasan, Bashar and Al Moubayed, Noura. "**Explainability-based Membership Inference Attacks and Defences**." *Scientific Reports* Under Review.

I also contributed to the following pieces of work which were also published during my PhD, although they do not necessarily fit into this thesis' theme:

- Zuo, Zheming and Watson, Matthew and Budgen, David and Hall, Rob and Kennelly, Chris and Al Moubayed, Noura "**Data Anonymization for Pervasive Health Care: Systematic Literature Mapping Study.**" *JMIR Medical Informatics* 9.10 (2021)
- Alhassan, Zakhriya and Watson, Matthew, et al. "**Improving Current Glycated Hemoglobin Prediction in Adults: Use of Machine Learning Algorithms With Electronic Health Records.**" *JMIR Medical Informatics* 9.5 (2021)
- Watson, Matthew and Chambers, Pinkie, et al. "**1859P Using deep learning with demographic and laboratory values from baseline to cycle 2 to predict subsequent renal and hepatic function.**" *Annals of Oncology* 32 (2021)

1.3 Thesis Structure and Contributions

The thesis starts with Chapter 2, which is a thorough review of current DL methods as applied to sensitive applications such as healthcare and bioinformatics and the hurdles that these algorithms must overcome to become more widely used in practice. I then move on to go in-depth on the methods that have been developed in an attempt to overcome these problems, providing thorough descriptions of all commonly used explainability methods, and metrics used to evaluate the effectiveness of these techniques (as well as providing some rationale of why each different algorithm exists). These explainability techniques are used throughout this thesis as both ways to explore how neural network training works, and as a tools to improve the quality of deep learning models. Chapter 2 continues with an exploration of our current understanding of model robustness and generalisation, as well as techniques that are commonly used to improve these attributes; this forms the basis of how I evaluate the methods proposed in Chapter 5. Next I explore data privacy in deep learning, along with training algorithms such as Federated Learning (FL) which are designed to address these concerns, with these techniques being revisited in Chapter 6.2 where I investigate the methods presented in Chapter 5 from a data privacy perspective.

Chapter 3 examines how off-the-shelf explainability techniques can be used to detect when malicious inputs are passed to DL models. Through the development of two novel auxiliary model architectures that utilise feature attributions, I show how adversarial inputs can be detected in both an attack- and model-agnostic manner. I explore the efficacy of these techniques, showing that they beat current state-of-the-art adversarial attack detection techniques, discuss the implications of this for sensitive applications such as healthcare and why it seems to be the case that explanations are so effective at classifying perturbed samples.

The success of the methods presented in Chapter 3 shows that model explanations are sensitive to even imperceptible changes to inputs. Inspired by this, in Chapter 4 I explore the use of explainability techniques to explore the training of deep neural networks. Specifically, I look at how hyperparameters that are orthogonal to the downstream task (such as the random seed or order of the training data) can vastly affect the input features used by the model (even when model performance is near identical). I discuss numerous avenues one could take to measure this inconsistency, and present a final quantitative metric that can be used to evaluate the *explanation inconsistency* of DL models. Finally, this chapter briefly discusses the implications of these results and why it is imperative that the problem is addressed.

The original post-processing technique described in Chapter 5 is designed to address the problem of explanation inconsistency which was introduced in the previous chapter. Thorough experimentation of this technique on a number of healthcare and biological tabular datasets verify that this technique does indeed reduce explanation inconsistency however, as is discussed within the chapter, it also has a number of disadvantages. To tackle these disadvantages, I take inspiration from the investigation as to *why* the post-processing technique works and embed this into model training. This results in an entirely novel Deep Explanation Ensemble (DEE) architecture and training procedure that utilises feature attributions during training and produces models with significantly better explanation consistency than current state-of-the-art methods.

In Chapter 6 I extend the evaluation of DEEs to other data modalities, problems and training scenarios. In Chapter 6.1, DEEs are tested on chest x-ray images from the MIMIC-CXR-EGD dataset. Interestingly, this dataset also includes the eye-gaze data from an experienced radiologist when analysing the same images. Through a comparison of this eye-gaze data with the explanations produced by a variety of different models, I show that the more consistent explanations of DEEs also have significantly more overlap with the expert's eye-gaze data. Then, in Chapter 6.2, DEEs are applied to the Federated Learning (FL) training setup and are shown to vastly improve upon the susceptibility of such models to Membership Inference Attacks whilst still achieving high levels of performance, unlike other privacy-preserving techniques such as Differential Privacy (DP).

Finally, Chapter 7.4 consolidates all previous chapters, discussing how the novel techniques presented in later chapters can be used to address the barriers to DL adoption in sensitive scenarios that were addressed in the earlier chapters.

1.4 Notation

A neural network $M : \mathcal{X} \rightarrow \mathcal{Y}$ is a function that takes an input from a d-dimensional training domain and outputs a value from the test domain; in this thesis, the target will be either a class or regression value. A specific feature of any given input $x \in \mathcal{X}$ is defined as x_i. An explanation, $E(M(x)) \in \mathbb{R}^d$, is a feature attribution map that quantifies the contribution of each feature of the input x to the value of $M(x)$ (i.e. how important is each feature to the final classification); the contribution of an individual feature is dented by $E_i(M(x))$.

CHAPTER 2

Literature Review

Machine Learning (ML), and in particular Deep Learning (DL), research has seen a dramatic rise in use in healthcare applications [39]. DL has successfully been applied to areas such as medical imaging [40], Electronic Health Record (EHR) analysis [28] and bioinformatics [41] in research settings, with some DL models achieving levels of accuracy that match [42] or even exceed [6] medical experts. However, despite these impressive results, it is still rare to see DL models actually deployed into real-world environments [16] and used day-to-day by clinicians. The main barrier to further adoption of DL in clinical settings is gaining the trust of medical professionals; in particular, clinicians cite the lack of transparency around how DL models make predictions as well as privacy and data security concerns as the main issues surrounding DL in healthcare [13, 43].

This chapter reviews current applications of DL to the healthcare domain, and how it has been applied in practice. For the majority of this thesis, a thorough understanding of the mathematics, notation and vocabulary outside of what is defined in this section is not needed. A basic understanding of probability will be required for some future definitions; these basic definitions have been omitted for the sake of brevity, but readers requiring a refresh are directed to [44]. Similarly, I assume a basic knowledge of AI/DL definitions - neural networks, basic network structures and the training process of a neural network are assumed knowledge. Again, these basic definitions have not been included in this section for the need to stop somewhere - it is necessary to assume some base level of knowledge - and as the teaching of DL techniques at (or even before) and undergraduate level is now commonplace, this seems like a good baseline. Readers interested in a review of these definitions are also referred back to [44].

It then briefly summarise the barriers DL must overcome before seeing widespread adoption in medicine before discussing the most recent advances in DL that are aimed at addressing these issues, such as explainability techniques for deep learning models and DL model robustness.

2.1 Deep Learning Applications and Datasets in Healthcare

There are three main areas that DL has been applied to in healthcare: medical imaging, EHR data analysis and bioinformatics. These advancements are made possible by the release of several large-scale medical datasets that enable the training of DL models. These datasets range from real-world data collected from hospitals that have been made freely available, such as MIMIC-CXR [45] and MIMIC-IV [46], to smaller scale genomics datasets such as the Codon Usage dataset [47]. Due to privacy concerns around publicly sharing patient's private data, experiments are also commonly carried out on private datasets [48]. Where possible, situations like these should be avoided due to transparency and reproducibility concerns, however it is sometimes an unavoidable consequence of sensitive data. This sub-section explores the largest and most frequently used healthcare and bioinformatics datasets and explains how they have been used to advance the field of healthcare DL. I then go on to explain some of the issues facing DL models in healthcare, and discuss why we haven't seen more widespread adoption of DL models in clinical settings. All datasets are summarised in Table 2.1 alongside the current state of the art model performance on each task, which are used for baseline comparisons in all future chapters.

Table 2.1: Summary table of all datasets used, along with basic dataset statistics and the state of the art results used as baselines for comparison throughout all chapters.

		Dataset Properties				
Dataset	**Modality**	**Num. Samples**	**Num. Features**	**Num. Classes**	**Federated**	**Baseline Accuracy**
MNIST	Images	60,000	784	10	✗	99 [49]
FEMNIST	Images	805,263	784	62	✓	85 [50]
Synthetic	Tabular	734,463	72	12	✓	70 [50]
INaturalist	Images	2.7M	150528	10	✓	84 [51]
COMPAS	Tabular	7214	466	2	✗	90 [30]
Adult	Tabular	32,561	205	2	✗	75 [52]
Texas	Tabular	348,700	252	100	✗	84 [53]
MIMIC-CXR-EGD	Images	1,083	150528	3	✗	76 [1]
MIMIC-CXR (Pneumonia)	Images	377,110	150528	2	✗	84 [27]
MIMIC-CXR (Cardiomegaly)	Images	377,110	150528	2	✗	82 [27]
MIMIC-IV (Mortality)	Tabular, Time Series	383,220	Variable	2	✗	81 [54]
Henan-Renmin	Tabular	110,300	62	2	✗	73 [54]
Codon Usage (Kingdom)	Tabular	130,000	64	5	✗	84 [47]
Codon Usage (DNA)	Tabular	130,000	64	3	✗	99 [47]
KAIMRC (Classification)	Tabular	66,652	15	2	✗	83 [28]
KAIMRC (Regression)	Tabular	66,652	15	1	✗	N/A
BCW	Tabular	569	30	2	✗	99 [52]

2.1.1 Small-Scale Healthcare Datasets

It is only within the last 5 years that truly large-scale healthcare datasets have become commonplace. Before this, most datasets consisted of comparatively few records (e.g. less than 5000) and only a small number of features. However, these small datasets are still commonly used as initial baseline tests when evaluation novel deep learning techniques. For example, the Breast Cancer Wisconsin (BCW) [55] dataset is a small 30-dimensional dataset of 569 records that contains features extracted from images of (possible) breast cancer cells. This dataset is commonly used to test binary classification models, where the task is to predict whether the associated cell is malignant or not.

Similarly, the Pima Diabetes Dataset [56] is an 8-dimensional dataset of 768 patients from Phoenix, Arizona, USA. The dataset consists of general patient demographic information (e.g. age), basic health-related variables (e.g. Body Mass Index and number of pregnancies) as well as some blood-test results. The goal of machine learning classifiers trained on this dataset is to predict whether or not a given patient has diabetes. Although by today's standards this dataset is small, it is still commonly used to test the validity of a technique before moving on to test on larger datasets [57, 58].

2.1.2 CheXpert

CheXpert [27] is a large, publicly available Chest X-Ray (CXR) dataset that consists of 224,316 chest radiographs from 65,240 patients who attended Stanford Hospital between 2002 and 2017. Using a rule-based label extraction tool on each image's associated (free-text) radiology report, each image is given up to 14 different labels: No Finding, Enlarged Cardiom., Cardiomegaly, Lung Lesion, Lung Opacity, Edema, Consolidation, Pneumonia, Atelectasis, Pneumothorax, Pleural Effusion, Pleural Other, Fracture, and Support Devices. Importantly, this dataset improved upon previous CXR datasets by including *uncertain* mentions of a label in the free-text report with each image having 4 possible values for each of the 14 labels: 0 (negative mention, i.e. "*No signs of pneumonia*"), 1 (positive mention, i.e. "*Signs of pneumonia*"), -1 (uncertain mention, i.e. "*May be signs of pneumonia*"), or *blank*.

The automatic labeler was evaluated on a hold-out validation set of 1000 radiology reports from 1000 distinct patients. Each of these validation reports were examined by 2 expert radiologists, and their diagnoses compared with the automatically extracted label from the original report. The labeler gained a macro-average F1 score of 0.948 on positive mentions, 0.899 for negative mentions and 0.770 for uncertain mentions.

2.1.3 MIMIC-IV

The Medical Information Mart for Intensive Care v4 (MIMIC-IV) [46] dataset is a large, real-world dataset consisting of EHR data from patients admitted to intensive care (ICU) at the Beth Israel Deaconess Medical Center, Massachusetts, USA. The dataset was extracted from the hospital's e-prescribing software and consists of data on 383,220 ICU stays that occurred between 2008 and 2019. Each record contains data on the patient's demographics (e.g. age, sex, ethnicity, comorbidities, etc.) as well as time-series data that contains their vital signs (e.g. heart rate, blood oxygen level), lab results (e.g. the results of blood tests) and medications prescribed throughout their stay in ICU. Each record also contains the reason they were admitted to the ICU, any diagnoses that were made during their stay as well as the length of their stay.

Typically, MIMIC-IV is used to train mortality prediction models, although it has also been used for multi-label disease classification and EHR knowledge graph generation. It is important to note that the data in MIMIC-IV has been fully anonymised, and so some patient data may be unavailable or masked. For example, instead of providing the actual dates that a patient was admitted to and discharged from hospital, modified *anchor dates* are provided. These anchor dates have been carefully constructed such that the length of each patient's stay remains the same, but the exact dates the patient was in hospital for cannot be calculated.

2.1.4 MIMIC-CXR

MIMIC-CXR [45] is a special subset of patients included in the MIMIC-IV dataset for which we also have a number of Chest X-Ray (CXR) images that were taken during the patient's stay in hospital. MIMIC-CXR contains 377,110 separate CXR scans, taken from 227,835 radiography studies (i.e. each study may contain multiple images). The patients included in the MIMIC-CXR dataset were admitted to the ICU between 2010 and 2016, and their CXR images can be linked with their patient data contained in MIMIC-IV. Originally, the MIMIC-CXR dataset consisted of the raw DICOM files for each CXR image - these are large, uncompressed data files that come straight from the x-ray machine, and require a large amount of pre-processing before they can be used by a deep learning model. However, more recently, the MIMIC-CXR-JPG [59] dataset released JPEG versions of these images after this pre-processing had been applied. This is advantageous not only because it significantly reduces the amount of computational power that needs to be spent on processing the images, but also because it standardised the pre-processing that is applied to the DICOM images.

Each study is related to a single patient, though each study may contain multiple images (usually from different angles) and each patient may have multiple associated studies. Each study also contains a free-text report written by the experienced radiologist treating the patient. Often, this dataset is used alongside labels automatically extracted from these reports using the CheXpert labeler (Chapter 2.1.2).

The CheXpert labeler was validated against the findings of 2 expert radiologists on a hold-out validation set of size 687 from the MIMIC-CXR-JPG dataset, gaining an average F1 score (across all 14 labels) of 0.83. Typically, these 14 labels are used to train either a binary classification model (to predict 1 of the 14 findings) or a multi-label classification model.

Typically, MIMIC-CXR is used to train models on this multi-label classification task, although sometimes specific labels of interest are chosen. Due to the usually small number of CXR images available, the de-facto standard for training such models is to finetune a pre-trained image classification model. This is a process where a network that has already been trained on a (usually extremely large) image dataset is taken and trained on the CXR classification task, with the idea being that the pre-trained model will have already learned to recognise important features that may be shared across both datasets [60]. There is no standard model architecture used for this throughout the CXR analysis literature, although Densenet-121, ResNet-18 and ResNet-80 pretrained on the ImageNet [61] dataset are some of the most commonly used methods [14] and as such are used throughout the rest of this thesis as baseline models for comparative purposes.

More recently, Vision Transformers (ViT) [62] have overtaken classic, large CNNs for image classification. Inspired by its success in Natural Language Processing, the ViT architecture adapts the BERT Transformer-based architecture [63] to image classification. Briefly, ViTs first split a given image into patches which are then linearly embedded and added to a positional embedding. This embedding is then passed to a standard Transformer, which creates the final feature map - for classification tasks, an extra learnable classification token may also be added. Although ViTs have been shown to outperform classic CNN architectures on standard image classification tasks such as ImageNet and CIFAR [62], they require an inordinate amount of data during training. This makes them difficult to apply to CXR images, where the amount of data is typically very limited - indeed, I briefly investigate the use of Vision Transformers in Chapter 6.2.4, and find that with the extremely limited amount of public data available they are unable to achieve even baseline levels of performance.

With the full release of MIMIC-IV, patient data can be linked between MIMIC-CXR and MIMIC-IV, opening up opportunities for multi-modal models to be proposed. This data linkage also allows for a wider range of classification tasks to be performed, e.g. predicting comorbidities rather than the labels extracted from the radiology reports.

2.1.5 MIMIC-CXR-EGD

The MIMIC-CXR-EGD [1] is a subset of 1,083 CXR images from the MIMIC-CXR-JPG dataset that have been used to collect eye-gaze data (EGD) from an expert radiologist. Through the use of eye-tracking software the radiologist's eye-gaze was recorded whilst they were analysing the image to provide a diagnosis, resulting in a heatmap of how long was spent looking at each area of the image (an example heatmap can be seen in Figure 2.2). A recording of radiologist's speech whilst analysing the image is also included (as well as an auto-generated transcript of this speech). This data is checked for validity throughout the process: the audio transcript was manually checked by three experts and checked by a qualified radiologist, whilst the eye-gaze data was calibrated throughout the data collection process by presenting 59 calibration images at different points during the analysis process. Alongside this information is a set of automatically-generated bounding boxes of the image, containing each of the following important areas: 'right lung', 'right upper lung zone', 'right mid lung zone', 'right lower lung zone', 'left lung', 'left upper lung zone', 'left mid lung zone', 'left lower lung zone', 'right hilar structures', 'left hilar structures', 'upper mediastinum', 'cardiac silhouette', 'trachea', 'right costophrenic angle', 'left costophrenic angle', 'right clavicle', 'left clavicle'.

Both raw eye gaze information and calculated fixation points are available for the expert's EGD. Both sets of data are calculated automatically from the eye gaze tracking software used by the study. The raw eye gaze data is a fine-grained dataset, containing data for each individual data sample collected. On the other hand, the fixation dataset contains one data point per fixation, which is gained by averaging the raw data to detect saccades.

The images included in this analysis were selected from the overall MIMIC-CXR-

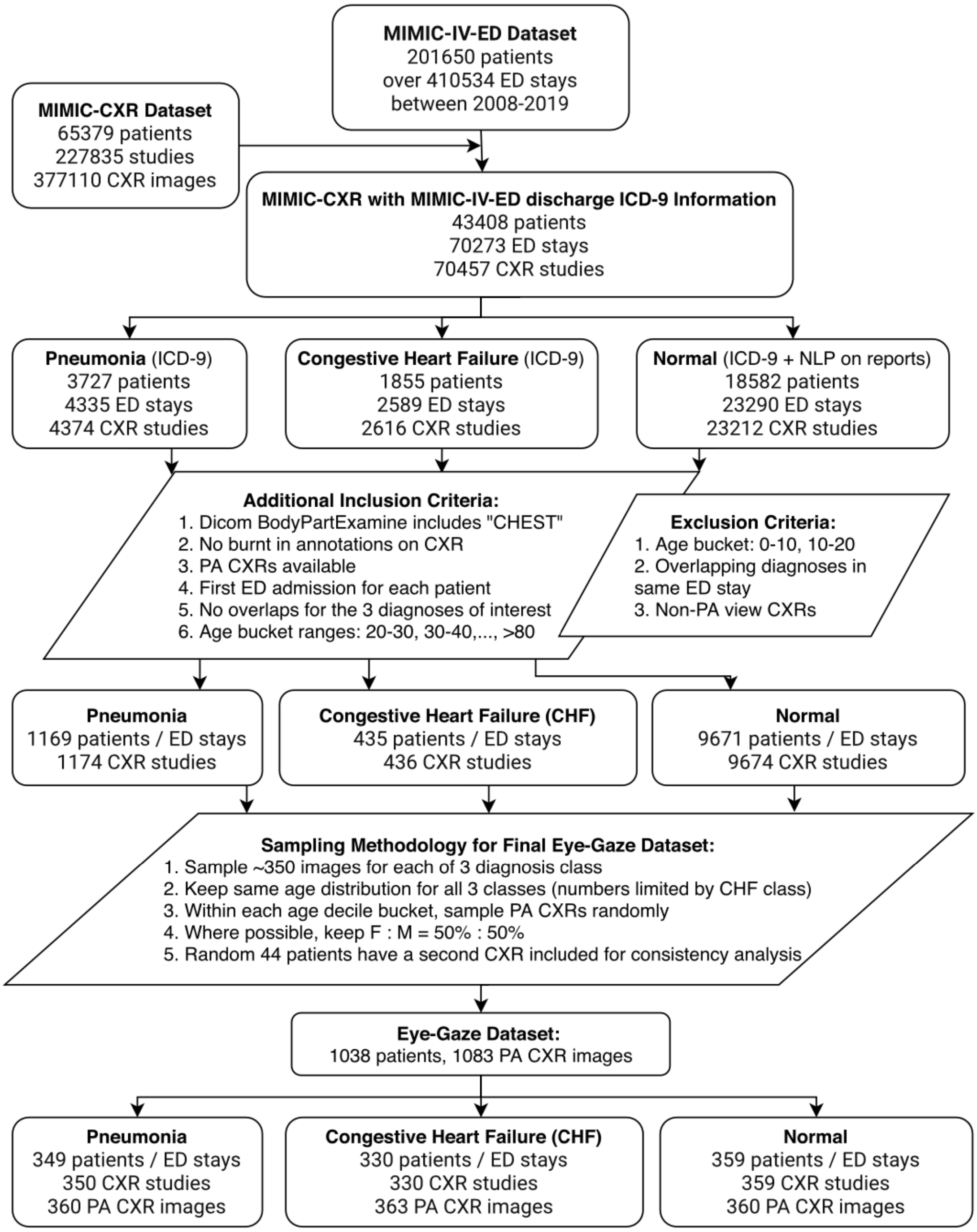

MIMIC-IV-ED Dataset
201650 patients
over 410534 ED stays
between 2008-2019
MIMIC-CXR Dataset
65379 patients
227835 studies
377110 CXR images
MIMIC-CXR with MIMIC-IV-ED discharge ICD-9 Information
43408 patients
70273 ED stays
70457 CXR studies
Pneumonia (ICD-9)
3727 patients
4335 ED stays
4374 CXR studies
Congestive Heart Failure (ICD-9)
1855 patients
2589 ED stays
2616 CXR studies
Normal (ICD-9 + NLP on reports)
18582 patients
23290 ED stays
23212 CXR studies
Additional Inclusion Criteria:
1. Dicom BodyPartExamine includes "CHEST"
2. No burnt in annotations on CXR
3. PA CXRs available
4. First ED admission for each patient
5. No overlaps for the 3 diagnoses of interest
6. Age bucket ranges: 20-30, 30-40,..., >80
Exclusion Criteria:
1. Age bucket: 0-10, 10-20
2. Overlapping diagnoses in same ED stay
3. Non-PA view CXRs
Pneumonia
1169 patients / ED stays
1174 CXR studies
Congestive Heart Failure (CHF)
435 patients / ED stays
436 CXR studies
Normal
9671 patients / ED stays
9674 CXR studies
Sampling Methodology for Final Eye-Gaze Dataset:
1. Sample ~350 images for each of 3 diagnosis class
2. Keep same age distribution for all 3 classes (numbers limited by CHF class)
3. Within each age decile bucket, sample PA CXRs randomly
4. Where possible, keep F : M = 50% : 50%
5. Random 44 patients have a second CXR included for consistency analysis
Eye-Gaze Dataset:
1038 patients, 1083 PA CXR images
Pneumonia
349 patients / ED stays
350 CXR studies
360 PA CXR images
Congestive Heart Failure (CHF)
330 patients / ED stays
330 CXR studies
363 PA CXR images
Normal
359 patients / ED stays
359 CXR studies
360 PA CXR images

Figure 2.1: Flowchart detailing sampling process for the MIMIC-CXR-EGD dataset [1].

JPG dataset based on a range of criteria detailed in Figure 2.1, including data from the patient's stay in ICU contained within MIMIC-IV. Most notably, each of the images must have 1 (and only 1) of the following diagnoses: Congestive Heart Failure, Pneumonia or Normal. MIMIC-CXR-EGD is an interesting dataset as it differs from traditional data releases in that it does not have a specific task/classification goal in mind; instead it is designed to enable the further analysis of automated segmentation/classification models by encouraging researchers to incorporate the eye-gaze data into their analysis.

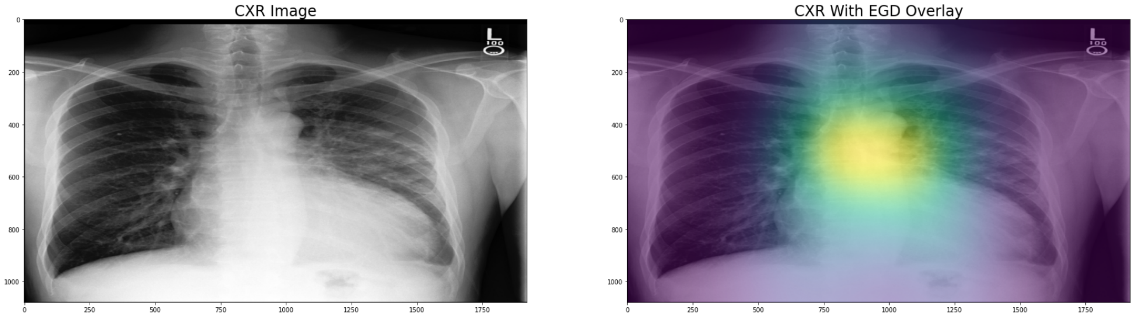


Figure 2.2: A random sample from the MIMIC-CXR-EGD dataset (left), and the same sample overlayed with the EGD heatmap (right).

As an example of how the eye-gaze data could be used in DL applications, the authors of the dataset also explored how the eye-gaze data could be incorporated into a multi-task U-Net model (Chapter 6.1) to improve the overall agreement between model saliency maps (generated using explainability techniques detailed in Chapter 2.2) and the expert's EGD. During training this multi-task U-Net model takes as an input both the CXR image and the EGD, and aims to both reproduce the EGD and produce an accurate classification label. A qualitative assessment of how this proposed architecture improved the agreement between model saliency heatmaps and the expert's EGD, however no quantitative comparison was provided (this forms the basis for the work in Chapter 6.1).

2.1.6 Codon Usage Dataset

Deep Learning is not limited to just EHR and medical imaging datasets, with it being increasingly used in bioinformatics [64] with applications including protein folding [65] and phylogenetic tree search [66]. The Codon Usage Dataset [47] is one of the only large, freely-available dataset that facilitates this type of research, consisting of the frequency of 64 different codons across more than 130,000 organisms. This dataset can be used for two separate classification tasks: **1)** predict the organism's phylogenetic Kingdom (from 5 distinct classes), and **2)** classify the DNA type of the organism (from 3 distinct classes).

2.1.7 Private Datasets

Due to their sensitive nature many healthcare-related datasets are not released publicly, mainly due to concerns around patient privacy [67]. Although this does raise concerns around the reproducibility of an author's work, it is somewhat of a necessary evil; for a dataset to be publicly released, a number of extremely onerous steps must be undertaken to ensure the dataset is fully anonymised, that no data can be linked back to an individual and that patient consent has been given (although it is important to note that the exact steps that must be undertaken will vary between different regions and institutions) [68]. This process costs a lot of time and money, resulting in many data owners keeping their data private. Additionally, under most forms of anonymisation that are currently codified in law (e.g. GDPR), much of the data's utility is lost when it undergoes anonymisation [69] and thus makes it less useful to researchers. All of this together means that, unfortunately, private medical datasets are still somewhat of a necessity in DL for healthcare research.

To support this thesis' research, I had access to one private EHR dataset from the King Abdulaziz Medical City located in the central and western regions of Saudi Arabia (KAIMRC) [28, 70, 71], which is used alongside publicly-available datasets to evaluate the techniques presented throughout the thesis. The dataset, which was originally collected to aid the development of ML models for diabetes prediction, spans from 2016 to 2018 and includes both patient demographics (e.g. age, sex, etc.) and lab results (e.g. cholesterol and eGFR levels) from a patient's last 6 hospital visits. The dataset contains 66,652 records of highly-detailed, pre-processed (i.e. missing data has already been handled) data. The KAIMRC data can be used for one of two tasks: **1)** using longitudinal data to predict which patients will go on to experience elevated HbA1c (a blood marker that is often used to diagnose pre-diabetes) levels or, **2)** a regression model to directly predict the HbA1c level of a patient. This task has been extensively studied on this dataset [28, 70], giving a number of baseline models that this thesis' techniques can be evaluated against.

2.1.8 Barriers to Further Adoption

As seen throughout Chapter 2.1, DL has enjoyed much success when applied to healthcare data with models being able to match (or sometimes even outperform) the performance of clinical experts. However, despite these successes, there are relatively few examples of these models actually being deployed in hospitals [16]. Although this can in part be attributed to the relative infancy of DL as a viable technique (and the need for healthcare models to undergo much more thorough testing than in other domains), there are also a number of fundamental issues with DL that prevents it from being more widely used [13, 72].

Arguably the largest barrier facing DL models are concerns around the trustworthiness of DL models [73]; both clinicians and patients alike must be able to trust the decisions made by a model. It is widely agreed that the best way to achieve this is to ensure that the models used are transparent and explainable - which is explored in much more detail throughout this thesis - but also to ensure that they are not prone to bias and can be held accountable for their decisions. Some of this requires medical policy to be updated too; for example, who should be held accountable if the decision made by a DL model is incorrect, and this results in harm to the patient? Currently, much more work must be undertaken by policy makers to address these issues [74] before we see further adoption of DL in healthcare. We explore current explainability techniques in Chapter 2.2, and the rest of this thesis is dedicated to how these methods can be used to overcome the barriers discussed in this section.

In order for DL models to be trustworthy, they must also be generalisable and robust. A large issue with medical DL models is the uncertainty around how well they will generalise to patient populations from different backgrounds to the one the model was trained on: for example, it is common for a DL model to be trained on data from only one (or perhaps a small number) of hospitals. However, this hospital may have a hugely different patient population to another hospital that wishes to also use the model [75] - this could result in the trained model performing poorly on the unseen data. Furthermore, there is an inherent degree of randomness present during the training of these models, which can affect the resulting model - this is explored in detail in Chapter 4 - and this type of randomness can affect the amount of trust non-experts will place in a model. This lack of robustness and generalisability is what makes DL models susceptible to adversarial attacks (such as those discussed in Chapters 2.4.1 and 2.4.3), which can also increase the uncertainty medical experts have in DL techniques.

The final significant issue facing DL practitioners in healthcare is patient privacy [76]. Modern data protection laws such as the European Union's General Data Protection Regulation (GDPR) [12] specifically protect citizen's health information, which means that anonymisation techniques must be used before the data can be used for DL purposes (unless patient consent has been given). Crucially, there is a large amount of uncertainty around how exactly these laws affect DL techniques, and where exactly the responsibility for data privacy lies: is it with the DL model developer, the hospital, or the data collector [77]? These are policy issues that must be sorted out on a case-by-case basis. Chapter 2.4 explores privacy from a deep learning perspective, highlighting current techniques that are used to improve a user's privacy when using DL models. In Chapter 6.2 I then look at how we can utilise model explainability to significantly improve the privacy of DL models.

2.2 Deep Learning Explainability

One of the main barriers facing DL practitioners in the healthcare domain is the lack of transparency offered by today's large, deep models [72]. This has led to the recent explosion in explainable deep learning [78], where researchers attempt to open up the *black-box* of DL. This research can be broadly split up into two distinct areas [79]: *post-hoc* (where techniques are developed to explain models after training) and *ante-hoc* interpretability (wherein new model architectures are built from the ground-up to be interpretable by humans). There is an inherent explainability-performance trade-off, particularly for ante-hoc explainability methods [80]: typically, the more accurate a model the more complex it is and hence it is also more difficult for a human to fully comprehend. This leads to an interesting philosophical argument: what actually makes a model explainable? For example, decision trees are largely regarded as a *white-box*, explainable model [17] but can become excessively large when used on complex data [81]. Even relatively small, well-tuned decision trees can arguably be difficult for a human to fully understand, with the number of terminal nodes increasing exponentially with depth. Is it truly possible for a human to fully understand the decisions made by such models?

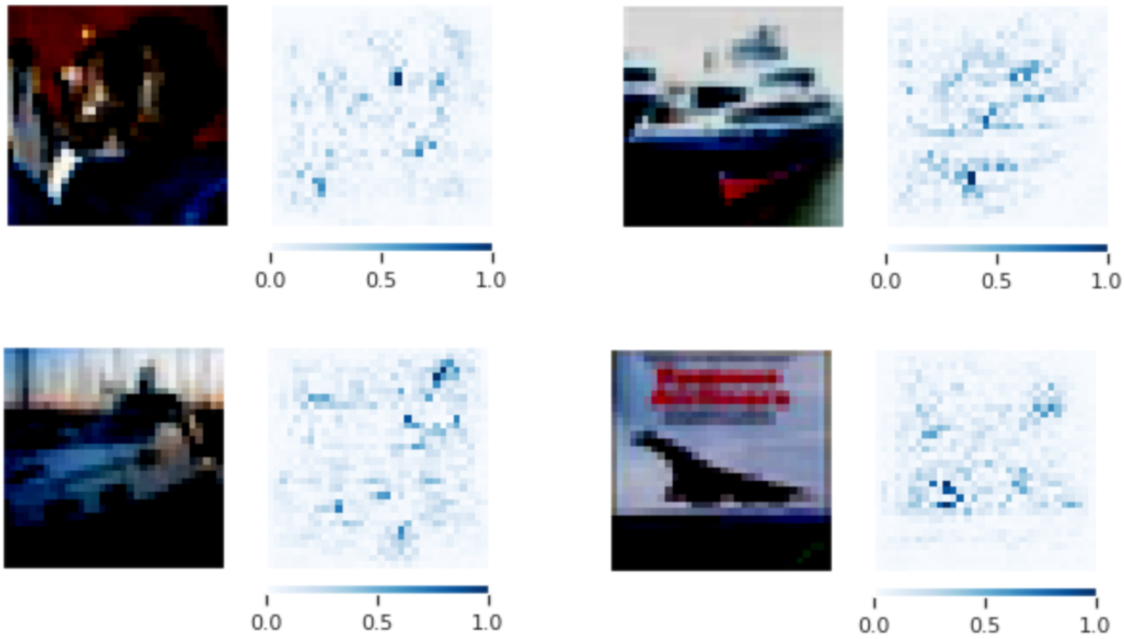


Figure 2.3: 4 samples from the CIFAR10 dataset with their associated absolute SHAP attributions from a small CNN trained on CIFAR10.

For this reason, for much of this work we focus on model agnostic post-hoc explainability techniques. These methods are able to explain the decisions made by any (or, in some cases, a wide-range of) deep learning architectures. Traditionally, they are applied after the model has been fully trained (although in Chapter 5 I will explore how they can be used during training) and produce *local explanations* - that is, they explain a specific decision (e.g. given a specific test instance $x \in \mathbb{R}^d$, explain the features of x which contribute most to $f(x)$) rather than explaining the whole model's behaviour [79]. Specifically, we want to generate a feature attribution map $E(x) \in \mathbb{R}^d$, where each $E_i(x)$ is the contribution of feature x_i to the black-box model's output $f(x)$; an example of the resulting attributions is shown in Figure 2.3. This is a large, well researched area with an ever increasing number of explainability techniques [78]. In the remaining part of this section, I explore some of these techniques, limiting ourselves to the most commonly used and influential techniques - a thorough review of the area can be found in [78].

2.2.1 Explainability Terminology

It is only recently that a precise taxonomy of terminology for explainable machine learning has been developed [82]. Prior to this, different studies may have used the same words but have meant different things [79, 83] - for example, what is the difference between an *explainable* technique and an *interpretable* one? For many these may mean the same thing, but there are subtle differences that should be defined. For clarity, the rest of the thesis uses the following definitions of explainability and interpretability, adapted from [17, 82]:

- **Interpretability** is the ability to be able to provide meaning from a technique to a human

- **Comprehensibility** is the ability of a model to present its learned knowledge in a form suitable for humans

- **Explainability** is the ability to have an interface between a human and a model which is both accurate to the model and comprehensible to the human. Note that this allows explanations to change based on the target audience - clinicians may require more detailed explanations than patients, for example

From these definitions, one can infer that an explainability technique is one that makes a traditionally black-box model interpretable. As such, while this thesis refers to improving the explainability of models, this is actually a proxy for improving the interpretability of models. The remainder of this section introduces a number of post-hoc explainability techniques that have been developed to provide interpretability to deep neural networks.

2.2.2 LIME

Local Interpretable Model-agnostic Explanations (LIME) [32] is an explainability technique that can be applied to any ML model. It aims to find an interpretable local surrogate model that is *locally faithful* around a given data point (i.e. the point to explain). A surrogate model is locally faithful if it accurately mimics the behaviour of the black-box model we want to interpret in the vicinity of the data point we are interested in (but outside of this area, the surrogate model need not correspond to the target model).

Given a black-box model f and some data point that we wish to explain, x, LIME first produces many perturbed versions of x. Given this new dataset of perturbed data points, LIME trains an interpretable model (e.g. a decision tree) which is weighted by the proximity of the sampled instances to the instance of interest. The weight each feature is given in this local model is then used as the feature's attribution value, as defined in Equation (2.1) where G is the class of all interpretable models, $\Omega(g)$ is the complexity of g (this metric is model dependent, e.g. for decisions trees it may be defined as tree depth) and π_x is the set of perturbed data points around x.

$$E(x) = \operatorname*{argmin}_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2.1}$$

LIME was one of the first widely used DL explainability techniques, mostly due to its ease-of-use and adaptability - the family of interpretable models G can be changed to suit almost any end-user, and allows the technique to be applied to any data modality. However, LIME has been shown to be extremely unstable [84], producing different explanations every time LIME is run. This is due to the random nature in which π_x is generated - every time LIME is run, a slightly different set of data points will be used to train the surrogate model. There is also uncertainty around how best to choose the size of π_x; the size of the dataset, and how far away we allow $x' \in \pi_x$ to be from the original data point x, can have a significant affect on the quality of the explanations and there is very little intuition on how to correctly set this hyperparameter [23].

2.2.3 SHAP

SHapley Additive exPlanations (SHAP) [85] is perhaps still the de-facto explainability technique used by DL practitioners. It is very closely tied to Shapley values from game theory [86], modelling each feature of the input as a player in a d-player game - a feature's attribution value is derived from their Shapley value, which is a measure of their contribution to the final value of the game (i.e. $f(x)$). Lundberg et al. [85] show that Shapley based methods satisfy three desirable properties:

- **local accuracy** - the explanation model matches the original model when provided the same input
- **missingness** - if any given feature is missing, then that feature must have an attribution value of 0
- **consistency** - if a model changes such that an input's contribution increases or stays the same regardless of other features, then its attribution should not decrease

In particular, it is proven that only the explanation model given in Equation (2.2) satisfies all three of these properties, where $\phi_i(f,x)$ is the importance of feature x_i to the model f, M is the number of features, $z' \in \{0,1\}^M$ is the coalition vector (which indicates which features are present), $|z'|$ is the number of non-zero entries in z' and $f_x(z') = E[f(z)|z_S]$ where S is the set of non-zero elements in z'.

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|(M - |z'| - 1)!}{M!}[f_x(z') - f_x(z' \backslash i)] \tag{2.2}$$

It is clear to see that it is extremely computationally intensive to compute Equation (2.2), with an exponential runtime due to the need to compute the change in model output for all possible permutations of features. To combat this, numerous approximation techniques have been proposed [85], from model-specific techniques such as TreeSHAP for tree-based models to model-agnostic methods such as DeepSHAP and KernelSHAP. These approximations allow pseudo-global feature attributions to be generated, where enough local explanations can be computed such that we can begin to understand the overall internal workings of a model. However, these approximations are not always appropriate (for example, KernelSHAP ignores feature dependence) and TreeSHAP has been known to produce unintuitive explanations.

It has also been shown that SHAP, along with some other explainability techniques, is susceptible to adversarial attacks [87].

2.2.4 Integrated Gradients

As the name suggests, Integrated Gradients (IG) [88] is a gradient-based approach for calculating feature attribution for neural networks. IG was an important step in the development of gradient-based attribution methods, as it was the first to not need instrumentation of the network. As a gradient-based method, the interpretation of IG is simple: the larger the absolute value of the gradient, the more influence that pixel has. IG is calculated through Equation (2.3), where $\frac{\partial f(x)}{\partial x_i}$ is the gradient of $f(x)$ over the ith dimension and x' is a "baseline" input which should be chosen such that $f(x') \approx 0$ (e.g. an image of all-black pixels satisfies this for image classification models).

$$\mathrm{IG}_i(x) = (x_i - x'_i) \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} \tag{2.3}$$

In practice Equation (2.3) is computed via an approximation using summations, which is both efficient and easily-implemented in most modern deep learning programming libraries. This efficiency and ease-of-use is one of IG's main advantages; IG is also easily applied to any DL model architecture and data modality, making it one of the most adaptable explainability techniques. However, the quality of the resulting explanation is highly dependent on a suitable choice of x' [89] and this leaves the technique open to both abuse and improper use by DL practitioners.

2.2.5 GradCAM

GradCAM (Gradient-based Class Activation Maps) [90] is an explainability technique that has become incredibly popular for generating explanations of CNN image classification models. GradCAM differs from IG as the gradients are back-propagated only to the final convolutional layer, rather than through the entire model; this results in an activation map that highlights important regions of the image rather than the per-pixel attributions provided by IG and other explainability techniques. Additionally, unlike IG (and some other methods) GradCAM does not require a baseline input thus removing one barrier to the adoption of the explainability technique. GradCAM can also be used to produce counterfactual explanations - that is, it can highlight regions of the image that, if changed, would cause the model to change its prediction.

One of the main disadvantages of GradCAM is that it can only be applied to the CNN architecture and as such is mostly limited to models that take images as their input. It has also been shown to sometimes highlight incorrect regions of the image [91] and be easily manipulated to produce any desired explanation [92]. However, it is still commonly used in many applications, particularly medical imaging [22, 93], due to its ease of use and highly-interpretable heatmaps.

2.2.6 Evaluating Explainability Techniques

As seen in the above sub-sections there have been numerous explainability techniques proposed, each with their own advantages and disadvantages. This can make it difficult to know which method should be used for which task, meaning the evaluation of explainability techniques is vital. This is a deceptively difficult task [17], with there being no consensus on the best way to evaluate an explainability technique: is it better to focus on how faithful produced explanations are to the model? Or should we focus on how easily the explanation can be interpreted by a human?

There are three main qualities that current metrics are currently designed to measure:

- **Faithfulness:** does an explanation match the inner workings of the model?
- **Understandability:** are the explanations compact/simple enough that a human can understand them?
- **Stability:** are the explanations robust to small perturbations?

A number of metrics have been proposed that aim to address these questions (and measure these qualities), and one should use a wide range of them to fully understand the quality of an explainability method. This thesis is largely focused on the *faithfulness* and *stability* of an explanation; in particular, the explainability metrics have been chosen to ensure that the methods introduced in this thesis are still able to generate explanations that are as close to the underlying model as possible. Indeed, this is especially important in Chapter 4, where explanations themselves are introduced as a model quality metric. Explanation infidelity [94], is one measure of how faithful explanations are to the model (i.e. how much does an explanation change when the input is slightly perturbed). Infidelity is defined in Equation (2.4), where $\boldsymbol{I} = x - x_0$ and x_0 is some baseline input; it can be seen as the expected error between the explanation and the difference in model outputs when the input is perturbed.

$$\text{INFD}(E, f, x) = \mathbb{E}_{\boldsymbol{I} \sim \mu_I}\left[\left(\boldsymbol{I}^T E(f, x) - (f(x) - f(x - \boldsymbol{I}))\right)^2\right] \tag{2.4}$$

Similarly, explanation sensitivity-max [94, 95] is a measure of how much an explanation changes when the input is slightly perturbed:

$$\text{SENS}(E, f, x, r) = \max_{\|x' - x\| \leq r} \|E(f, x') - E(f, x)\| \tag{2.5}$$

where r is the radius for which the perturbations will be sampled within. This metric is useful as it can be quickly approximated via Monte-Carlo sampling. Both infidelity and sensitivity-max are used to measure how much explanations change compared with how much the DL model output changes (when an input is slightly perturbed): ideally, one wants these metrics to be as low as possible, as this would show the explanations are faithful to the original model. However, faithful explanations are not sufficient for *good* explanations: they must also be interpretable by humans (ideally to non-DL experts as well) and actually highlight regions of the input that the model is finding "important" (it is conceivable that a given explanation method may have low infidelity/sensitivity, but that the feature attributions actually do not correspond to the features used by the model).

Explanation accuracy [17] is defined as the accuracy of a DL model when trained on explanations rather than the original input; it attempts to measure how well the explanations match the features used by the model (although it should be noted that this is extremely dependent on the performance of the explained classifier). Evaluating the comprehensibility of the explanations is a much more difficult task; it is difficult, if not impossible, to quantitatively measure such a property as it is inherently related to the explanation's audience. This type of explanation quality can only be evaluated via qualitative, usually subjective, measures such as trust and confidence [96]. Although this type of human-centred evaluation is critical when deploying DL techniques in practice, it is costly in terms of both time and money, meaning it is only really performed on small scales and towards the end of a model's development life cycle [97].

For image data, one interesting way of evaluating explanation quality is to compare it with the Eye-Gaze Data (EGD) of an expert who was tasked with analysing the image. While datasets with such ground-truth explanations are limited, those that are available allow for a powerful analysis of deep learning models. For example, [1] do this on a small set of CXR images: they asked a trained radiologist to diagnose the x-rays, tracking their eye movements, and compared this data with the GradCAM explanations from a model tested on the same set of images. Of course, this method does not necessarily evaluate the explanation technique itself, and instead actually evaluates the quality of the model's learned features. Although an argument could be made that this is not exactly a fair comparison - the radiologist was analysing a high-resolution image, whereas the model is given a scaled-down version (which, as such, may have fewer visible features) - in the case of these experiments, the model explanations are so far out of alignment with any noticeable, important part of the image (e.g. the general chest area) that we can conclude the deep learning models are likely learning spurious correlations. As we will see in Chapter 6.1, this work shows that the learned features of classical DL models is poor (for CXR diagnosis, at least) and can be significantly improved upon by utilising some of the explanations techniques seen in this chapter during training. Indeed, the remaining chapters of this thesis all focus on how the methods introduced in section can be applied in novel ways to improve the robustness and overall quality of DL models (e.g. improving features such as privacy and security), and how this could increase clinician's trust in models when applied to healthcare tasks.

2.3 Deep Learning Model Robustness

As DL becomes ever more popular, we are becoming increasingly aware of its limitations. In many applications, DL models have been shown to biased decisions [8,98], or be susceptible to shortcut learning (wherein a model learns to "cheat" and use spuriously correlated features rather than those that are causally related to the target) [18]. The use of spurious correlations (that is, features that are correlated with the target, but not the two are not causally related [99]) is worrying, as it leads to models being more susceptible to privacy leaks, erodes trust and more generally results in models that are not as robust as they could be [18, 100]. A particularly egregious example of this from the healthcare domain can be seen in [101], which studied a well-performing pneumonia detection model that took CXRs as an input. While the model achieved a respectable AUROC of 0.773, the authors showed that the model had actually learned to detect regions of the CXR that indicated which hospital the image was taken in (due to different machines and setups being used in different hospitals, CXRs can differ ever so slightly). As some hospitals had higher rates of pneumonia than others, this was a good proxy for pneumonia prediction; however, it is clear to see that the model would fail if deployed in practice. In another more recent example, despite widespread claims of success in applying ML to COVID-19 tasks [102], many of these models succumb to numerous pitfalls such as making spurious correlations or being unable to generalise [103, 104]. The lack of robustness to issues like this is one of the main barriers facing further adoption of DL in healthcare. It is believed that a lot of these issues are down to our lack of understanding of how deep neural network training actually works [105] - in this section we briefly introduce some of these issues, as well as some techniques that have been developed that aim to alleviate them.

2.3.1 Model Generalisation and Underpsecification

One major concern for DL applications is that, in many real-world scenarios, there is a significant difference between the data used for training and the data the model will be applied on in practice - often, there is such a significant shift that the underlying causal structure of the data will be different [10]. Even though some DL models have been shown to generalise well to real-world datasets [36], it is still not fully understood how or why this occurs; for example, how is it SOTA vision models are able to converge and generalise, even when trained on unstructured noise [11]? Couple this with recent work that suggests deep neural networks are immune to the bias-variance trade-off, with networks not exhibiting the classical U-shaped test error curve (Figure 2.4) [106, 107], and the picture becomes even more complex. Specifically, overparamerterised networks (i.e. networks where there are many more learnable parameters than training samples) are able to learn noisy datasets well as they can simply ”remember” all samples in the training set. Typically, one would expect a model that has learned in this way to achieve poor generalisation performance (it has not learned any important features, simply the entirety of the training set) and yet recent studies have shown that some such models are indeed able to perform well on a fresh testing set [108]. This is a surprising result, and is not yet fully understood - it is widely agreed that more theoretical work must be carried out to investigate this phenomena [109].

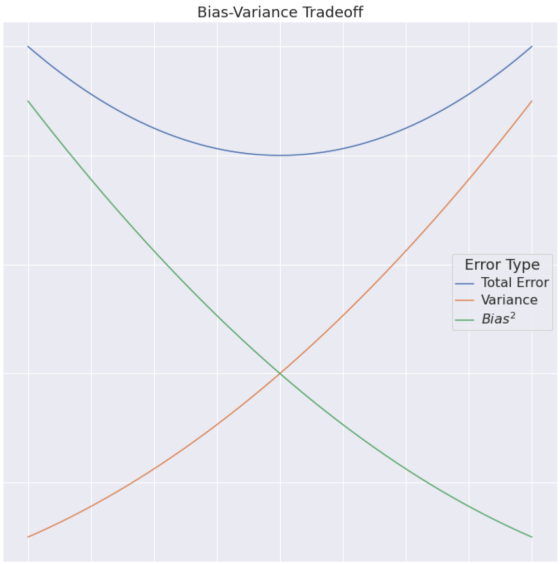


Figure 2.4: The *expected* U-shape bias-variance trade-off curve for machine learning models.

This lack of theoretical understanding can severely affect the development of DL models. Clearly, these issues need to be addressed before DL models have any hope of being widely adopted in sensitive scenarios such as healthcare.

Moreover, shortcut learning [18] (or decision rules that work well on standard benchmarks but fail to generalise to more complex situations) has recently been shown to be prevalent across many different machine learning domains. This means that we can no longer assume that a model's performance on one dataset indicates that it is well-suited to the task at hand [110] and suggests that many models may not be as good at generalisation as was once thought [35].

Extensive work has gone into trying to explain these phenomena [20, 99, 111], with many studies attempting to explore how models train and learn variations in data, yet still no consensus has been reached. This thesis argues that the lack of understanding of exactly how these deep learning models work [38] and generalise is ultimately preventing us from addressing the aforementioned issues. Understanding how the stochastic nature of the training process affects what properties of the data is captured by the model is fundamental. Chapter 4 explores how DL explainability techniques can be used to further explore DL model training, begin to uncover how stochasticity affects training, and use these techniques to question a model's robustness. Then, Chapter 5 extends this work to propose an entirely new training algorithm that addresses some of these questions.

2.3.2 Modern Neural Network Architectures

Modern deep learning architectures are not only designed to achieve higher levels of task accuracy, but also increased robustness, generalisation and transparency. For example, GaborNets [112] are a class of CNNs that utilise Gabor Filters [113] with learnable parameters in the place of the first convolutional layer. Gabor Filters are suggested to closely mimic the behaviour of a human's vision cortex, detecting lines with specific direction, spatial frequency and scale, and it has been shown that GaborNets are able to more accurately capture orientation information (in the first layer) than traditional CNNs. While GaborNets do see a slight reduction in performance on baseline image classification tasks than traditional CNNs, it is argued that having a network that closely models human behaviour is beneficial in some sensitive applications, such as healthcare. However, as we will be explored in Chapter 6.1, this is not necessarily the case.

Ensemble models [114, 115] have long been proposed to improve model performance. Ensemble methods take a number of models (also called the ensemble's sub-models) trained on the same task/data and combine the output of all of these models to produce a final output (shown in Figure 2.5), with the idea being that the error contained in any one of the sub-models will be compensated for by the other sub-models. Ensemble techniques are typically used to improve upon the performance of a baseline model, with ensemble architectures reaching the top of the performance leaderboards for many DL tasks [116]. Due to their increased complexity, ensembles are inherently less explainable than their regular counterparts [17] - however, this issue is something that will be addressed via the introduction of the novel Deep Explanation Ensemble technique in Chapter 5.

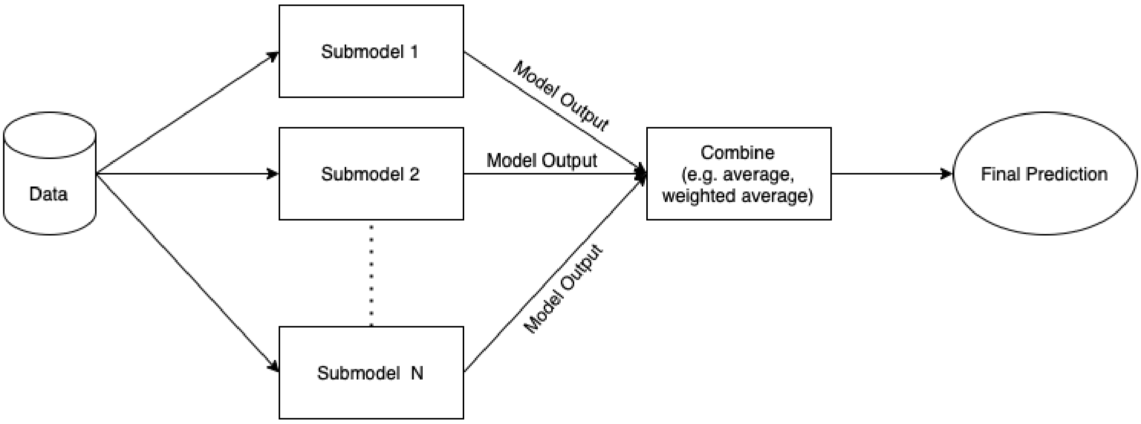


Figure 2.5: A basic ensemble modelling framework.

There are several reasons why ensemble techniques are able to achieve higher levels of performance than a single model alone [117, 118]. Firstly, it is well known that there are typically many hypotheses (i.e. set of weights) reachable by a DL model that perfectly fit its training data (with this being particularly true when the size of the training data is small) - this is precisely why DL is prone to overfitting (i.e. remembering) the training data. This leads to a model achieving perfect performance on the training set, but poor performance on anything outside the training data. With an ensemble model, however, this issue is somewhat alleviated as each of the sub-models will reach a slightly different hypothesis and so the chances of them all overfitting to the same set of data is smaller. Secondly, ensembles allow a wider range of models to be explored. By increasing the search space during training, we increase the likelihood that we will find a set of weights that accurately model our data.

Furthermore, recent ensemble architectures have also been shown to increase robustness to malicious inputs [115] and improve model generalisability [119]. In particular, Hyper-deep Ensemble Models [25] have been shown to further increase the generalisability of the model. These extend the idea of traditional ensemble models, which consist of models with different weights, to also include models which have been trained with different hyperparameters. This is a simple addition to the training procedure, where not only are the sub-models trained with different random initialisations (as with traditional ensemble techniques) but a random search across the hyperparameters is also performed. These models are shown to outperform traditional ensemble models on the CIFAR10/100 [120] benchmark datasets, as well as showing they are more robust to data corruptions. However, as explored in Chapter 4, they still produce inconsistent results when retrained - an issue which Chapter 5 aims to address.

2.4 Privacy in Deep Learning

By its very nature, DL requires *a lot* of data and this naturally raises privacy concerns, particularly when private information such as a person's healthcare records are involved; people can be reluctant to share their data for use in a DL model when they are unsure exactly how it is used and how securely it will be stored [7]. Furthermore, neural networks have been shown to be susceptible to host of adversarial attacks, ranging from attacks that can trick a model into making a certain decision to membership inference attacks. This section, explores some of these attacks and current defences against them. Chapter 3 investigates how off-the-shelf explainability techniques can be used to protect against some of these attacks, and Chapter 6.2 explores how a new type of model training create models that are inherently robust to such attacks.

2.4.1 Adversarial Attacks

Adversarial attacks are a concerning weakness of DL models. It has been shown that it is possible to construct an adversarial input for a DL model by taking a sample from the dataset and adding some carefully constructed noise to it - this adversarial input then results in the model outputting the incorrect prediction [121, 122]. Importantly, the changes between original input and the adversarially perturbed input can be so small that it is unnoticeable to the human eye. There are a number of different ways of constructing these attacks, with them most commonly being applied to image data - although specific attacks have been developed for other modalities. This section explores attacks that may be relevant in healthcare scenarios (e.g. those on medical images and EHR data), although it is worth noting that many of the attacks can also be applied outside of the medical domain (e.g. on any image, or any time-series data).

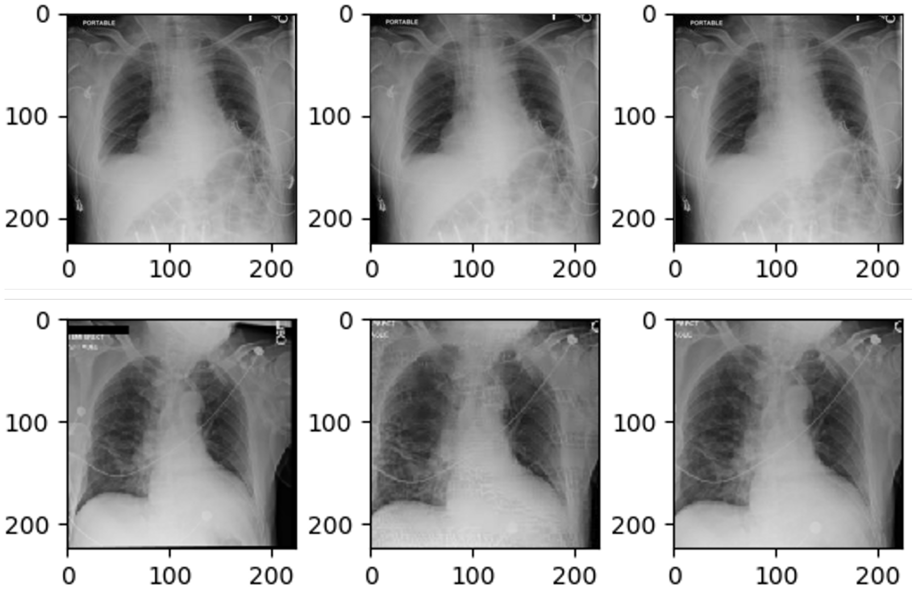


Figure 2.6: Random adversarial examples generated on the MIMIC-CXR dataset. Images on the left are the original images, the middle have been generated via PGD, and the right via C&W.

Medical data has a number of properties that makes it more susceptible to adversarial attacks than other data modalities [123]. In particular, medical imaging is a highly standardised domain - images are typically collected under a very specific set of circumstances, meaning there is not much variation between images. This makes it easier to generate adversarial attacks: they don't need to be able to adapt to different setups (e.g. lighting, orientation) like they do when applied to traditional image classification tasks. Additionally, many medical images have an ambiguous ground truth - it is not always clear, even to medical experts, what the correct diagnosis may be. If an attacker were to target these types of images, it makes it easier to construct examples which humans correctly label but fool the DL model.

Both of these points result in traditional adversarial attacks being applicable to medical images [123]. Projected Gradient Descent (PGD) [124] generates adversarial samples by using gradient descent to maximise the model's loss whilst keeping the size of the perturbations small (typically within some $\mathcal{L}_\infty$ ball). A set of three more advanced attacks is proposed in [125] that use a similar approach but, instead of using projected gradient descent, the optimisation problem in Equation (2.6) is solved. Figure 2.6 shows examples of two adversarial attacks applied to medical imaging data. This attack is shown to be more effective than the standard PGD attack, being able to bypass even robust DL models. The authors of this attack show that it never fails to produce an adversarial sample when tested on models trained on MNIST, CIFAR and ImageNet.

$$\begin{aligned} \text{minimise} \quad & \|\delta\|_\infty + c \cdot f(x+\delta) \\ \text{such that} \quad & x + \delta \in [0,1]^n \end{aligned} \tag{2.6}$$

Alongside these attacks, which are applicable to any data modality, some attacks are designed with specific applications in mind. For example, the Longitudinal AdVersarial Attack (LAVA) [126] is designed to create adversarial samples for time-series data such as Electronic Health Records. EHR data proposes a unique challenge in that it can be easier for a human to detect perturbed features - it is easier to notice a change in one or two EHR features than it is in slight changes to pixels in an image. LAVA combats this by using saliency scores to determine which features should be perturbed: features with high saliency scores aren't changed as much, as it is assumed a clinician will focus closely on these features and thus be more likely to notice an attack. On a private EHR dataset, the authors of LAVA showed that it was able to reduce model performance (AUROC) from 0.5 to 0.08, both beating the performance-drop of PGD and whilst also perturbing fewer features (hence making the attack harder to detect).

2.4.2 Defences Against Adversarial Attacks

As new adversarial attacks are developed, so must new defence mechanisms. A simple solution is to train a binary classifier on normal dataset samples and adversarially perturbed samples with this method being shown to perform fairly well on the CIFAR10 dataset, detecting at least 75% of adversarial images (overall performance is dependent on the type of attack used) [127]. However, this method requires a large number of adversarial samples to be available, which is not only time-consuming to generate but is also often impractical: in a real-world scenario, it is impossible to know what type of adversarial attack will be used, and it is impossible to train a binary classifier on all possible attacks.

It has also been shown that it is possible to detect noisy adversarial samples using Bayesian uncertainty estimates and density estimates of the model's final hidden layer [128], with both techniques complementing each other to detect adversarial samples the other cannot. By estimating the sub-manifold of data that correspond to a class c, it is possible to detect samples which then lie outside this manifold - these samples have likely been adversarially perturbed. Then, by calculating the Bayesian uncertainty (available in networks that utilise dropout) we are able to detect samples which the model is highly uncertain about - again, these are likely to be adversarial samples. This pair of techniques has been shown to be able to detect attacks on the CIFAR, SVHN and MNIST datasets with high accuracy, as well as working well on medical images [105]. However, experiments in Chapter 3 will show that they do not perform very well on EHR data and are extremely model-dependent.

ML-LOO [129] is an adversarial attack detection method that uses the Leave One Out (LOO) explainability technique to detect perturbed samples. When using LOO, feature attribution is calculated via the reduction in the probability of the selected class when the feature is masked/removed. ML-LOO has been shown to outperform all other classical adversarial attack detection methods, however it is extremely computationally intensive: for every combination of features, the model must be ran. Furthermore, like all other techniques previously discussed, this technique requires retraining as new adversarial attacks are created.

2.4.3 Membership Inference Attacks

In addition to adversarial attacks, it has been shown that deep learning models are susceptible to memorising training data even when they have generalised well [130], which leaves them open to a number of different Membership Inference Attacks (MIA) [131, 132]. Given a DL model M, and input x, the goal of a MIA is to determine if x was included in the training of the model. These attacks can be used to infer information about x - for example, their relationship to the goal of the classifier M. Susceptibility to these attacks is a large privacy concern, with the US National Institute of Standards and Technology specifically classifying a successful MIA as a privacy violation [133].

Broadly, membership inference attacks can be separated into two groups: black-box and white-box attacks. White-box attacks assume that the attacker has full access to the target model. On the other hand, black-box attacks only allow limited access to the model (usually some form of its outputs) - black-box attacks are the most well-researched MIA type, as they more closely mimic real-world attacks [131]. Standard DL models have been shown to be highly susceptible to this type of attack, with membership inference being able to be inferred from the model output's alone [132], through the creation of "shadow models". These shadow models are designed to mimic (as closely as possible) the behaviour of the target model. This allows the attacker to be able to train a black-box binary classifier on these shadow models, which can then be transferred to the target model (all the while not requiring access to the target model). It has been shown that similar attacks are viable even when the target model is able to generalise well [134].

While all MIAs based on the shadow model architecture require access to the target model's confidence score, there are a growing number of attacks that require access to the predicted labels only [135]. These attacks work by perturbing the input x and observing how this affects the model's output - these observations are then used to determine how close to the decision boundary x is, with the idea being those samples which were in the training set will be further away from the decision boundary. These attacks have been shown to work on a wide range of model architectures [135]. Chapter 6.2 analyses how new training techniques introduced in Chapter 5 create models that are extremely robust to membership inference attacks, and investigate why this is the case.

2.4.4 Federated Learning and Differential Privacy

Federated Learning (FL) is an alternative deep learning training paradigm that is designed to improve users' privacy. In the FL setting, a central coordinator and multiple distinct remote parties contribute to the training of a global model in such a way that the remote parties' data remains private [136, 137]. Traditionally, with deep learning models this is achieved through Federated Averaging [138] wherein each remote device trains its own version of the global model (with its own private data), and the global server collates all private models into one global model; in this setup, a user's data never leaves their own device as only the final model is shared. This has applications not only in situations where there are many individuals each with their own private data (e.g. predictive text [139]), but also in scenarios where multiple data processors have data on many individuals that they want to aggregate into a single training set (e.g. hospitals, each with their own patient population [24]). Although FL allows the training of a ML model without clients explicitly sharing their data, it alone is not enough to provide sufficient privacy protection and instead must be used in conjunction with additional privacy-preserving methods [140]. For example, it has been shown that membership inference attacks are viable in the FL setting [141] as well as other attack methods that utilise unique properties of federated models [142].

One of the main defences against membership inference attacks (and many other privacy-related issues) in both traditional and federating settings is the use of Differential Privacy (DP) during training [140, 143, 144]. For a model to satisfy (ϵ, δ)-DP it must follow Definition 1; that is, it is formally guaranteed that for two datasets $\mathcal{X}, \mathcal{X}'$ that differ by exactly one sample, two models trained on these datasets will produce statistically similar results (i.e. the similarity of the results are bounded by ϵ).

Definition 1 (($\boldsymbol{\epsilon}, \boldsymbol{\delta}$)-DP) *A randomised model $f : \mathcal{X} \rightarrow \mathcal{Y}$ satisfies (ϵ, δ)-DP if, for any two adjacent inputs $x, x' \in \mathcal{X}$ and any subset of outputs $S \subseteq \mathcal{Y}$ it holds that*

$$\mathbb{P}[f(x) \in S] \leq e^{\epsilon}\mathbb{P}[f(x') \in S] + \delta$$

An algorithm that satisfies DP protects both the content and the output of the model, though often at the cost of model performance and increased computation. In the context of Deep Learning, DP is typically achieved via the DP-SGD algorithm, which is a differentially-private version of Stochastic Gradient Descent (SGD) [143]. Briefly, DP-SGD works by injecting noise into the gradient at each batch (providing the privacy required by DP) and clipping the norm of each gradient (ensuring that one training sample does not have an overwhelming influence over training, which could result in a privacy violation). It has been proven that differentially-private algorithms provide an upper bound on the impact of MIAs [19,145], however when applied to real-world settings many assumptions required by DP no longer hold [99,146] leading to over-estimated levels of privacy [147]. In fact, there is such a significant difference between the theoretical analyses and real-world applications of DP that it has been shown that it is not necessarily a sufficient defence against MIAs and that deeper analyses are needed when studying MIAs on differentially-private algorithms [53]. Chapter 6.2 will further explore how, under real-world assumptions, DP and FL do not significantly reduce a model's vulnerability to Membership Inference Attacks, and investigate how explainability techniques such as those discussed in Chapter 2.2 can be used to infer training set membership. Chapter 6.2 will then go on to show how the training algorithm proposed in Chapter 5 can be utilised in a federated setting, and explore its robustness to such attacks.

CHAPTER 3

Attack-Agnostic Adversarial Attack Detection using Explanations

As seen in Chapter 2.1, applications of machine learning in healthcare have shown great success. However, in Chapter 2.4.1 we also saw that these models are highly susceptible to adversarial attacks and that examples of such attacks are effective when applied to medical data such as EHR [126] and imaging data [148]. The presence of adversarial attacks is of particular concern in the medical domain as it would be unethical to deploy a machine learning model to clinical practice if it is considered vulnerable to such malicious attacks, even if the likelihood of an attack is low [148].

Healthcare DL models are at particular risk of adversarial attacks [72, 105, 148]. Fraud is already pervasive in the US' healthcare economy, with institutions systematically inflating costs and physicians billing for the largest amount possible [148, 149] and, with machine learning algorithms likely to be used for medical decisions in the near future [150], adversarial attacks on ML models will be a new avenue for fraud to occur. The pharmaceutical and medical device markets are also domains where adversarial attacks on medical machine learning systems are a risk. The large amounts of money involved in these markets (the median revenue for a single cancer drug is estimated to be $1.67 billion [151]) combined with the increasing number of drug/device approval decisions being made based on digital surrogates for patient responses (for example, in medical imaging [152]) means that extremely valuable decisions are being made by machine learning algorithms and as such are a likely target for adversarial attacks.

As Chapters 2.1.8 and 2.4.1 explains, medical DL models are particularly vulnerable to these attacks, and medical and policy experts alike agree that DL models must be robust to such attacks in order for them to be deployed into real-world scenarios. As part of this effort many techniques have been developed to combat these attacks, ranging from techniques designed to detect adversarial samples to modified training techniques that aim to make models that are inherently robust to adversarial inputs. However, as discussed in Chapter 2.4.2, this is far from a solved problem, especially in the medical domain - there are very few adversarial attack defences designed for time-series EHR data, with other techniques yet to be tested on medical data, and the majority of the best-performing techniques require retraining when a new type of attack is discovered. This chapter further explores this issue, showing that existing adversarial attack defences fail to sufficiently protect healthcare-based DL models against perturbed samples. We also introduce a novel method that utilises off-the-shelf explainability techniques to detect adversarial samples, and show that this detection technique is attack-agnostic. The contributions of this chapter are as follows:

- It proposes the first adversarial sample detection technique that works effectively with EHR data
- It proposes a novel and simple method for detecting adversarial attacks using explainable techniques and demonstrate that it beats the state of the art on both medical imaging and EHR data despite the sparse, temporal and high-dimensional nature of the data
- It shows that the method is model agnostic and will support any machine learning model, unlike previous techniques

- By framing the adversarial detection as an anomaly detection problem, the approach is able to generalise to any attack type without the need to retrain

3.1 Methodology

As adversarial attacks change parts of the input, I hypothesise that ML models place more importance upon these perturbed sections of the input when passed an adversarially perturbed sample - it then follows that one could use explainability techniques to detect when these regions are activated. I introduce novel solutions that utilise SHAP values to detect adversarial attacks and demonstrate that it works on both medical imaging and EHR data. The proposed solutions consist of both fully- and semi-supervised methods, and exploits the differences between the distribution of SHAP values of genuine and perturbed samples in order to accurately detect adversarial samples. Furthermore, as SHAP values are consistent across the entire genuine dataset, the semi-supervised solution is able to generalise to adversarial attacks generated by alternative (i.e. unknown) methods without the need for retraining.

3.1.1 Datasets and Classification Models

Throughout this chapter, all experimentation is performed on 2 EHR datasets for experimentation: MIMIC-III [153] and Henan-Renmin[1], and 1 medical imaging dataset: MIMIC-CXR [45]. The Henan-Renmin dataset contains records from 110,300 patients, however with significantly fewer features than MIMIC-III; 62 features per patient comprised of basic examinations and clinical tests. The class label for each record is a combination of three possible diagnoses: hypertension, diabetes and/or fatty liver. For further explanation of the two MIMIC datasets, refer back to Chapter 2.1.

RETAIN [54] is a state-of-the-art model designed specifically to work with EHR data. The model aims to mimic typical physician practice by inspecting EHR data

[1] http://pinfish.cs.usm.edu/dnn/

in reverse-time order, such that more influence is given to more recent visits when making the final classification. In order to provide interpretable results, RETAIN has a two-level neural attention model that first detects key visits and then detects the key diagnoses from these visits. I train RETAIN on the MIMIC-III dataset. This results in an accuracy of 81% when predicting patient mortality. To ensure that my adversarial attack detection method adapts to different datasets, I also train the RETAIN model on the Henan-Renmin dataset to predict hypertension, with an accuracy of 73%. Hypertension is chosen as it is the most prevalent single label, providing mostly balanced classes. This is the same base task as originally experimented on in [54], allowing the same set of hyperparameters to be used: all hidden layers have size 128, dropout is performed with a probability of 0.6 and the L_2 regularisation coefficient is set to 0.0001. Both of these results show that the baseline models are able to achieve levels of performance that would be expected when compared with the results of other studies [54, 153].

In addition to EHR data, I also evaluate my proposed techniques on medical imaging data. Specifically, evaluations is focused on CXRs with the Cardiomegaly label, reducing the 14-label multi-label classification problem from MIMIC-CXR to a simpler binary classification problem. To do so, first CheXpert is run on the radiologists' reports to extract the diagnosis which results in 14 labels, each of which is classified as either a positive mention, a negative mention or an uncertain mention. Following the methods of [27], all uncertain labels are treated as positive mentions, and images without any Cardiomegaly labels are removed (if these were included, it would be difficult to apply a label to them without making further assumptions). The Cardiomegaly label was chosen as the evaluation task as this is both a common diagnosis and provides a balance between positive/negative labels with a low number of uncertain mentions. I fine-tune Densenet-121 [154] (pre-trained on ImageNet [61]) on MIMIC-CXR, based on the method presented by Rajpurkar *et al.* [40], to predict a diagnosis of Cardiomegaly, achieving an accuracy of 82%. This model is fine-tuned with the following hyperparameters, found after performing a grid-search on the validation set: batch-size of 8 (limited due to compute resources available), learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Similarly to the EHR models, this is inline with the accuracy achieved by similar studies [40] and confirms that the baseline models are properly trained.

3.1.2 Adversarial Sample Generation

State-of-the-art adversarial sample generation techniques introduced in Chapter 2.4.1, that are known to be successful on medical data, are used to generate adversarial samples of the three evaluation datasets. LAVA [126] is used for the two EHR datasets. Both RETAIN trained on MIMIC-CXR and RETAIN trained on Henan-Renmin see a significant reduction in accuracy, as shown in Table 3.1. The reduction in accuracy is similar to that reported in [126].

Table 3.1: Table showing accuracy of the models on the original and adversarial attack datasets. As PGD necessarily performs perturbations until the sample is classified incorrectly, the MIMIC-CXR model must achieve an accuracy of 0% on the adversarial set.

Model	**Accuracy original data**	**Accuracy adv. data**
MIMIC-III RETAIN	81%	43%
Henan-Renmin RETAIN	73%	44%
MIMIC-CXR Densenet121	82%	0%

Projected Gradient Descent (PGD) [124] is used to generate the CXR adversarial samples. As can be seen from Figure 3.1, the attacks generated by PGD use perturbations so small that they are impossible to detect via the human-eye, and Table 3.1 shows that PGD successfully produces adversarial samples that are able to mislead the model into making an incorrect classification. In order to test my method's ability to generalise to different attack types, I use the attack method proposed by Carlini & Wagner [125] (C&W). Unlike PGD which uses L_∞ norm to measure perturbation size, C&W uses the L_2 distance metric to produce a second set of adversarial samples for the MIMIC-CXR dataset. These two approaches have been chosen as they perturb the images differently and hence allow the adversarial attack detection technique's ability to generalise to be tested.

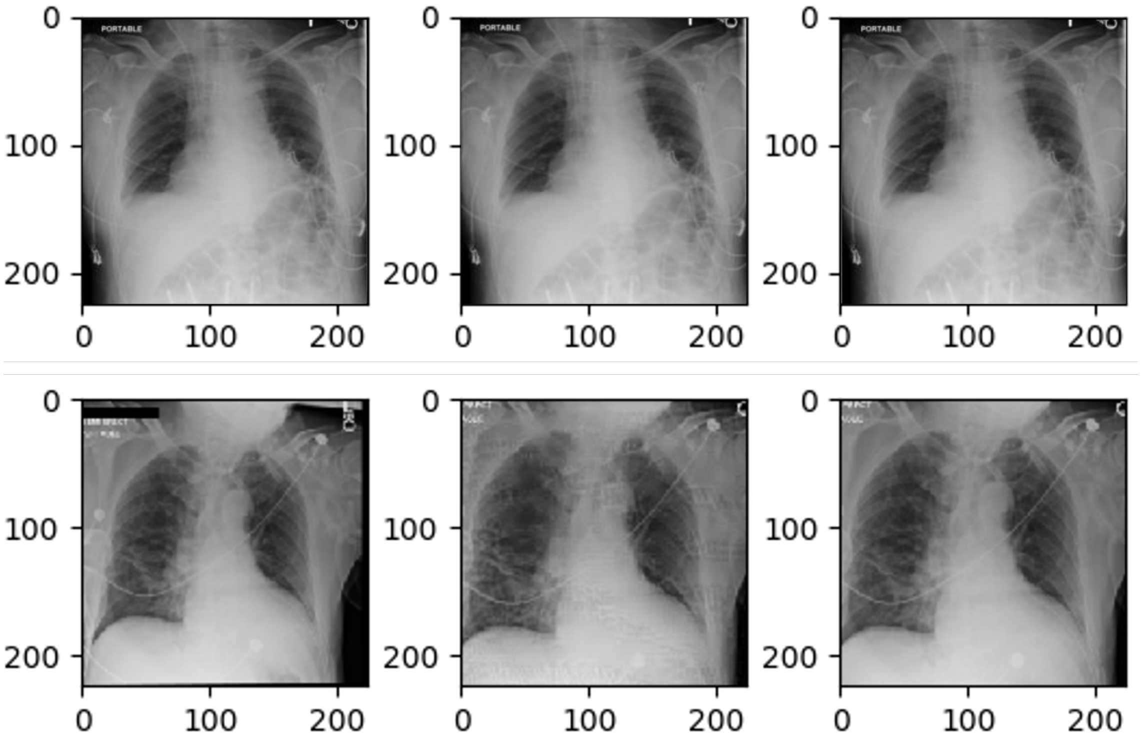
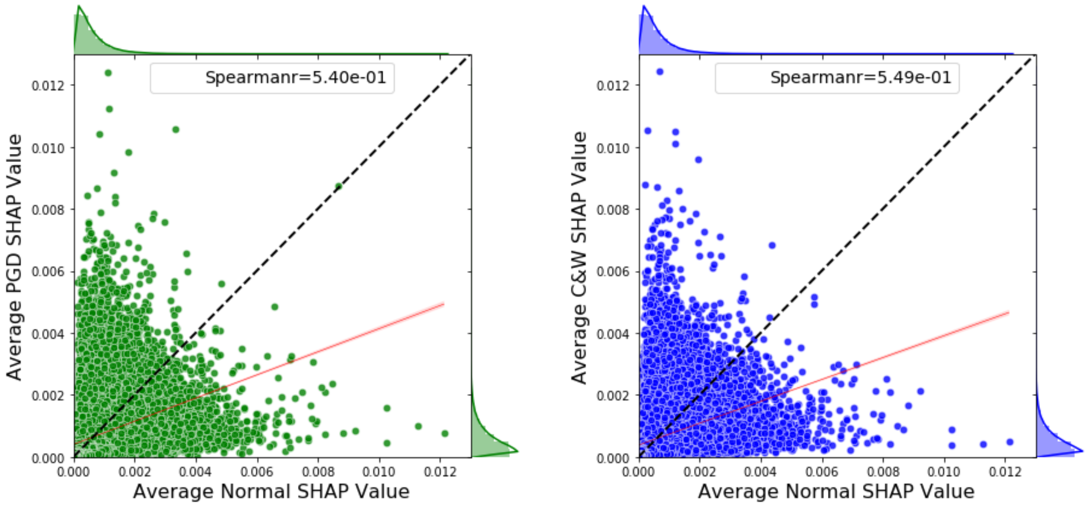


Figure 3.1: Random adversarial examples generated on the MIMIC-CXR. Images on the left are the original images, the middle have been generated via PGD, and the right via C&W.

3.1.3 Adversarial Attack Classification

As adversarial attacks subtly change small parts of the input, I hypothesize the SHAP values for an adversarial sample will be different than those for a genuine sample. This hypothesis is confirmed by Figure 3.2, which shows how PGD and C&W affect the distribution of SHAP values compared to the SHAP values of genuine data (correlation is low between the two with most values away from the perfectly correlated linear line). This demonstrates that although adversarial attacks methods aim to make the minimal feature perturbations possible, they still greatly impact the distribution of the explanation of the model predictions. Figure 3.2 also demonstrates that the PGD and C&W attacks perturb the samples differently.

In order to quantify the importance that our models place on different parts of their respective inputs, SHAP (Equation (2.2)) values as calculated by GradientSHAP [85] are used. SHAP values reflect the contribution of each individual feature to a model's prediction, which is important when only a small number of



(a)

(b)

Figure 3.2: Figures showing the average absolute importance of each feature in the original MIMIC-CXR dataset, calculated using SHAP values against the adversarial samples. (a) Scatter plot of the SHAP values of PGD adversarial samples on the Y axis against the SHAP values of original sample on the X axis, the dashed line represents the ideal line while the red line is the linear fit. The histogram of each axis is plotted. The Spearman Rank correlation value is reported.(b) Scatter plot of the SHAP values of C&W adversarial samples on the Y axis against that of the original set on the X axis.

features are changed under perturbation during the adversarial attack. SHAP values for the unperturbed (genuine) dataset are calculated, as well as for the set of perturbed samples to generate the data for the negative and positive class respectively. Figure 3.3 demonstrates how the SHAP values for a sample change when the model is looking at a perturbed sample, illustrating how a model focuses on different parts of the input when presented with an adversarial sample: notice how the model seems to utilise clusters of pixels in the chest area in the original picture while the important pixels are scattered across the attack images.

I propose both fully- and semi-supervised methods using SHAP values to detect adversarial samples utilising this information.

SHAP-MLP: Trains a simple multi-layer perceptron (SHAP-MLP) on the set of SHAP values from both genuine and adversarial samples of the dataset. The model consists of an input layer, output layer and a single hidden layer. More details about the model are in Chapter 3.2.

SHAP-Conv: Trains a convolutional neural network (CNN) on the set of SHAP values from both genuine and adversarial samples. The CNN consists of two convolutional layers, the first going from 3 channels to 16 with a kernel of size 5 and the second going from 16 channels to 32 with a kernel size of 5. We use max pooling with a kernel size and stride of 2, and the ReLU activation function throughout. Following the convolutional layers is a series of 3 fully connected layers of sizes 89888×256, 256×84 and 256×1. Dropout is applied with a probability of 0.4 after the second convolutional layer and again after the second fully connected layer.

SHAP-AE & SHAP-VAE: Typically, an adversarial attack can be seen as any sample which a model classifies incorrectly; this can include genuine images which the model misclassifies. SHAP-MLP and SHAP-Conv both attempt to classify these images as adversarial. However, it is often more useful to only detect samples which have been specifically perturbed to be adversarial [128]. This results in a smaller number of samples being present in the adversarial set. Therefore I propose the use of anomaly detection methods to detect the adversarial samples.

Two semi-supervised models are experimented with: autoencoders (SHAP-AE) and variational autoencoders (SHAP-VAE) [155] trained to reproduce SHAP values of genuine samples. The reconstruction error of the autoencoder, i.e. the error between the original and reconstructed value, is then used as a measure to detect an adversarial sample. For SHAP-AE, mean squared error (MSE) is used as the loss function. For SHAP-VAE, MSE plus the Kullback-Leibler divergence is used. As the autoencoder is trained only on genuine SHAP values, the reconstruction error from adversarial SHAP values is expected to be higher - the (V)AE has not learned how to reproduce these adversarial values. An SVM can then be trained to classify reconstruction error into two classes (adversarial and genuine). The performance of both methods is reported in Chapter 3.2.

3.2 Experiments and Results

This section reports the results of our experiments and compare our approach to two current state of the art adversarial attack detection methods: a Kernel Density based approach and ML-LOO. These methods, introduced in Chapter 2.4.2, were chosen as they are two very different, yet both state of the art, methods. ML-LOO uses feature attribution values calculated via the Leave-One-Out method to detect adversarial attacks: however, unlike the proposed methods, I will show that it does not generalise well to unseen attacks and is extremely computationally intensive. The Kernel Density based approach, on the other hand, can only be applied to models that utilise dropout and does not perform very well on EHR data.

3.2.1 Experiments on EHR data

I first report the results of experiments on EHR data. Throughout all experiments, SHAP values are normalised such that they have a mean of 0 and variance of 1, and utilise a train/test split of 80/20. SHAP-MLP is trained on both the genuine and adversarial SHAP values from the MIMIC-III dataset. A grid-based cross validation search method is used to find the optimal hyperparameters for SHAP-MLP, resulting in a hidden layer of dimension 160 and a learning rate of 0.01 with the Adam optimiser. This leads to an accuracy of 77%. Similarly, on the Henan-Renmin dataset, a hidden layer dimension of 140 and learning rate of 0.01 are optimal, achieving an accuracy of 81% (Table 3.2).

Table 3.2: Results of adversarial sample detection. HR column reports the accuracy on the Henan-Renmin. CXR (C&W) reports the accuracy on C&W generated samples, having been trained on C&W samples and CXR (PGD) the accuracy of a model trained on PGD samples tested on PGD samples.

Method	**Datasets**					
	MIMIC-III	HR	CXR (C&W)	CXR (PGD)	CXR (Train: PGD;Test: C&W)	CXR (Train: C&W;Test: PGD)
SHAP-MLP	**77%**	**81%**	**100%**	99%	58%	46%
SHAP-AE + SVM	65%	53%	79%	79%	77%	79%
SHAP-VAE + SVM	66%	53%	85%	88%	**86%**	**88%**
SHAP-Conv	N/A	N/A	**100%**	**100%**	55%	65%
Kernel Density [128]	67%	67%	84%	83%	72%	66%
ML-LOO [105]	N/A	N/A	71%	78%	71%	71%

A similar approach is used for testing the autoencoder-based methods. SHAP-AE and SHAP-VAE are both trained on the set of genuine SHAP values from MIMIC-III and Henan-Renmin. After performing the same hyperparameter optimisation method described above, a grid-search finds that an autoencoder with 2 hidden layers (in both the encoder and decoder), a latent space size of 20 and a learning rate of 0.01 with the Adam optimiser provides optimal results. Experiments find that an SVM with an RBF kernel with $C = 1$ and $\gamma = \frac{1}{M}$ (where M is the number of features) gives the best results compared to logistic regression, and SVMs with other parameters, that are validated using grid-based cross validation search. Similarly, SHAP-VAE has a latent space size of 5 and a learning rate of 0.01 with the Adam optimiser. For the loss function, the MSE is added to the Kullback-Leibler divergence. An SVM using an RBF kernel with $C = 1$ and $\gamma = \frac{1}{M}$ (where M is the number of features) gives the optimal results.

3.2.2 Experiments on Imaging Data

To test the proposed solutions' ability to work on different data modalities, the same set of experiments are performed on the MIMIC-CXR dataset. CNNs are shown to achieve superior performance when compared to other model structures [156], hence the use of convolutions in SHAP-Conv allows the model to work well on imaging data. This is highlighted by the fact that it outperforms all other methods on all medical imaging experiments carried out, achieving a 100% accuracy on both attack types (Table 3.2). Class imbalances in the dataset do not affect the results as the proposed adversarial attack detectors work on balanced classes (non-perturbed images and perturbed images), and I have chosen to focus on the Cardiomegaly label within MIMIC-CXR as it itself provides a balance of positive/negative classes, reducing the possibility of any class imbalances in the training data affecting the models.

To test the semi-supervised models' ability to generalise to different attack types, experiments wherein the models trained on the MIMIC-CXR PGD data on MIMIC-CXR data perturbed by the C&W attack and vice versa are ran. Table 3.2 shows that both SHAP-AE and SHAP-VAE are able to generalise to different attack types, achieving identical accuracy when C&W-perturbed examples are added to the test set, confirming that our model can generalise to different attack methods without the need for retraining. This is extremely useful, as it means the proposed model is able to detect unseen attacks. However, as SHAP-MLP and SHAP-Conv are both fully-supervised and are trained on both the genuine and adversarial samples, they are unable to generalise to different attack types. Interestingly, while neither model are able to generalise, SHAP-Conv performs better when trained on PGD images whereas SHAP-MLP achieves a better performance when trained on the C&W samples. This could indicate that PGD perturbs images in such a way that higher-level features are affected (which will be more difficult for SHAP-MLP to detect), whereas C&W changes features on a lower level which SHAP-MLP has more success in recognising.

The ability of SHAP-AE and SHAP-VAE (both with SVMs) to generalise to different adversarial attack techniques is further demonstrated through Figure 3.4; both of these techniques have a significantly smaller inter-quartile range than the other techniques tested, showing that the performance of these models is not affected by the type of attack that they are attempting to detect. SHAP-VAE is the clear best performer on CXR data with a stable high performance in all settings.

3.2.3 Comparison to existing methods

The adversarial sample detection method outlined in [105] is used to run the kernel density based adversarial detection method presented in [128] on the MIMIC-CXR and MIMIC-III datasets. The kernel density of the final hidden layer of both the Densenet-121 and RETAIN models are estimated, performing grid-based cross validation search to find the optimal bandwidths, and then a logistic regression classifier is fit on the estimated densities to detect adversarial samples. A bandwidth of 0.1 produces optimal results; the results are reported in Table 3.2. As seen by the significant 20% drop in accuracy when C&W images are added to the test set, this method is unable to detect tasks it has not been trained on - a significant disadvantage when new attacks are constantly being developed.

The proposed techniques are also compared against the state-of-the-art explainability-based adversarial detection method ML-LOO [129]. The experiments of the authors on Densenet-121 are followed, with LOO features being extracted from the same layers, and the inter-quartile range of these feature attribution maps being utilised to detect adversarial samples. ML-LOO's ability to generalise is tested in the same way as SHAP-AE and SHAP-VAE. ML-LOO is able to maintain comparable accuracy on the unseen attack type with a $>10\%$ lower detection accuracy compared to SHAP-VAE. The Leave-One-Out (LOO) feature attribution method is also extremely computationally intensive, and is impractical for datasets with large feature spaces. The proposed methods, however, does not suffer from the same issue as they are able to utilise one of many possible approximations when calculating SHAP values (for example, throughout this chapter the GradientSHAP approximation [85] is used).

The proposed methods outperform the state of the art on all data modalities, as reported in Table 3.2. Additionally, SHAP-AE and SHAP-VAE are both able to generalise to different attack types without retraining. In contrast, Kernel Density suffers a significant drop in accuracy when tested on unseen attack types in the test set, showing it is unable to accurately classify attacks it has not been trained on, while ML-LOO maintains its performance but at a significant computational cost. Our results are compatible with those of [105, 148] in terms of EHR being a more difficult data to address with SHAP-MLP beating Kernel Density's performance by over 10% in accuracy.

3.3 Discussion

The presented results demonstrate the difficulty to detect adversarial attacks on EHR data. This is due to both the challenges associated with the data, and how LAVA generates adversarial samples; unlike the PGD and C&W attacks on medical imaging data, LAVA is a saliency-based attack method. This results in smaller changes being made to the SHAP values of adversarial samples, and so they are naturally more difficult to detect.

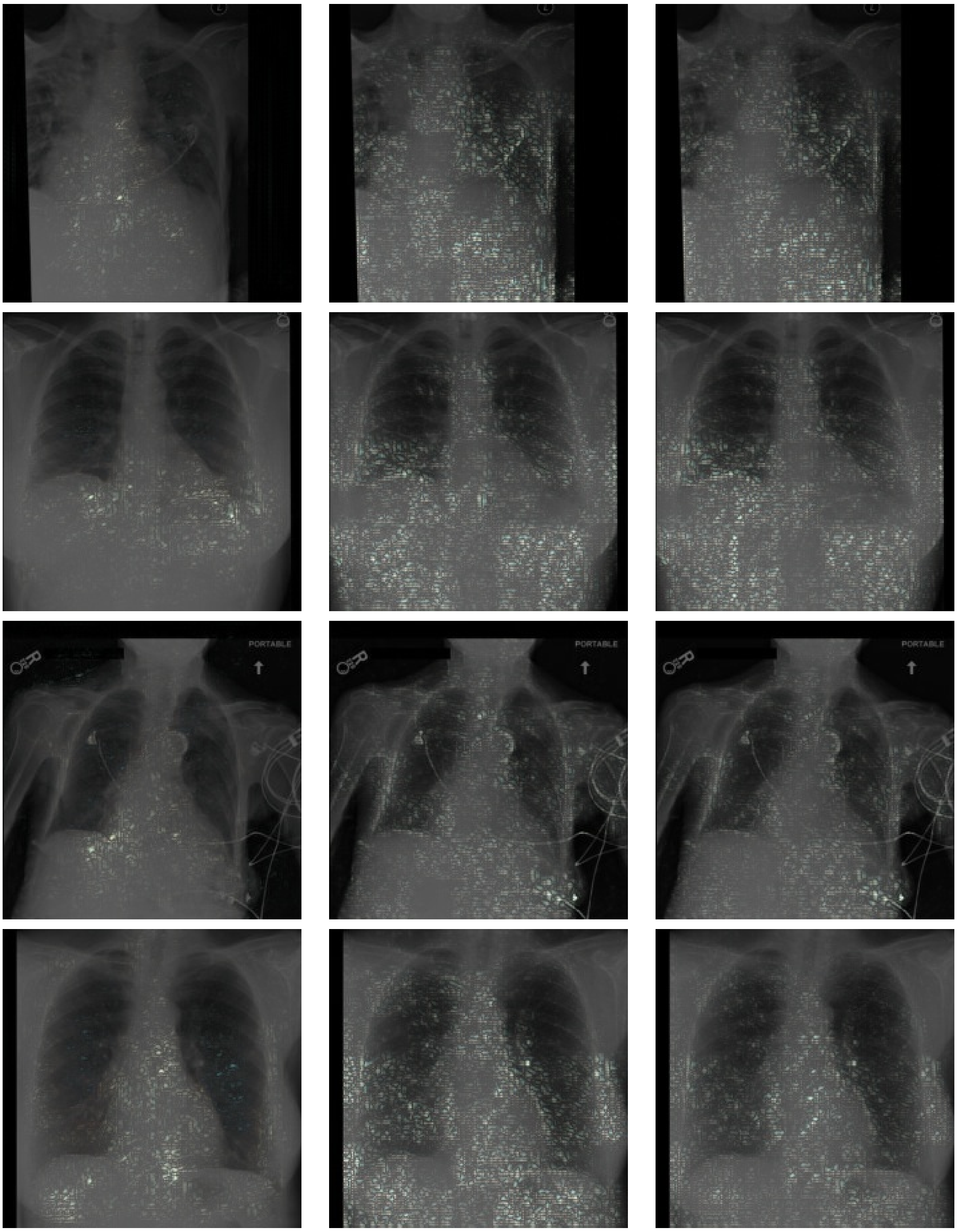
The MIMIC-CXR data is easier to work with. However, a thorough inspection of the distribution of original labels of the adversarial examples that the proposed model fails to detect finds that for all labels apart from Cardiomegaly (the label our model is trying to predict) the distribution of positive/negative labels is the same as in the original dataset. Upon investigation of the distribution of Cardiomegaly labels, I find that the proposed semi-supervised adversarial detection methods incorrectly classifies a higher proportion of positive samples as adversarial than negative samples (40% of the incorrectly classified samples are CXRs with the Cardiomegaly diagnosis, whereas in the dataset only 29% of images have the label). This shows that class imbalance in the dataset leads to difficult-to-detect adversarial samples. As the original model will most likely also have an inherent difficulty to classify one of the classes (due to the class imbalance in the training data), the adversarial sample classifier needs to learn to classify *both* perturbed samples and misclassified-genuine samples as adversarial. As the SHAP values of misclassified-genuine samples will be much closer to that of the genuine training set, this is difficult to do.

The ability of all the proposed models to work on different datatsets is useful in medical scenarios where multi-modal data [157] and non-standardised data formats [148] are common. Additionally, the ability to detect adversarial samples from unseen adversarial attacks is invaluable, as it reduces the need for bespoke detection techniques to be developed when new attack methods are discovered.

3.4 Conclusion

In this chapter I've introduced a novel method of detecting adversarial samples using SHAP values that is able to adapt to different attack types and data modalities. The method is the first such technique designed (and proven) specifically to work on both EHR and medical imaging data, despite the challenges of high-dimensionality, sparsity and temporality that it presents, and as such beats the current state of the art adversarial attack detection techniques on these data modalities. It is also able to generalise to different attack methods without any additional training. By using SHAP values we are able to explain how different attack methods work on different datasets, and use this information to detect samples which have been adversarially perturbed.

The novel methods presented in this chapter directly address one of the three main issues that were introduced in Chapter 2.1.8 as barriers to further adoption of DL in healthcare: that of robustness to adversarial attacks. Through a unique application of off-the-shelf explainability methods in the DL pipeline, we have seen that it is possible to make any classification pipeline robust to adversarial attacks, which should increase clinician's trust in DL approaches. In Chapter 5 and Chapter 6.2 I will expand upon these techniques even further and demonstrate how utilising explanations directly in the training of DL models can provide model robustness against a whole new attack vector, providing an extra layer of security that is “baked into” the model itself.



(a) (b) (c)

Figure 3.3: Each row is a different random sample from the MIMIC-CXR dataset overlayed with SHAP values when that image is passed through the finetuned Densenet121 model. (a) The heatmap of SHAP values overlayed on a genuine sample from the MIMIC-CXR dataset, (b) The heatmap of SHAP values overlayed on the same image after being perturbed via PGD, (c) The heatmap of SHAP values overlayed on the same image after being perturbed by C&W.

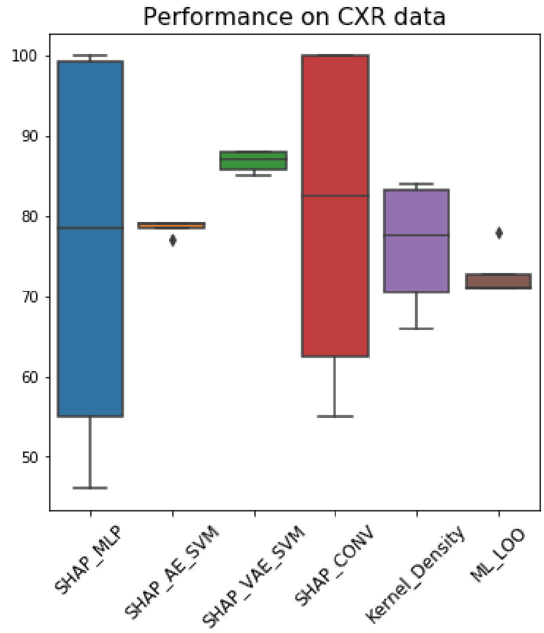


Figure 3.4: Box plot reporting the performance of adversarial sample detection methods on CXR data. The lower and upper limits of the boxes show the lower and upper quartiles of the data, the middle of the box the median and the lower and upper whiskers ±1.5 × IQR.

CHAPTER 4

Explanation (In)consistency

In the previous chapter we saw how explainability methods can be used as part of a DL pipeline to supplement a non-robust model (i.e. how to make a model more robust using explainability), and explored how this can be used to protect against adversarial attacks and hence breaks down one of the barriers to the deployment of DL models in real-world applications. However, there are still numerous issues that must be overcome for DL models to become commonplace in sensitive domains. Perhaps most importantly is the verification that our models are indeed learning causal relationships, and not relying on the types of spurious correlations that we saw in Chapter 2.1.8. For example, in healthcare we want our models to capture the same important underlying causal inter-relationships that medical professionals learn through experience. In order to ensure this, we must make our models both transparent and explainable, in order to ensure that the relevant stakeholders (patients, medical practitioners) can place their trust in the model, and to help prevent "catastrophic failures" [158, 159].

A model that could be proven to use causal features would be robust to spurious correlations and changes in model training perpendicular to the classification task; however, this level of model analysis is currently extremely difficult (if not impossible in most real-world use cases) to achieve [160]. Without this level of robustness there will be no trust for its use in the real-world. Current DL training methods often fail to satisfy this requirement, as robustness/trust is yet to be an intricate part of the evaluation and optimisation of said models [10, 99].

Recent theoretical and experimental work has demonstrated the challenge of generalisation for DL models and their vulnerability to small changes in the data [111]. Ensemble models, where multiple, slightly different models work together to make a final prediction, have been proposed to alleviate these issues [115, 161]. However, while these techniques can improve the robustness of models, they are rarely inherently explainable and do not necessarily understand causal relationships. Additionally, a fundamental requirement of trustworthy models is the interpretability of their decisions. The development of explainable DL techniques to date use either model agnostic post-hoc or model specific approaches. However, the quality of explainable methods is still very difficult to quantify and is geared to be truthful to the model not the data [79, 94].

Chapter 3 shows how an over-reliance on non-causally related features can lead to models being vulnerable to adversarial attacks. On the other hand, the novel methods introduced in Chapter 3.1.3 show that explanations are extremely sensitive to changes in a model's inputs (so much so that they can detect malicious inputs with near-perfect accuracy) - the next logical step is to investigate whether they can also be used to measure differences between two models. This chapter aims to answer this question, and explores the limits of explainable machine learning and highlights fundamental problems in the training and generalisation of neural networks. Most notably, it:

- Demonstrates how the noise learned by a deep learning model can change significantly when factors such as the random seed, initial weights or even training set order are changed (whilst all other variables remain unchanged)

- Proposes a measure of the consistency of explanations to quantify the problem and discuss its impact on the interpretation of the explainable output in relation to the input features importance.

- Shows that even the current state-of-the-art ensemble models present with the same issues, and discuss the implications of these findings on the viability of deploying machine learning models in sensitive fields such as healthcare.

4.1 Problem Motivation

As covered in Chapter 2.3.1, there is little in the way of a concrete, mathematical understanding of why, how and to what degree DL models are able to generalise to unseen data distributions. In particular we looked at issues such as shortcut learning, spurious correlations and a bias-variance trade-off for DL models that contradicts our traditional interpretation of model training. I argue that, ultimately, our lack of mathematical understanding of neural networks is holding back the development of DL models and that more work should be done to understand exactly how, and why, DL models are able to train.

Inspired by the results in Chapter 3.2, which show that post-hoc explainability techniques are very sensitive to changes in a model's input, in this Chapter I look at using explanations to understand the inner workings of model training - specifically, I use explanations to investigate how changes to the training process (no matter how small) can drastically change the resulting model. Notably, I demonstrate that generated explanation can be unstable and inconsistent due to variations in model training that are irrelevant to the classification task and that even model architectures that are designed to overcome some of these problems (such as those discussed in Chapter 2.3.2) fail to mitigate against this problem.

A closer look at explainable outputs of DL models allows us to understand how the randomness introduced during the training significantly affects the explanation of the model's decisions despite consistent accuracy levels. This raises important questions around the robustness of these models. On the contrary, kernel methods (namely SVMs) are robust against these changes, suggesting that it is the stochastic nature of deep learning model training that may be causing these issues to arise. I argue that these issues significantly impede our ability to confidently suggest DL models for use in healthcare, as they imply that the models might be relying on spurious correlations in the data leading to models producing inconsistent explanations upon retraining.

4.2 Measuring Explanation Consistency

I argue here that consistency of the explanations produced by a model regardless of orthogonal changes to hyper parameters (i.e. hyperparameter changes that do not affect the architecture/structure of the model) is a strong surrogate to model robustness. Fidelity of explanations on the micro level, i.e. input features, is the basis to quantify explanations [94, 162].

This chapter explores validating explainability on the macro level, i.e. the robustness of the produced explanation regardless of changes to model training that are orthogonal to the model architecture, data content, and classification task. Intuitively speaking, the consistency of explanations across model variations engender trust in these models as the end user does not expect changes in the explanation due to an incremental model update. Existing similarity metrics (e.g. cosine similarity, root mean squared error) are ill-suited to this task as they are unable to accurately quantify the small (yet important) changes that we are particularly interested in. A binary classifier is well-suited to this task, however, as even when the saliency maps 'look' similar, the classifier will be able to use these small differences to separate the values.

4.2.1 A Measure of Consistency

Given a dataset $X = \{x_1, ..., x_N\} \subset \mathbb{R}^d$, where $d \in \mathbb{N}$ is the dimension of the sample data, we have a classification task $Y(x_i) \in \{0, 1\}^n$, where n is the number of classes in a classification setting. The metric that will be proposed in this section aims to evaluate the consistency of explanation method E, where $E(Y(x_i)) \in R^d$ assigns a weight to every input feature based on its influence on $Y(x_i)$.

Assume we have V variations of the model Y, which will indicated as $Y^v, v \in \{1, \ldots, V\}$, then I define the explanation separability of any two of these variations as:

$$S_{(a,b)} = \mathbb{E}_i\Big[D\Big(E(Y^a(x_i)), E(Y^b(x_i))\Big)\Big] \tag{4.1}$$

where $i \in \{1, \ldots, N\}$, and D is a similarity measure between the two explanations provided by E of the output of the two models Y^a and Y^b. The larger $S_{(a,b)}$ is then the more distinct the explanations produced by the same model architecture under the training conditions, a and b. As $S_{(a,b)}$ measures the (in)consistency across all inputs in the dataset, $S_{(a,b)}$ takes the form of the expected value of the difference between all tested hyperparameter pairs (a, b) data points.

Without loss of generality I assume $S_{(a,b)}$ to be normalised in the range $[0, 1]$ and define consistency as:

$$C = 1 - \frac{\sum_{(a,b)} S_{(a,b)}}{\alpha} \tag{4.2}$$

where α is the number of comparisons made between variations of the trained model. The separability metric $S_{(a,b)}$ should be defined such that when the explanations are completely separable (i.e. $S_{(a,b)} = 1$) then the consistency $C = 0$, and vice-versa.

4.2.2 Choosing a Suitable Separability Metric

The definition of $S_{(a,b)}$ should be determined based on the characteristics of X, e.g. data dimension and sparsity, and as such it makes sense that slightly different definitions may be appropriate in different scenarios. There are numerous different definitions that could be chosen ranging from information-theoretic measures of similarity to statistical metrics of similarity (note that similarity metrics can be modified to fit my definition of $S_{(a,b)}$ by "flipping" their output to ensure that $S_{(a,b)} = 0$ when a, b are identical).

Throughout this chapter, and the remaineder of this thesis, I use the testing accuracy of a binary model, $M_{(a,b)}$, trained to classify between $E(Y^a(x_i))$ and $E(Y^b(x_i))$ for $i \in 1, \ldots, T$, where T is the size of the testing set. Equation (4.2) can then be re-written as:

$$C = 1 - \frac{\sum_{(a,b)} 2 * |M_{(a,b)} - 0.5|}{\alpha} \tag{4.3}$$

where $|.|$ is the absolute operator. $S_{(a,b)}$ is set to $2 * |M_{(a,b)} - 0.5|$ to normalise the classification accuracy and make it more meaningful as separability by measuring its distance from theoretical random baseline. An accuracy $M_{(a,b)} = 1$ means the two explanations are completely separable with $S_{(a,b)} = 1$ and $C = 0$, and on the other extreme an accuracy $M_{(a,b)} = 0.5$ means that there is perfect agreement between a and b resulting in $S_{(a,b)} = 0$ and $C = 1$.

However, while I have chosen to use the cross validated training accuracy of a binary classifier to measure the distance, D, between the explainability values, as noted earlier different distance measures could be used and it may be the case that different distance metrics are suited better to different applications and datasets. When choosing a separability metric, it is important to determine whether the chosen distance metric is sensitive enough to detect the small changes in the explanations that we wish to detect. Each possible consistency metric will have various advantages and disadvantages, and it may be that some are better suited to different scenarios; one of the reasons a binary classifier is used throughout this chapter is its suitability to almost any scenario and data modality.

As there are so many separability metrics available to use, I provide evidence that the chosen method (using a binary LR classifier) is indeed the best suited metric to use for our specific scenarios. Table 4.1 contains the values of other, classical consistency (i.e. divergence) measures that have been tested on 4 CNNs (of identical architecture) trained on MNIST with different random seeds. Jensen-Shannon (JS) divergence is based upon Kullback-Leibler (KL) divergence, and is a method of measuring the similarity between two probability distributions; due to its relation to KL divergence, JS divergence is common in machine learning applications, making it a prime candidate for use here. JS divergence is better suited for measuring consistency as it is normalised, and hence lies in the range $[0, 1]$. Its main disadvantage is that it measures the divergence between probability distributions, and not samples drawn from a distribution. This requires the distribution of the explainability values for the two models we want to test to be estimated from the explanations we have generated. This adds an extra layer of complexity to the calculation, and could lead to errors where differences in the techniques and assumptions used to estimate the probability functions. For the experiments reported in Table 4.1 Kernel Density Estimation (KDE) was used, a method of estimating an unknown probability density function using a kernel function [163]. While this has produced good results for this set of experiments, the most effective kernel density estimation technique is entirely problem-dependent, whereas the binary classifier method discussed in the previous section is more generalisable to a wide range of data.

There are also a number of statistical hypothesis tests that are designed to test whether two sets of samples are drawn from the same distribution. The 2 sample Kilmogorov-Smirnov (KS) test is a two-sided test for the null hypothesis that the 2 sets of samples are drawn from the same continuous distribution [164]. Using the KS test as a consistency measure has the benefit of having a solid statistical underpinning, but it encounters problems when carrying out on real-world datasets. While it is possible to accurately compute the test statistic (reported for a small set of model in Table 4.1), the associated p-values are incomparable, meaning it is impossivle to accurately complete the hypothesis test. In all experiments (except those which are testing a model against itself, where a test statistic of 0 and p-value of 1 was calculated), the test statistic calculations returned a p-value of 0. A similar issue arises when the Wilcoxon signed-rank test is used, which is a non-parametric alternative to the paired t-test which can work on highly non-normal data that works on the null hypothesis that the median differences between pairs of samples are 0. While these results (i.e. calculating a p-value of 0) highlight that the results are highly statistically significant (and hence the null hypothesis can be rejected, resulting in the conclusion that the explanations are drawn from different distributions), as all of the calculated p-values are 0 it is not statistically correct to use results from hypothesis tests to quantify to what degree the explanation's from two models are (in)consistent (i.e. we are unable to infer if one architecture produces more consistent explanations than another), whereas our results with a binary LR classifier allow us to do so.

This is not to say that JS divergence or KS/Wilcoxon hypothesis tests are entirely

M1 Seed	M2 Seed	JSD	KS	Wilcoxon	LR
1	1	0	0	0	0.5
1	12303	0.8062	0.9744	7.877e+09	0.973
1	15135	0.8012	0.9690	1.738e+10	0.978
1	16959	0.7346	0.8890	2.464e+11	0.975
12303	12303	0	0	0	0.5
12303	15135	0.8228	0.9913	4.350e+08	0.979
12303	16959	0.7900	0.9567	3.316e+10	0.974
15135	15135	0	0	0	0.5
15135	16959	0.8122	0.9810	6.611e+09	0.975

Table 4.1: Table reporting the Jensen-Shannon divergence, 2 sample Kilmogorov-Smirnov and Wilcoxon signed-rank *test statistics* on the SHAP values from a small subset of the MNIST CNNs tested. The p-values for all hypothesis tests were calculated as 0. Kernel Density Estimation was used before calculating the Jensen-Shannon divergence of the explanations. LR is the accuracy of Logistic Regression classifiers trained on the SHAP values, as used throughout this paper as $M_{(a,b)}$.

unsuited to use as a consistency measure. This chapter focuses on experiments on image data, where inputs contain a large number of features; applications where inputs have fewer features may find it possible to calculate the p-values for hypothesis tests, or to produce more accurate probability functions for the explanations. In these cases, it may be appropriate to use one of these measures. However, my choice of a binary classifier is easy to use in any scenario, to any dataset and is easy to interpret and quantify.

4.3 Experimental Setup

Experiments are performed across two publicly available computer vision datasets. MNIST is used for efficient baseline tests, with experiments then being expanded to use MIMIC-CXR-JPG [59]. Experiments investigate a wide breadth of different model architectures, explanation methods, and training variations. For both datasets, the recommended train/test/val splits is used. For reproducibility, the specific hyperparameters used for each experiment can be found in Table 4.4.

MNIST Experiments: Were carried out with the following model variations: **1) MLP** with two hidden layers of sizes 412 and 512 respectively and a dropout layer, **2) Small-CNN**, a convolutional neural network with 1 convolutional layer with kernel size 3, followed by a max pooling and fully connected layer, **3) CNN** with two convolutional layers with kernel size 3, using max pooling and fully connected layers in between, **4) GaborNet**, a Small-CNN network with the first convolutional layer restricted to use Gabor filters [112], **5) ResNet18** [165] with the first convectional layer modified to take 1 channel inputs and the final output layer to have an output size of 10, and **6) SVM** with RBF kernel. We also train two ensemble models: **1) ADP ensemble** [115] using the default hyperparameters and consisting of 10 ResNet sub-models, and **2) Hyperensemble** a hyper-batch ensemble [25] using the default hyperparameters with 3 sub-models.

MIMIC-CXR-JPG Experiments: The dataset contains 377,110 chest x-rays (CXRs) images from 227,827 studies [59]. Each study has up to 14 associated labels denoting the disease(s) which are present in the CXR images. This study focuses only on images with the Edema label; this gives a subset of 77,483 images of which 47.2% present with the disease (have a positive label) and the remaining 52.8% do not (have a negative label). The labels are used as presented in the MIMIC-CXR-JPG dataset: these have originally been extracted from free-text radiology reports via the CheXpert tool [27, 59]. The MIMIC-CXR-JPG dataset is used to demonstrate the issues raised in this chapter on a real-life healthcare application. Experiments in this chapter have been focused on the Edema label as otherwise we are left with a multi-label classification problem (as one CXR image may show multiple diagnoses), which would make isolating the source of variation very difficult to guarantee. Specifically, the Edema label was chosen as it provides a large number of images whilst also having largely balanced classes. The scope for experimentation with MIMIC-CXR-JPG is necessarily more limited than that with MNIST, as the data requires more complex networks to gain optimal performance. For model creation the same process as CheXNet [40] is followed, fine tuning a pre-trained Densenet-121 model. Additionally, a voting ensemble consisting of 3 pre-trained Densenet-121 models is also trained on subsets of the training dataset.

On both datasets, the models are trained repeatedly. For each run the model's hyperparameters are changed, leading to variations in the randomness used during training without changing the architecture of the model. Specifically, the following

Model Architecture	Dataset	Shuffle	Random Seed	Dropout
MLP	MNIST	98.195 ± 0.9550	98.18 ± 0.94	98.25 ± 0.8292
SVM	MNIST	93.825 ± 0.7746	94.218 ± 0.3943	n/a
Small-CNN	MNIST	98.385 ± 0.0250	98.345 ± 0.015	98.3267 ± 0.0330
ADP Ensemble	MNIST	98.5 ± 0.14	99.0875 ± 0.2573	n/a
CNN	MNIST	97.5 ± 0.5	99.2170 ± 0.0443	99.1580 ± 0.0595
GaborNet	MNIST	95.031 ± 0.2769	95.034 ± 0.2742	95.054 ± 0.2934
ResNet18	MNIST	99.083 ± 0.2514	99.471 ± 0.0438	n/a
Densenet-121	MIMIC-CXR	76.005 ± 0.8363	75.4535 ± 1.2539	n/a
Densenet-121 Ensemble	MIMIC-CXR	81.98 ± 0.34	80.8533 ± 0.5311	n/a
Hyperensemble	MNIST	n/a	99.32 ± 0.0082	n/a

Table 4.2: Table reporting mean model accuracy (± standard deviation) across model training variations on the base classification task.

hyperparameters are changed: **1)** the random seed used during training, **2)** the dropout rate used in the networks (where applicable), and **3)** the order of the training data. It is important to note that the train/test/val splits remain the same, rather it is the order in which the training data is passed to the model during training which changes. The accuracy of the models on the base classification task (i.e. MNIST or MIMIC-CXR) are summarised in Table 4.2.

To inspect the consistency of decision explanations as a result of changing these hyperparameters, two state-of-the-art explainability techniques are used: SHAP [85] and Integrated Gradients (IG) [88]. These two techniques were chosen as they represent a wide range of state of the art feature-attribution explanation methods: I) SHAP is a permutation-based model-agnostic approach, so can be applied to the output of any model II) IG is gradient based making it applicable for all neural networks architectures. The explanation consistency for each explanation technique per model and dataset is calculated, taking into account every training variation. A Logistic Regression (LR) classifier is used as the binary model to classify between $E(Y^a(x_i))$ and $E(Y^b(x_i))$ as per Equation (4.3). This LR model takes the explanation values (i.e. SHAP values, IG values) of the two models as input, and is trained to classify which model the values originated from. The average training accuracy from 10-fold cross validation of the LR model is used. The higher the accuracy of the LR models, the more separable the explainability values are, suggesting that the two models are placing importance on significantly different parts of the input.

To confirm that the underlying problem lies in the models themselves, and not

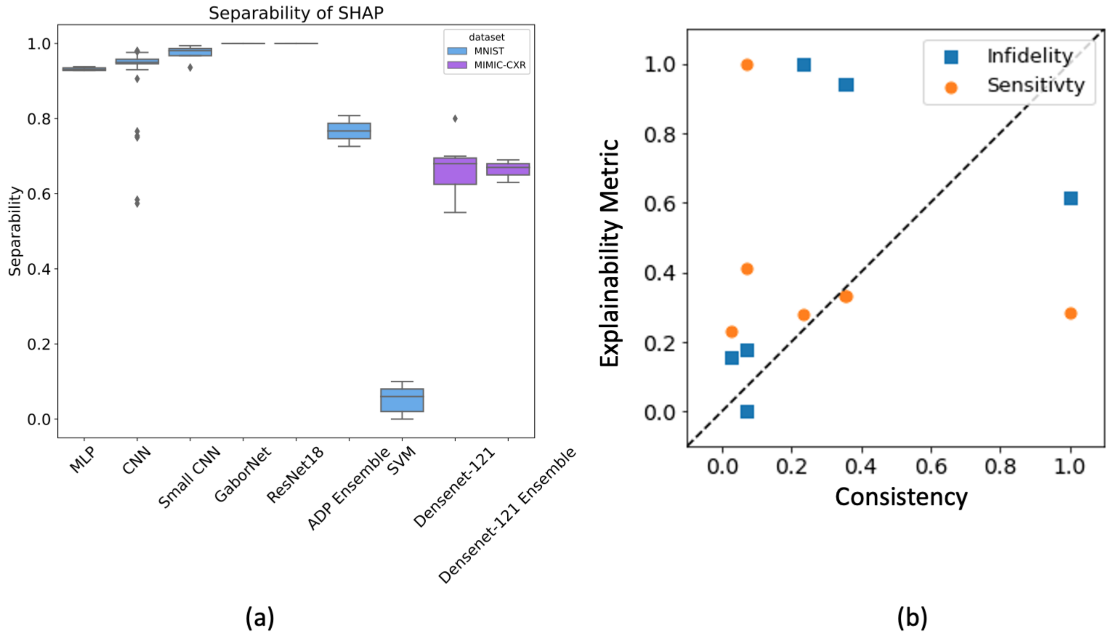


Figure 4.1: (a) Box plot of $S_{(a,b)}$ for SHAP across all training variations (a, b), for all model architectures tested. (b) Plot of SHAP explanation consistency of model architectures vs. SHAP infidelity and sensitivity of the same models across both MNIST and MIMIC data.

the explainability techniques used, the quality of the explanations are measured via three different explanation quality metrics (introduced in Chapter 2.2.6) that are designed to ensure the explanations produced accurately represent the models: (in)fidelity, sensitivity and explanation accuracy. These three metrics have been chosen specifically as they each evaluate a different aspect of the explanation and together provide a holistic view of an explanation technique's quality.

4.4 Results and Discussion

Through visualisation of the explanation differences, it is possible to discern whether the lack of consistency between variations is a cause for concern when deploying deep learning models to real-world scenarios. Figure 4.3 demonstrates the change in explanations between two variations of the same Densenet-121 model using SHAP. There are two main sets of differences in the images: **1)** areas of the image that are clinically significant (e.g. the lungs and the heart), and **2)** areas in background portions of the image. Those differences that are in clinically relevant to diagnosis

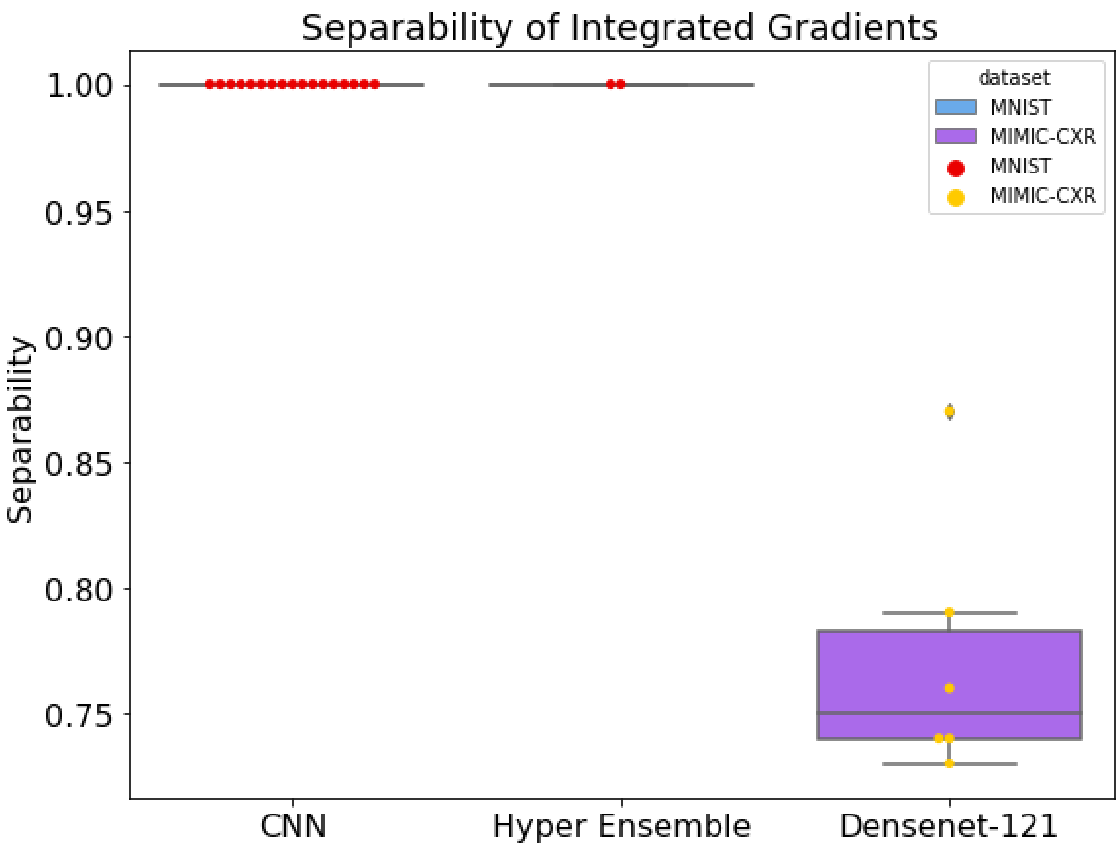


Figure 4.2: Boxplot of the separability $S_{(a,b)}$ of the Integrated Gradients explanations. For clarity, note that the CNN and Hyper Ensemble models are trained/tested on the MNIST dataset, and Densenet-121 on MIMIC-CXR.

can result in significantly reduced trust in the model, as we ideally want a model which has learnt the entire set of causal links present in the data (whereas these differences show that the two models have learnt to look at different sets of causal features). The remaining differences are in the background noise of the images, which suggests that the models are potentially picking up spurious correlations, with each model learning different sets of spurious correlations. Neither of these scenarios are desirable. Examples on Small-CNN trained on MNIST are shown in Figure 4.5 - similarly to the CXR samples, changes in the SHAP values are mainly centered around the areas of the image that are critical for number classification. These results are significant - it suggests both that variations in the training setup of a model changes the importance of the fundamental features that we would expect to be causally linked to the final classification, and on more complex tasks are also changing the spurious correlations learned by models.

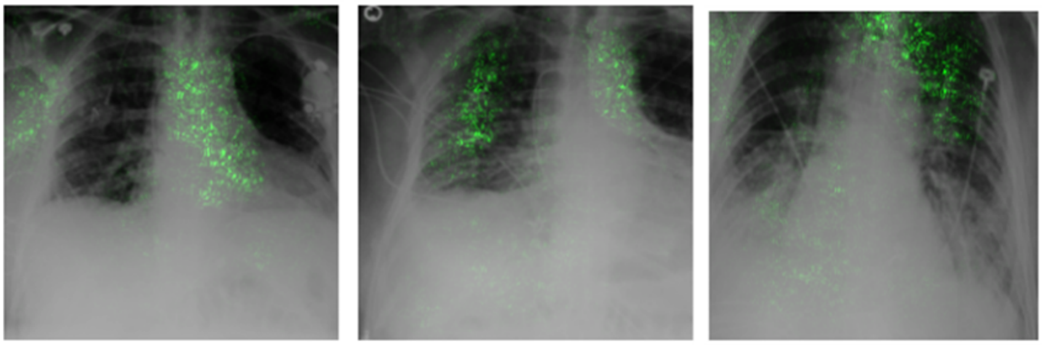


Figure 4.3: 3 random samples from the MIMIC-CXR-JPG dataset overlayed (in green) with the difference between the normalised SHAP values from two Densenet121 training variations.

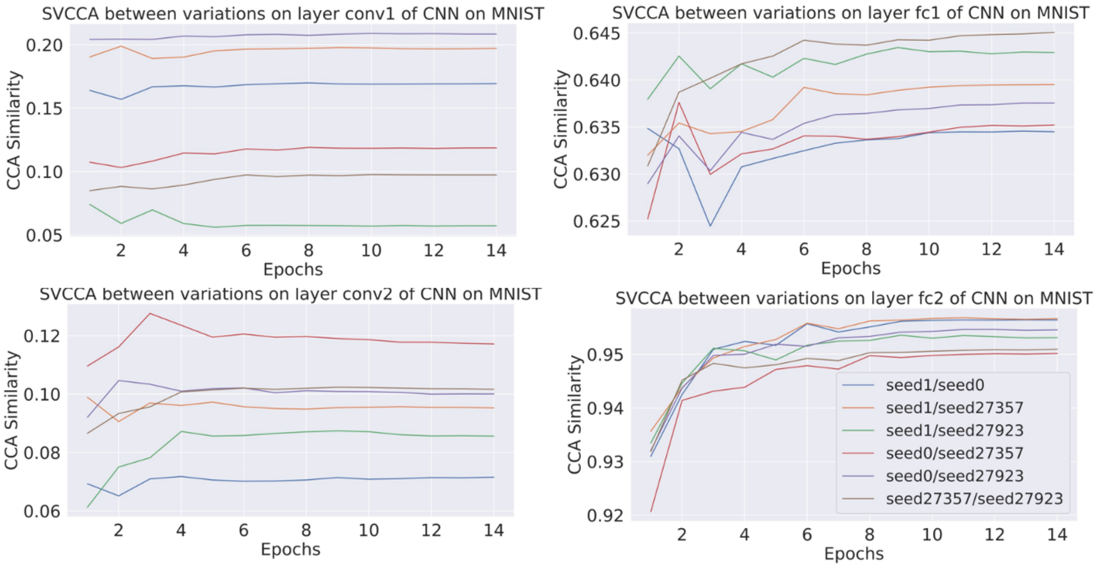


Figure 4.4: Figures showing the CCA similarity as training progresses between layer parameters. Each coloured line is a separate training variation pair of a CNN trained on MNIST.

Following, the accuracy of all models tested on MNIST and MIMIC-CXR-JPG is reported, as well as the consistency of the explainability methods per model/dataset. Table 4.3 contains each model architecture's consistency, and a further breakdown of the consistency for the different types of training variation tested. For all model architectures, the degree of consistency is similar irregardless of which hyperparameters is changed; this suggests that deep learning models are sensitive to all training hyperparameters, and not just a select few.

Figure 4.1(a) and Figure 4.2 further demonstrate the variation in the separability measure ($S_{(a,b)}$) across all models/datasets. These figures show that there is very

		Consistency					
Model Architecture	Dataset	α	Overall	Shuffle	Random Seed	Dropout	Accuracy
MLP	MNIST	6	0.0668	0.062	0.066	0.0687	98.125 ± 0.9270
SVM	MNIST	10	0.9444	0.96	0.94	n/a	94.0556 ± 0.6213
Small-CNN	MNIST	6	0.0252	0.018	0.06	0.034	98.3486 ± 0.0360
GaborNet	MNIST	12	0	0	0	0	95.038 ± 0.2824
ResNet18	MNIST	10	0	0	0	n/a	99.425 ± 0.0626
ADP Ensemble	MNIST	6	0.2193	0.192	0.233	n/a	99.083 ± 0.2514
CNN	MNIST	12	0.0652	0.052	0.0564	0.0914	98.9976 ± 0.5756
Densenet-121	MIMIC-CXR	6	0.3329	n/a	0.3329	n/a	75.6723 ± 1.1379
Densenet-121 Ensemble	MIMIC-CXR	4	0.3367	n/a	0.3667	n/a	80.8 ± 0.7483
CNN (IG)	MNIST	12	0	0	0	0	98.9976 ± 0.5756
Hyperensemble (IG)	MNIST	2	0	n/a	0	n/a	99.32 ± 0.0082
Densenet-121 (IG)	MIMIC-CXR	6	0.168	0.115	0.2033	n/a	75.6723 ± 1.1379

Table 4.3: Table reporting the consistency between training variations for the models tested and the average accuracy of the model architecture on the base classification task. The Shuffle, Random Seed and Dropout columns report the consistency of models when *only* the respective hyperparameter was changed. The Overall column reports the overall consistency of that architecture, taking an average of the consistency across all hyperparameters. α refers to the number of models tested for the overall architecture consistency (see Equation (4.2)). Please refer to Table 4.4 for α values for the shuffle, seed and dropout consistencies.

little consistency of either SHAP and IG for any training variation when used with deep learning models. The experiments find that SVMs do not suffer from the same issue as deep learning models, achieving very high levels of consistency across both random seed and training shuffle variations. This provides evidence for my hypothesis that it is the stochastic nature of deep learning model training that may be causing these issues to arise. Figure 4.2 shows the boxplot for IG, with even more pronounced separability, which can likely be attributed to how IG is calculated based on the weights of the network. Figure 4.1(a) does not show any real link between the size/depth of a network architecture and its consistency.

Interestingly, both GaborNet and ResNet18 are completely inconsistent. The purpose of Gabor filters in CNNs is to more accurately simulate our biological understanding of human vision, and so one would expect the feature maps learned by these networks to more accurately represent those parts of an image that a human would use for feature recognition. However, although this may still be the case, our results show that these models may still be relying on noise and/or only learning a subset of the important features each time the model is trained; if either of these were not true, we would expect the models to be more consistent. The purpose

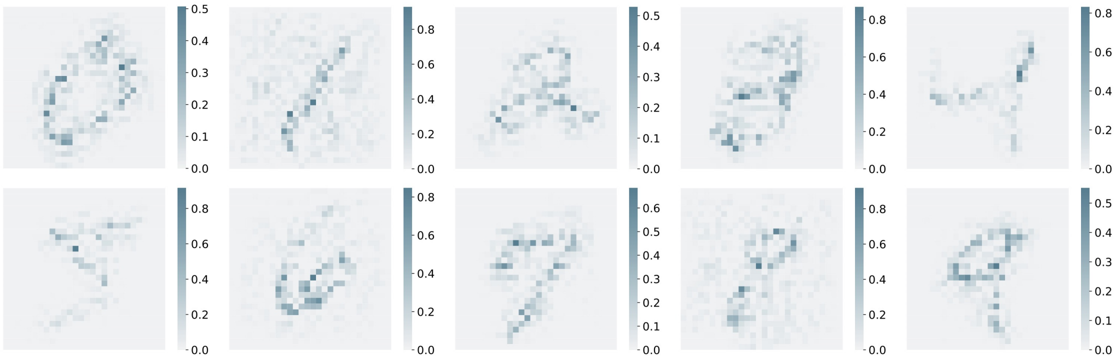


Figure 4.5: The difference between normalised SHAP values from two CNNs (each trained with different random seeds) for a randomly chosen sample from each MNIST class.

of testing the ResNet18 architecture was to investigate whether overparameterised networks also suffer from this inconsistency problem; as can be seen in Figure 4.1(a) and Table 4.3, they do. This implies that even models which have many more times the number of parameters than data points are converging to slightly different points on the loss landscape when small hyperparameter changes are made. It also suggests that even high capacity networks, which one would expect to be able to learn the entire set of meaningful features, are in fact either not able to do so or are still relying on some set of noisy features.

Figure 4.1(b) shows the correlation, or lack thereof, between explanation consistency and (in)fidelity and sensitivity as measures of the explanation's quality across all experimental settings. For both cases there is a weak Pearson correlation (0.4 for (in)fidelity and -0.3 for sensitivity). This is not surprising as those metrics are designed to be faithful to the model, not to the underlying data. This leads to these quality metrics producing similar values across all variations, as they focus on the whether the explanation is a good representation of what the model considers important whereas the purpose of this work is focus on how these important features can change across runs. In general, ensemble approaches seem to have higher consistency but it is still significantly lower than that of SVMs. To further measure the quality of the SHAP and IG explanation, the explanation accuracy for each model is also calculated. Each individual model's explanation infidelity, sensitivity max and accuracy is reported in Table 4.4. These explanation quality metrics support two conclusions: **1)** the weak correlation between the quality metrics and consistency shows that these metrics are unable to detect inconsistent models, and **2)** as the metrics are reporting that the explanations are indeed faithful to the model, then it must be that the models themselves (or the training process) are responsible for the inconsistency rather than the explanation techniques themselves.

SVCCA [166] is used to inspect the similarity of layer parameters between two training variations, and investigate how these change as training progresses. SVCCA views neurons using their activation vectors, and uses an amalgamation of Singular Value Decomposition and Canonical Correlation Analysis to analyse these representations. Figure 4.4 shows the SVCCA similarity between layers of CNN trained on MNIST with different random seeds. It clearly shows a high degree of similarity for the final layer, whereas the middle convolutional layer (`conv2`) shows a significant difference. This corroborates the explainability consistency results; the final layers (`fc2`) are similar (and so the models will produce similar outputs, resulting in similar performance levels), whereas all other layers are significantly different (and so the explanations, which take into account the whole model, are different). In addition to this, the two convolutional layers show an extremely low degree of similarity between the two models, hence the feature maps learned by these two models are likely also not similar (resulting in the final explanations having high degrees of separability).

4.5 Conclusion

In this chapter I've introduced a consistency measure of explainable machine learning and demonstrated that deep learning models converge to learn different features when the same model is trained with different random seeds, training set orders and dropout rates. By validating the quality of the explanation techniques used, and using both gradient-based and perturbation-based techniques, we have shown that this is a fundamental problem with deep learning models rather than an issue with the explanations. Additionally, I've verified that SVMs are immune to this problem. I argue that there is still significant work that need to be done to build robust trustworthy deep learning solutions in real-life healthcare applications - in Chapter 5 we will explore methods that utilise model explanations during training that aim to address the problem of explanation inconsistency, and then explore how this has positive affects in other areas of model robustness in Chapters 6.1 and 6.2.

Model Type	Dataset	Dropout	Seed	Shuffle	Explanation Accuracy	Sensitivity	Infidelity
CNN	MNIST	0.0	1	False	96	2.40.	0.0019
		0.1	1	False	97	2.04	0.0018
		0.2	1	False	97	2.02	0.0020
		0.3	1	False	98	1.69	0.0019
		0.4	1	False	98	1.53	0.0016
		0.25	1	False	98	1.84	0.0016
		0.25	12303	False	98	1.70	0.0014
		0.25	15135	False	98	1.58	0.0020
		0.25	16959	False	97	1.67	0.0018
		0.25	20878	False	98	1.61	0.0020
		0.25	79266	True	99	1.51	0.0014
		0.25	79870	True	99	1.67	0.0011
Small-CNN	MNIST	0.0	1	False	99	1.07	0.1810
		0.2	1	False	98	1.00	0.1429
		0.25	1	False	98	1.00	0.1521
		0.25	26417	False	99	1.02	0.1011
		0.25	91110	True	99	1.01	0.1174
		0.25	98281	True	99	1.01	0.1402
GaborNet	MNIST	0.0	0	False	99	1.38	0.2808
		0.1	0	False	99	1.41	0.2256
		0.2	0	False	99	1.42	0.1900
		0.3	0	False	99	1.44	0.1702
		0.4	0	False	99	1.46	0.1523
		0.25	257	False	99	1.17	0.1489
		0.25	6339	False	99	1.34	0.2508
		0.25	29062	False	99	1.40	0.1683
		0.25	51303	False	98	1.45	0.2352
		0.25	17939	True	98	1.34	0.1567
		0.25	23682	True	98	1.28	0.1190
		0.25	27442	True	99	1.31	0.1274
		0.25	53307	True	99	1.27	0.1089
ResNet18	MNIST	0.25	21609	False	99	1.15	0.7214
		0.25	23474	False	99	0.96	0.4426
		0.25	29246	False	99	2.34	0.5284
		0.25	48769	False	98	0.83	0.5007
		0.25	58626	False	99	1.21	0.7121
		0.25	72	True	98	1.21	0.5572
		0.25	1507	True	98	1.42	0.8697
		0.25	4439	True	99	0.97	0.5402
		0.25	10250	True	99	2.10	0.8867
		0.25	21033	True	99	1.01	0.9018
MLP	MNIST	0.0	1	False	99	3.49	0.1748
		0.2	1	False	99	5.56	0.1573
		0.25	1	False	99	4.85	0.1508
		0.25	27833	False	99	3.76	0.1926
		0.25	72	True	99	3.39	0.1427
		0.25	79870	True	99	3.74	0.1399
Densenet121	MIMIC	n/a	2	False	99	1.5966	0.9994
		n/a	3	False	99	1.5031	1.0719
		n/a	4	False	99	1.5987	1.0020
		n/a	5	False	99	1.1431	0.4659
		0.25	6	True	99	1.5122	0.9994
		0.25	7	True	99	1.6078	1.1217
ADP	MNIST	n/a	0	False	99	1.2187	0.9110
		n/a	42	False	99	1.4250	1.4376
		n/a	100	False	98	1.2297	0.9730
		0.25	1	True	99	1.3491	0.9912
		0.25	10	True	98	1.3100	1.266
DNE	MIMIC	n/a	1	False	80	1.5499	1.0357
		n/a	42	False	84	1.3709	0.7340
		0.25	4242	True	81	1.5683	0.6510
		0.25	1000	True	82	1.6932	0.8493
SVM	MNIST	n/a	30828	False	99	1.5763	0.2070
		n/a	31599	False	99	1.1686	0.9074
		n/a	8253	False	99	1.0238	0.6214
		0.25	91244	True	99	1.5439	0.5006
		0.25	79870	True	99	1.5894	0.4823

Table 4.4: Table reporting explanation quality metrics on SHAP across all model architectures and training variations tested. DNE denotes Densenet-121 Ensemble. Where Shuffle is `True`, Seed refers to the seed used for shuffling the dataset and not the training seed.

CHAPTER 5

Deep Explanation Ensembles

Through a novel application of explainability techniques the previous chapter has shown how there are inherent problems with the stochastic nature of neural network training which results in notable inconsistencies in model explanation's when hyperparameters such as the training seed and order of the training data are changed. This is a significant problem when it comes to sensitive applications such as healthcare and finance where, as discussed in Chapter 2.1.8, transparency and explainability is paramount. This, coupled with concerns between the gap between training and real-world data distributions and shortcut learning (Chapter 2.3.1), suggests that possibly DL models are not learning causal features and instead are relying on spurious correlations.

In this chapter I explore this problem further, by investigating and proposing solutions to the inconsistency between models trained on the medical and biological tabular datasets introduced in Chapter 2.1. I focus on these applications as it is in these sensitive situations that inconsistent models pose the most significant risks and barriers to the adoption of ML. These are also highly specialised areas of expertise where interpretation of model output can have significant influence and can also be directly challenged. Furthermore, by initially restricting my evaluations on tabular datasets it allows a more focused analysis of the proposed methods; tabular data are inherently easier to understand and explain than other data modalities with a (typically) small number of features that are usually well understood by domain experts. Note, however, that I extend the analysis of the Deep Explanation Ensemble technique proposed in this chapter to image data in Chapter 6.1.

Concretely, this chapter:

- Extends the experiments in Chapter 4.3 to 6 tabular classification tasks
- Proposes and evaluates a post-processing technique that can be applied to explanations from any model architecture that significantly improves explanation consistency as defined in Chapter 4.2
- Proposes a novel ensemble architecture and training algorithm that takes advantage of explainability techniques during model training to produce an overall model that is more consistent than the sum of its parts
- Evaluates the effectiveness of this new architecture on the same tabular datasets as our initial experiments, and compare our results to the current state of the art
- Discusses how this technique could be used in practice and identify potential future directions.

5.1 Methods

This section initially proposes a novel post-processing technique that can be applied to explanations from any model architecture. The ideas introduced in this method are then taken and embedded in the training of a novel ensemble architecture, creating a model that learns to produce consistent explanations. The following notation is used to describe models and their explanations. A machine learning model m_i is passed an input x to produce an output o, such that $m_i(x) = o$. For classification tasks, the final prediction p of m_i is then the class with the highest predicted probability $\operatorname{argmax} m_i(x) = p$. An explanation for input x on model m_i is given as $E_{m_i}(x)$, with the importance value for a given feature $x_{j,k}$ given as $E_{m_i}(x_{j,k})$.

5.1.1 Post-processing Technique

In signal processing, it is common to reduce the amount of noise present in a measurement by taking a number of replicated measurements and averaging them out, in a process called signal averaging [167]. The following post-processing technique takes inspiration from signal averaging - the two problems are similar in that the spurious features in the explanations can be seen as the noise, with our goal being to maximise the "signal" (i.e. importance) of the causal features (and minimise the importance of the spurious features). This results in a method which takes the explanations from P models (each of the same architecture, but trained with different hyperparameters), take the average of these explanations and then apply some thresholding to the resulting explanation (with the purpose of the thresholding being to include only those features that are most important across all P models).

However, to be able to effectively apply this technique, one must consider how the thresholding is applied (is it applied pre- or post-averaging?), how is the threshold level chosen, and how should cases where the P models do not agree on the prediction be handled (two models, m_1 and m_2 that produce different outputs on the same input will likely have wildly different explanations). The final question is the simplest to answer - the method should take the most prevalent class as the final prediction p and only include explanations from models where $\operatorname{argmax} m_i(x) = p$, similarly to ensemble models.

When considering whether to apply thresholding pre- or post-averaging, it is prudent to first consider when it is easiest to determine the threshold level. If this were to occur after averaging, the decision becomes much more difficult - one would have to consider how the averaging of the explanations may affect the distribution of explanation values. If instead the thresholding is performed pre-averaging, then a technique can easily use the distribution of each individual model's explanations to determine a threshold.

This leads onto how to choose the threshold level. Initially, experiments focused on using a single value t that could be tuned as a hyperparameter, such that any feature x_i with importance $E_{m_i}(x_i) \leq t$ is set to 0. However, this results in a number of problems - most importantly, it is extremely hard to tune as the appropriate value can vary vastly depending on the task (and by extension, the dataset) learnt by the model. Instead of this hand-tuning approach, I propose a probability density function (PDF) based approach, allowing for a much simpler choice of threshold - one can choose to remove any feature which has probability $\leq t$ of appearing (in essence, removing the lower t percentiles of features). Note, however, that when using this technique one must also choose how to estimate the PDF of the explanations as accurately as possible. For this I propose using Kernel Density Estimation (KDE) with a Gaussian kernel, although it should be noted that the performance of KDE depends a lot on choosing the right hyperparameters [168]. In this scenario, Scott's rule of thumb [169] is sufficient for choosing an optimal value for the bandwidth; this has been shown to work extremely well for distributions that are normal (which, given a large enough number of samples, we can assume model explanations will be), although is also surprisingly robust in cases where the underlying distributions are non-normal too [170].

This results in the following end-to-end post-processing technique:

1. Train P separate models (of the same architecture, but with different training hyperparameters) on the task

2. On the test set X, compute an explanation $E_{m_i}(X)$ for each model $m_1, ..., m_P$

3. Use KDE to estimate the PDF for each set of explanations $E_{m_i}(X)$

4. Use the PDFs/CDFs to determine a threshold t such that $P(x \leq X) = p$ (where p is the desired proportion of features we wish to ignore)

5. Use the calculated threshold t to threshold the explanations $E_{m_i}(X)$:

$$E_{m_i}(X) = \begin{cases} E_{m_i}(X) & \text{if } E_{m_i}(X) > t, \\ 0 & \text{otherwise} \end{cases}$$

The efficacy of this method is discussed in Chapter 5.2.3, however there are both advantages and disadvantages to this method which can be discussed without the need for quantitative results. The process is easy to follow, and can be easily applied to any model architecture and explanation technique. Yet, a large part of the method's effectiveness relies on the accuracy of the KDE calculation - which can be very difficult to discern without an accurate ground truth to compare to. Furthermore, the threshold probability p must be chosen by hand, and there is no easy way to do so without trial and error - the ideal p will vary greatly between tasks and even different model architectures trained on the same task. This choice is made even more difficult by the fact that there is no single metric that can be used to determine how well a chosen value performs, as there needs to be a balance between having a high explanation consistency whilst still retaining explanations of the important features (e.g. setting $p = 1.0$ will result in perfect explanation consistency, but will set the entire explanation to 0). An ideal solution would retain the generalisability of this one whilst reducing the need for by-hand trial and error hyperparameter tuning, and also ensuring the explanations remain useful and faithful to the model. Furthermore, as a post-processing technique, this method does not actually fully address the issue of inconsistent models - ideally, a full solution would result in models whose learned features are consistent. In the following section, I introduce a completely novel model architecture and training algorithm that utilises some of the ideas from this post-processing technique to create models that (as we will see in Chapter 5.2) learn consistent features.

5.1.2 Deep Explanation Ensembles

Deep Explanation Ensembles (DEE) are a novel ensemble architecture that improves explanation consistency. As Chapter 4 showed, ensemble models do slightly increase the explanation consistency of their sub-models. Furthermore, as explored in Chapter 2.3.2, it has been frequently shown that ensemble architectures out-perform non-ensemble models, reduce the risk of overfitting and perform more complex classification tasks than would be possible with a single model alone [171]. More complex architectures have been shown to be more robust, be less susceptible to adversarial attacks and allow for better uncertainty quantification [20,25,172]. In particular, hyperparameter ensemble models have recently been proposed, wherein the ensembles not only combine weight diversity, but also hyperparameter diversity [25] - however, despite improving in many areas upon baseline models, we saw in Chapter 4 that these models do not show any significant improvement in explanation consistency.

However, in this section I have combined a modified ensemble architecture with a unique training procedure to create a model that produces consistent explanations by considering explanations from a wide set of models trained with differing hyperparameters. The final model is encouraged to use only important features that are shared between every model trained, resulting in a fully trained model that produces consistent explanations.

The core idea of this new architecture is that each ensemble consists of S sub-models $e_1, ..., e_S$, each of which is trained with a different hyperparameter setup. Note that only the random seed or order of the training set should be changed; hyperparameters such as learning rate and hidden layer size should remain identical across all S sub-models. The S sub-models are trained in tandem, with the loss function designed to force each e_i to learn to use similar features (this is described in more detail in Chapter 5.1.2). The final explanation ensemble model is then an average across all sub-models, such that $E(x) = \frac{\sum_S e_i(x)}{S}$. This section describes the explanation ensemble architecture in more detail, including the training process and discriminator that allows the sub-models to learn similar features.

The Explanation Ensemble Discriminator

The aim of an explanation ensemble is to make each of the S sub-models to learn to use a similar set of features, with this being achieved through the training of a discriminator D. If the S sub-models cover a wide range of hyperparameters, then one would expect that they will cover a wide range of learned features (this is follows from the results of inconsistent explanations shown in [173]), and as such the final model will have learned to ignore a large set of noisy (i.e. spurious) features. These two models are trained in tandem, in a minimax two-player game: the goal of D is to learn how to discern between real and fake samples while the goal of G is to learn the features of the true data distribution in order to fool D into making incorrect classifications.

I propose to use a discriminator D in the training of the ensemble model, which is trained on the explanations from the ensemble sub-models; the purpose of this discriminator is to then classify which of the S sub-models the explanation originated from. The goal of the training of the S sub-models is then to modify their weights such that the generated explanations then fool D into making incorrect decisions (whilst still balancing the final accuracy of the sub-models too). The exact details of this training process are described in Chapter 5.1.2.

The proposed discriminator D is a simple Multi-Layer Perceptron (MLP) with 1 hidden layer: there is an input layer (of the same size as the data samples), 1 hidden layer of size 32, a ReLU activation and finally an output layer (of size S, the number of sub-models). This discriminator joins S sub-models to create the whole explanation ensemble model, where each of the S sub-models can be of any architecture suited to the base task at hand (e.g. an MLP for classification or regression). Figure 5.1 shows an overview of our explanation ensemble architecture.

Explanation Ensemble Training

The training for explanation ensembles is the most important aspect of the model - there are a number of conflicting goals that it is aiming to achieve, and it is imperative that the training is setup in such a way that each of these goals can be achieved whilst also ensuring the model is easy to train. There are two objectives of the training process: **1)** maximise model accuracy on the task at hand, and **2)** minimise the difference between generated explanations of the S sub-models (i.e. maximise the error of D) - the resulting ensemble model should then have high performance/accuracy and, as the final feature importance values have been learnt across S different hyperparameters (and thus "averaged-out"), high(er) explanation consistency. Summarising these two objectives leads to the following loss function for the explanation ensemble

$$\text{loss} = \sum_i \texttt{CELoss}(m_i(x), y) - \beta \cdot \texttt{CELoss}(D(E_i(x)), i) \tag{5.1}$$

where $\texttt{CELoss}(\cdot, \cdot)$ is cross-entropy loss, y are the ground truth labels for the training task and $\beta \in [0, \infty)$ is a hyperparameter for specifying the weight the discriminator plays during training. For all experiments in this chapter, β is set

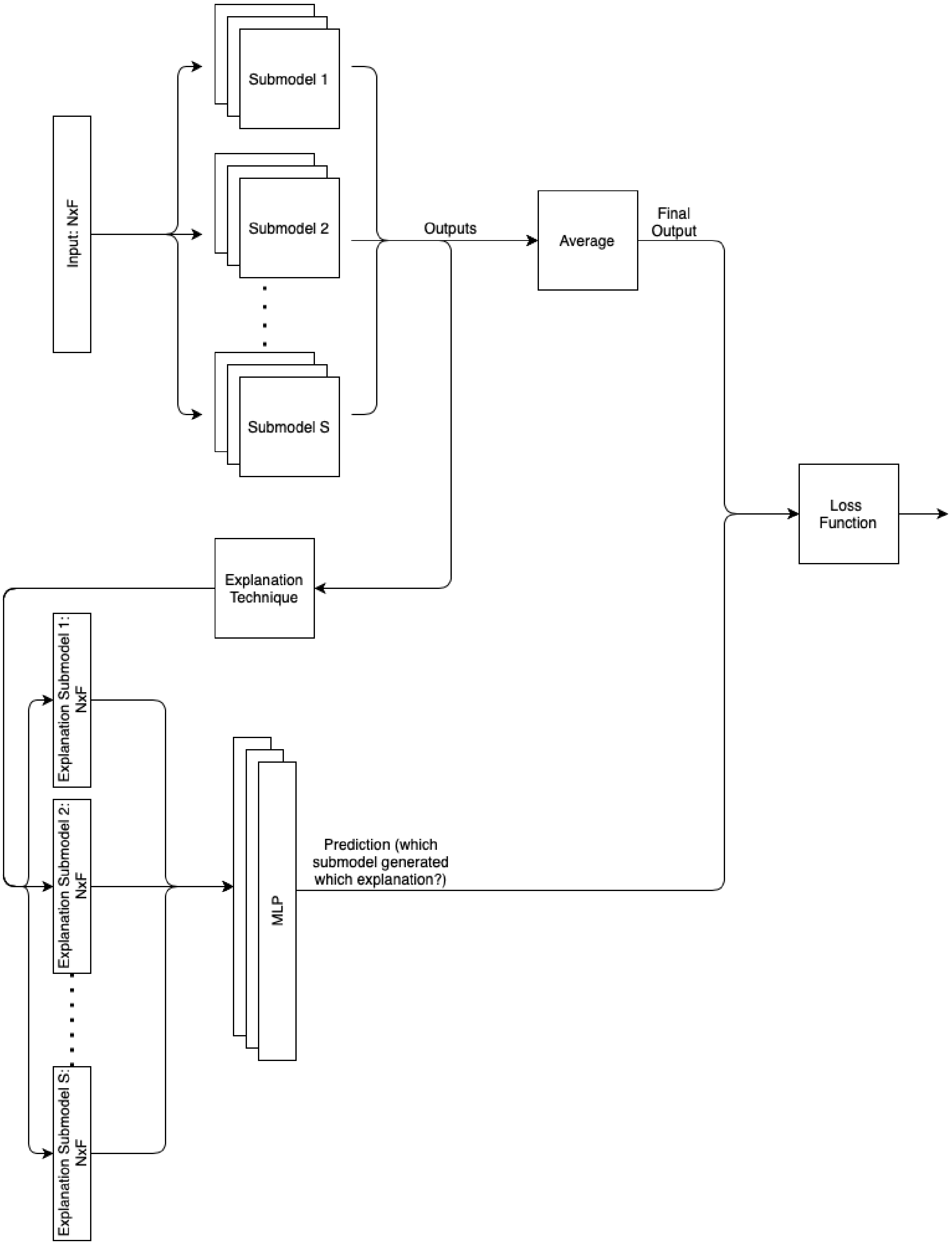

Input: NxF
Submodel 1
Submodel 2
Submodel S
Outputs
Average
Final Output
Loss Function
Explanation Technique
Explanation Submodel 1: NxF
Explanation Submodel 2: NxF
Explanation Submodel S: NxF
MLP
Prediction (which submodel generated which explanation?)

Figure 5.1: Diagram of our explanation ensemble architecture and data flow

such that the two parts of the loss function have the same order of magnitude. This loss function requires that explanations are generated for each sub-model in each training epoch; any explanation technique (within the limits of the computational power available: many explanations techniques are too computationally intensive to make them viable options to be calculated across the whole training set S times each epoch) could be used here.

Equation (5.1) describes how the explanation ensemble model learns to fool the discriminator while minimising the classification (or regression, or other task-specific) loss. During an epoch where this loss function is used, only the weights of the S sub-models are updated - the discriminator remains the same. Thus, every n epochs *just* the discriminator D *alone* is trained (without back-propagating through the sub-models), allowing the discriminator to learn how to accurately classify which sub-model a given explanation was calculated from. Chapter 4 shows that, for many (if not most) tasks, this explanation classification task is easy for an ML model to learn to a high degree of accuracy (in fact, this is a direct result of the fact that ML models so far have shown low levels of explanation consistency) and so D is able to learn how to do so in a single epoch. To summarise, the general training process of an explanation ensemble is as follows, and is formally detailed in Algorithm 1:

1. For each $i \in [S]$ run $m_i(x)$ with the correct hyperparameters (i.e. training seed)

2. Calculate the explanations $E_i(x)$

3. If $e \mod n = 0$ update the discriminator D using the loss $\texttt{CELoss}(D(E_i(x)), i)$, where e is the current epoch

4. Otherwise, update each of the S sub-models according to the loss function in Equation (5.1)

Training of the proposed architecture is inherently unstable; for instance, the loss of the discriminator is minimised if every feature in the data is given the same importance value - however, for this to be possible each of the sub-models must necessarily be outputting the same class, regardless of the input x. This leaves n as a hyperparameter that can be optimised (e.g. using a grid-search), though as an initial starting point $n = 2$ has been found to result in stable training across all experiments.

Algorithm 1 Explanation Ensemble Training

```
for $e \in [0..\text{epochs}]$ do
    for $x, y \in$ batches do
        for $i \in [S]$ do
            random_seed ← random_seeds[i]
            data_order ← data_orders[i]
            $outputs \leftarrow m_i(x)$
            $explanations \leftarrow E_i(x)$
        end for
        if $e \mod n = 0$ then                                  ▷ $n$ is input as a hyperparameter
            loss = $\text{CELoss}(D(E_i(x)), i)$                    ▷ $\text{CELoss}(\cdot, \cdot)$ is cross-entropy loss
        else
            loss = $\sum_i \text{CELoss}(m_i(x), y) - \beta \cdot \text{CELoss}(D(E_i(x)), i)$     ▷ $\beta \in [0, \infty)$
        end if
    end for
end for
```

5.1.3 Explanation Computation

To generate explanations for all models tested, SHAP [21] values across the whole dataset are calculated. As discussed in Chapter 2.2.3, SHAP is highly versatile and can be applied to any data modality; alternative feature attribution methods such as Grad-CAM and Information Bottleneck Attribution [174] are restricted to certain data types. The methods presented in Chapter 4.2 are used to calculate the explanation consistency for these models.

5.1.4 Alternative Explanation Consistency Calculations

As was discussed in Chapter 4.2.2, there are a number of other methods that can be used to measure the consistency of the model explanations. To further explore where different consistency measures may be applicable, as well as using binary LR classifiers to measure explanation separability, I also approach the problem from an information theoretic background, using statistical distance measures to quantify the difference between the produced explanations. Being symmetric, smooth, and bounded Jensen-Shannon Divergence (JSD) is aptly suited to this task [173, 175], allowing the comparison between the probability distributions of the explanations for two models. The main disadvantage of this technique is that JSD is only defined for probability distributions, whereas we only have a finite number of samples for each model's explanations. To alleviate this issue, Kernel Density Estimation (KDE) is performed on the explanations from a model to estimate the probability density function. For each dataset/task pair, KDE is ran on the explanations for each model (both baseline and explanation ensemble models). Then, for each pair of baseline models and each pair of explanation ensembles (for a given task), the JSD is calculated, with higher values indicating the two sets of explanations are dissimilar. This can be used to calculate the JSD consistency of the explanations

$$C_{JSD} = 1 - \frac{\sum_{(a,b)} J(a \parallel b)}{\alpha} \tag{5.2}$$

where $J(a \parallel b)$ is the JSD between the explanations of model a and model b.

5.1.5 Explanation Quality Metrics

To test the faithfulness of the explanations to the models (that is, to ensure that the explanations are accurately describing the changes in the model), I use explanation sensitivity, explanation infidelity and explanation accuracy from Chapter 2.2.6, where each quality metric was chosen as they measure faithfulness in different ways.

5.1.6 Statistical Hypothesis Testing

As well as reporting the results for both performance and consistency the statistical significance of the results is also investigated by performing statistical hypothesis tests on both the model performance results and the explanation consistency results. Note that one cannot assume that the data (i.e. the performance metrics and explanation consistency) is normally distributed, and so parametric tests such as Student's t-test are not viable. Similarly, one cannot assume that the distribution of the differences between the baseline ensembles and explanation ensembles are symmetric and so the Wilcoxon Signed Rank test would also be invalid. Instead, a non-parametric version of these tests must be used - specifically, the Mann-Whitney U test is used, setting the null hypothesis H_0 as the two distributions being equal.

Both the test statistic U and the corresponding p-value are calculated for each dataset, comparing both the performance metric and the separability between the baseline ensembles and explanation ensembles. The hypothesis tests are performed at the $\alpha = 0.05$ significance level, meaning that the null hypothesis H_0 will be rejected if $p < 0.025$ (using a two-sided version of the Mann-Whitney U test).

5.1.7 Ablation Study

Like any ensembling technique, explanation ensembles are more computationally expensive during both training and inference time than traditional models, and that this may have an impact on their use in production environments [176]. It is also important to determine that all parts of the proposed technique are critical to the end result, and that improved explanation consistency is not a result of a single part of the system. Three post-training methods that attempt to address this issue are evaluated: submodel averaging, random sub-model selection, and a combination of checkpoint and submodel averaging. Checkpoint averaging is a weight averaging technique that has been shown to lead to better model generalisation [34]. Checkpoint averaging is performed (by taking the 10 most recent saved checkpoints at the end of training) on both the baseline models and the normal ensemble models, calculating the explanation consistency for these two techniques as detailed above. In submodel averaging, one creates a single model by averaging the model weights of each of the n sub-models trained in the explanation ensemble - this results in just one model that will be much quicker to run at inference time. In random submodel choosing, one simply picks one of the sub-models of the explanation ensemble at random to use at inference time; with the intuition being that, as the model has still been trained to produce explanations similar to those of the other $n - 1$ sub-models, it should still produce better explanations than traditionally trained models. Furthermore, the checkpoint averaging technique that has been tested on normal architectures is also combined with submodel averaging.

5.2 Results

To thoroughly test the ability of the proposed explanation ensemble model to improve the consistency of the produced explanations, models are trained on 6 different tasks on 4 distinct healthcare/biological datasets. First, these tasks and datasets are briefly re-introduced (having first been fully introduced in Chapter 2.1), explain the motivation behind the inclusion of each dataset, then report the results of the experiments.

5.2.1 Datasets and Base Model Architectures

In an effort to keep these initial experiments as simple and interpretable as possible, experiments are limited to tabular datasets. Decisions based on tabular data are inherently easier to understand and explain - there are a (typically small) number of distinct features, and often these features will be well understood by domain experts. In contrast, features (and thus explanations) of more complex data modalities are harder to define. For example, in an image each individual pixel is a feature and yet humans (and indeed many ML models) will utilise superpixels (groups of pixels) when making decisions. This makes explanations on these data types more difficult to analyse. It also introduces difficulties when comparing the explanations of two different samples - in tabular data, feature importance values can be directly compared, whereas this comparison is difficult to accurately define as the features between most other data modalities are not necessarily aligned. For these reasons, initial evaluations in this study are limited to measure the effectiveness of our proposed methods on tabular data, and leave investigations on other data types to Chapter 6.1.

Deep learning models are being increasingly used to analyse Electronic Health Record (EHR) datasets for the prediction of mortality, phenotyping, de-identification and other related tasks [177]. Further examples of tabular dataset come from genome analysis, on tasks such as pattern identification and kingdom classification [178]. The application of ML to both of these areas also rely heavily on model interpretability, and the trust of domain experts (e.g. clinicians and biologists) [9], and so by extension consistent explanations from models are imperative. The purpose of this chapter is to investigate the (in)consistency of explanations produced by models on these datasets, and inspect whether our proposed explanation ensemble architecture improves upon the consistency. Therefore, for each dataset, a state-of-the-art neural network for the given dataset is re-implemented and used as the base model for the explanation consistency experimentation. This results in a *base model architecture* for each dataset/task which forms the basis of the experimentation. These base model architectures are then taken and used as sub-models to train a *normal ensemble architecture*, as well as the proposed *Deep Explanation Ensemble (DEE) architecture*. This allows comparison of the proposed network with both a standard baseline and an ensemble baseline. A summary of the datasets, tasks and baseline models (and hence ensemble sub-models) used can be found in Table 5.1.

EHR Datasets

Three different EHR datasets are used, all of which are first introduced in Chapter 2.1 - here they are briefly re-introduced alongside descriptions of the baseline models used as sub-models for the DEEs and as used for baseline comparisons. The **Breast Cancer Wisconsin (BCW)** dataset [55] is a small, classical ML dataset that has been used frequently as a baseline test for the performance of ML models on healthcare data. Each entry in the BCW dataset consists of a set of features extracted from a digitized image of a fine needle aspirate of a breast mass, with the features describing: radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal dimension. The aim of the task is to train a classification model to predict which tumors are malignant. Following the results of [179], a small Multi Layer Percetpron (MLP) is used for this classification problem. The MLP consists of an input layer, a hidden layer of size 40, a second hidden layer of size 15 and then the output layer; the ReLU activation function is used, with LogSoftmax being used on the output of the final layer. The model is trained over 14 epochs, with a learning rate of 0.001, batch size of 64, Negative Log Likelihood (NLL) loss and the Adam optimiser.

The second EHR dataset used is **KAIMRC**: a private EHR dataset collected from King Abdulaziz Medical City located in the central and western regions of Saudi Arabia [28]. The dataset spans 2016 to 2018, and includes patient demographics (e.g. age and Body Mass Index), lab results (e.g. cholesterol levels) and vital signs during this period. For a detailed description of the features included in the dataset, and their clinical relevance, we refer the reader to [28]. The dataset was collected to aid the development of ML models for diabetes prediction. This dataset is used for two separate, albeit related, tasks. **1)** To train a classifier to predict patients with elevated HbA1c levels using longitudinal data, and **2)** To train a regression model to predict HbA1c levels. The KAIMRC classification task is similar to the BCW task in that it is a binary classification problem, but the KAIMRC dataset is much larger and more complex than BCW and thus has been chosen to evaluate our proposed explanation ensemble models on real-world datasets. Similarly, the KAIMRC regression task is used to verify the proposed deep explanation ensemble methods work on regression as well as classification. The methods presented in [28] are followed to create the baseline MLP models. The KAIMRC classification MLP uses 3 hidden layers of sizes 48, 48, and 24 respectively, using ReLU activation functions after each hidden layer and Sigmoid on the output. Mean-squared error (MSE) was used for the loss function and the Adam optimiser was used. The KAIMRC regression model follows the same general structure, with dropout with probabilities 0.2 and 0.1 after the first and second hidden layers respectively.

The final EHR dataset used is **MIMIC-IV** [180]. MIMIC-IV is a large, freely-available medical dataset collected from the critical care unit of Beth Israel Deaconess Medical Center from 2008 to 2019. MIMIC-IV contains patient information (e.g. age, weight, height, comorbidities), lab events (e.g. cholesterol, creatinine, bilirubin, HbA1c levels), vital signs and medication prescribed of 383,220 patients. MIMIC-IV is a time-series dataset and as such each record (e.g. patient) will have a different number of features, and the exact features present for each record will vary. The `flexible-ehr` framework [181] is used to train a model for mortality prediction. `flexible-ehr` consists of an embedding layer (embedding the input to a layer of size 32) followed by a Long Short-Term Memory (LSTM) module (with a hidden dimension of size 256), which is then passed into an MLP (with 4 hidden layers of sizes 32, 64, 128, and 256) [182]. The setup and hyperparameters suggested in the original paper are followed exactly, and are reported in Table 5.1. This dataset and model architecture not only allows the evaluation of the proposed method's ability to perform on very large-scale datasets, but also the effectiveness of explanation ensembles on complex sub-model architectures; all other experiments in this paper use MLPs of varying layouts, whereas `flexible-ehr` is a much more complex architecture consisting of an embedding layer, LSTM and MLP.

Genomics Datasets

One genomics dataset is utilised for two different tasks. The **codon usage dataset** [47] consists of the usage frequency of 64 codons for more than 13,000 organisms. The methods presented in [47] are followed to train two different models; one to classify the organisms kingdom (from 5 distinct classes), and the other to classify the DNA type of the organisms (from 3 possible classes). The same data pre-processing (removing organisms with less than 1000 codons and those with DNA types in categories 2 or higher) is performed, resulting in 12,964 samples in the final dataset. As per their methods, both MLPs consist of a single hidden layer with 9 neurons. The purpose of evaluating the proposed techniques on these two tasks is to evaluate the performance of explanation ensembles on multi-class classification problems (whereas previous classification-based experiments are exclusively binary classification problems).

	Dataset Descriptors				**Baseline Model Hyperparameters**		**Baseline Training Hyperparameters**		
Dataset Name	**Task**	**Num. Samples**	**Num. Features**	**Num. Classes**	**Model Architecture**	**Num. Hidden Layers**	**Num. Epochs**	**Learning Rate**	**Batch Size**
Breast Cancer Wisconsin	Binary classification	569	10	2	MLP	2	14	0.001	64
KAIMRC	Binary classification	18,844	24	2	MLP	3	14	0.001	32
KAIMRC	Regression	18,844	24	N/A	MLP	3	14	0.001	32
MIMIC-IV	Binary classification	383,220	N/A	2	`flexible-ehr`	4	20	0.0005	128
Codon Usage (Kingdom)	Multi-class classification	12,964	64	5	MLP	1	16	0.0001	32
Codon Usage (DNA)	Multi-class classification	12,964	64	3	MLP	1	20	0.0001	32

Table 5.1: Summary of the dataset and tasks used to evaluate Deep Explanation Ensembles alongside baseline model and training hyperparameters. Note that MIMI-IV is a time-series dataset and so each entry will have different numbers of features, and the KAIMRC (Regression) task has no target class as it is a regression problem.

5.2.2 Model Performance Results

Multiple versions of each baseline model are trained and then inspected to investigate how changing their training hyperparameters affects model performance and explanation consistency. For each training task the training hyperparameters are systematically changed, changing only one hyperparameter at a time, in order to isolate the affect of each change. The experiments are designed to investigate both changing the random seed and training set order. For each task 10 models are trained with the same random seed but different training set orders, and another 10 models with different random seeds but the same training set order. Each model is given the same train/test split - it is only the order the training set is passed to the model that is changed.

Traditional ensemble models are also trained on each classification task. Each ensemble consists of 10 sub-models, using the same architectures described in Chapter 5.2.1. The results of these models are compared and contrasted with those from the explanation ensembles in order to discern whether any changes in model performance/consistency originates from the use of the general ensemble architecture or the proposed specific explanation-based architecture.

The proposed deep explanation ensemble architecture is also trained, with the baseline model architectures for each task used as the deep explanation ensemble sub-model as detailed in Chapter 5.2.1. Initially, 10 sub-models per ensemble are used. As detailed in Chapter 5.1.2, the discriminator is trained on alternate epochs and with a low learning rate of 0.00001. Qualitative experiments show that β should be set such that the discriminator loss is one order of magnitude less than that of the classification loss, and so $\beta = 0.1$. 10 explanation ensemble models are trained with different random seeds (but keeping the training set order the same) and 10 models are trained with different training set orders (but the same random seed). The experiments are repeated 3 times (with different seeds/orders) to allow for the calculation of standard error (noting that the availability of compute time limited the scale of these experiments). The performance of the models and the consistency of the explanations are recorded and compared with the results from the base models. Similarly to the experiments on the normal ensembles and baseline models, checkpoint averaging of explanation ensembles is also evaluated.

Tables A.1 to A.4 report the performance metrics and hyperparameters used for each individual baseline model trained, for the KAIMRC, BCW, Codon Usage and MIMIC-IV datasets respectively. These results are compared to the current state-of-the-art results for each dataset. The reason for this is twofold: 1) it ensures that when the explanation ensembles are compared to the baseline models one can easily compare it to state of the art models, and 2) it will help to confirm that any explanation inconsistency present in the baseline models are not the result of improper training. Table 5.2 shows a summary of the performance of the baseline models, alongside the variation in performance when training hyperparameters are changed. It also highlights how all of the baseline models achieve equal (or near-equal) levels of performance compared with the current state of the art for their respective tasks.

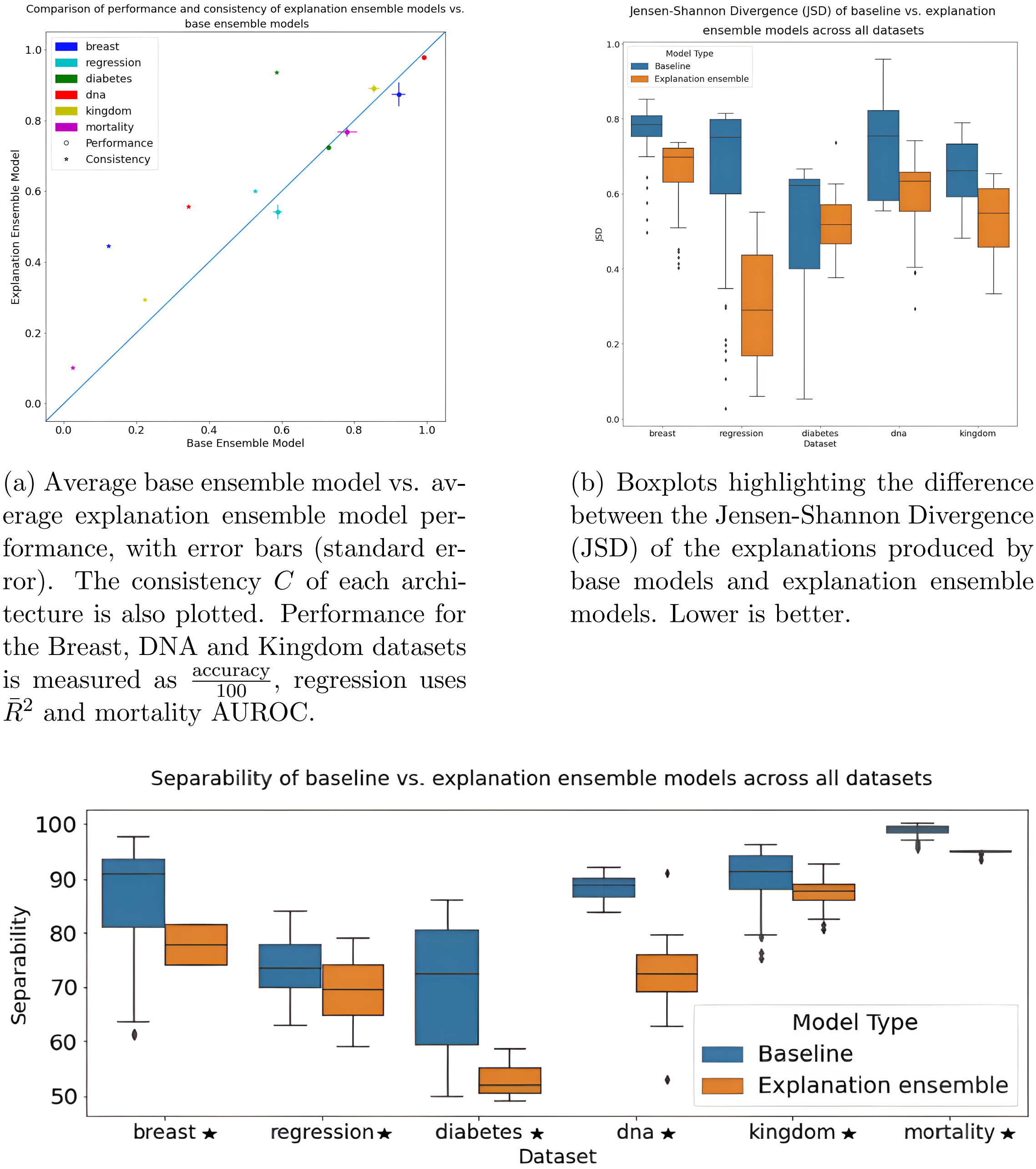
Dataset (Task)	**Seed**	**Shuffle**	**Overall**	**SotA Performance**
KAIMRC (Classification)	82.576 ± 0.4668	83.381 ± 0.1811	83.12 ± 0.4779	83.22
KAIMRC (Regression)	0.5927 ± 0.0113	0.579 ± 0.0122	0.5858 ± 0.01326	n/a
BCW	92.185 ± 1.7315	91.5 ± 2.8319	91.843 ± 2.3172	99.04
Codon Usage (Kingdom)	85.280 ± 1.8029	85.38 ± 1.1778	85.33 ± 1.4367	84.25
Codon Usage (DNA)	99.268 ± 0.0950	99.166 ± 0.0921	99.217 ± 0.1033	99.15
MIMIC-IV	76.362 ± 2.5808	79.736 ± 1.8906	78.049 ± 2.7769	84.72

Table 5.2: Summary of mean accuracy/R^2 (± standard deviation) for the baseline models when the seed and training set order is changed during training. The state of the art (SotA) model performance is also reported to confirm the models are properly trained.

Model Architecture	**Dataset (Task)**	**Seed**	**Shuffle**	**Overall**
Normal Ensemble	KAIMRC (Classification)	83.244 ± 0.2367	83.212 ± 0.0920	83.228 ± 0.1702
	KAIMRC (Regression)	0.51 ± 0.01673	0.524 ± 0.03007	0.517 ± 0.02532
	BCW	77.890 ± 11.563	71.736 ± 11.466	74.813 ± 11.330
	Codon Usage (Kingdom)	90.134 ± 1.6527	90.568 ± 1.2715	90.351 ± 1.4088
	Codon Usage (DNA)	99.150 ± 0.2141	99.122 ± 0.2270	99.136 ± 0.2086
	MIMIC-IV	77.23 ± 0.7935	75.96 ± 0.6299	76.60 ± 0.7520
Explanation Ensemble (Ours)	KAIMRC (Classification)	72.173 ± 0.3998	72.423 ± 0.6856	72.298 ± 0.5365
	KAIMRC (Regression)	0.5504 ± 0.0181	0.5408 ± 0.0243	0.5515 ± 0.0197
	BCW	87.824 ± 4.0860	86.783 ± 2.5061	87.361 ± 3.3173
	Codon Usage (DNA)	98.032 ± 0.5420	97.6523 ± 0.5411	97.863 ± 0.5447
	Codon Usage (Kingdom)	89.176 ± 1.1797	89.055 ± 1.0404	89.110 ± 1.0497
	MIMIC-IV	77.338 ± 0.000083667	77.32 ± 0.0001225	77.329 ± 0.0001370

Table 5.3: Summary of mean accuracy/R^2 (± standard deviation) for the normal ensemble models and explanation ensemble models (ours) when the seed and training set order is changed during training.

Similarly, the performance of the baseline "normal" ensemble models is reported in the same way. Tables A.5 to A.7 list the accuracy of each individual model trained, and the hyperparameters used during training. A summary of the spread of performance of the baseline, normal ensemble models is shown in Table 5.3 - by comparing this table with the results in Table 5.2, one can see that the normal ensemble models neither improve nor degrade performance when compared to our baselines and the current state of the arts.



(a) Average base ensemble model vs. average explanation ensemble model performance, with error bars (standard error). The consistency C of each architecture is also plotted. Performance for the Breast, DNA and Kingdom datasets is measured as $\frac{\text{accuracy}}{100}$, regression uses $\bar{R}^2$ and mortality AUROC.

(b) Boxplots highlighting the difference between the Jensen-Shannon Divergence (JSD) of the explanations produced by base models and explanation ensemble models. Lower is better.

(c) Boxplot showing the separability of base model explanations vs. explanation ensemble explanations. Lower separability values indicate the binary classifier found it more difficult to distinguish between the explanations from two models trained with different hyperparameters, and results in higher explanation consistency. Lower is better. Stars indicate datasets where the difference between the two architectures is statistically significant, following the results of a Mann-Whitney U test.

Figure 5.2: Figures comparing the explanation consistency C (**a**), JSD explanations consistency C_{JSD} (**b**), and explanation separability (**c**) between baseline models and our proposed explanation ensembles across all tasks tested.

The performance of the baseline models is compared against the performance of explanation ensemble models trained on the same task. Tables A.8 to A.11 report the individual performance of our explanation ensembles, alongside the hyperparameters used during training. Table 5.3 summarises these results, and also shows the degree of variation when training hyperparameters are changed. This information is summarised in Figures 5.2a and 5.3, which highlight the differences in spread and location of model performance when training hyperparameters are changed. For all datasets the mean explanation ensemble performance is always within a 10% range of the base model performance; although this does represent a slight decrease in model performance when explanation ensembles are used, I argue that this is only a slight decrease that would be worth the trade-off given that explanation consistency is significantly improved.

5.2.3 Post-processing Technique Consistency Results

Figure 5.4 show the general trend of explanation consistency as the threshold value p is increased, across all datasets. As expected, the explanation consistency increases as the threshold value p increases; the larger the p value the more feature attributions the post-processing technique will remove and so one would expect the consistency to increase. This is further supported by the hypothesis that features with lower attribution scores from explanation techniques are typically the noisy features (i.e. non-causal) learned by models, and that it is in these low-valued features where much of an explanation's inconsistency stems from [173] - removing these noisy features will then necessarily increase explanation consistency.

However, Figure 5.4 also shows that the standard error of the results is very high across most datasets. As the post-processing technique heavily relies on the results of kernel density estimation, the performance of the technique heavily relies on the performance of the KDE. KDE is known to be extremely sensitive to the choice of hyperparameters (namely, bandwidth) [170], with the quality of the computed PDF being very dependent on the chosen bandwidth and underlying data. As the proposed technique chooses KDE bandwidth based on a rule-of-thumb (refer back to Chapter 5.1.1 for more details), it is possible that some experimental setups are

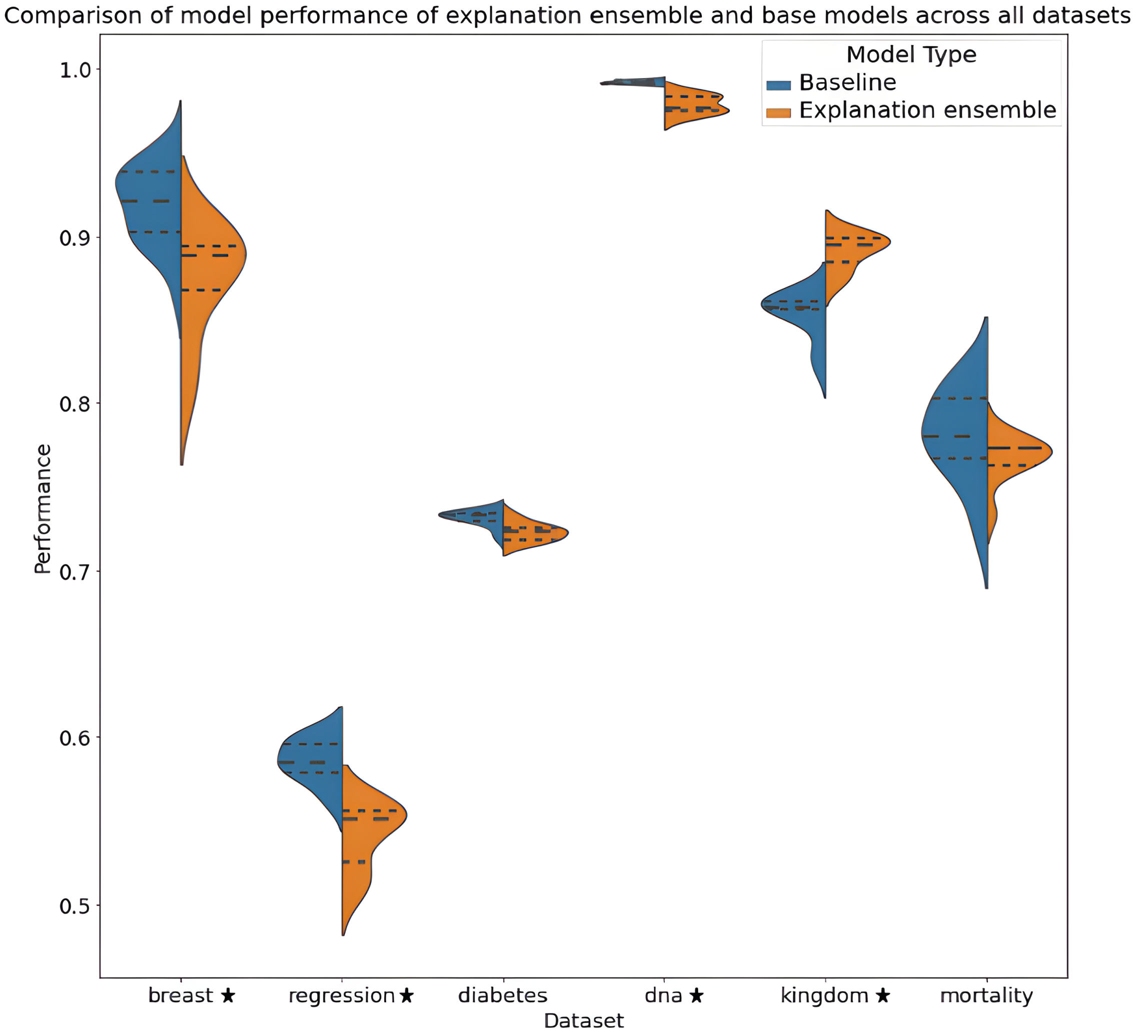


Figure 5.3: Violin plots showing distribution of model performance across all datasets, for both the base and explanation ensemble architectures. The dashed lines represent the 25th, 50th and 75th quantiles respectively. Performance for the Breast, DNA and Kingdom datasets is measured as $\frac{\text{accuracy}}{100}$, regression uses $\bar{R}^2$ and mortality AUROC. Stars denote datasets where there is a statistically significant difference in the two architectures, following the results of a Mann-Whitney U test.

inadvertently using poor choices of bandwidth (and that this is in turn resulting in varying results from our post-processing technique). This is an inherent limitation of the post-processing technique, and is one of its disadvantages over the proposed explanation ensemble architecture. Furthermore, the post-processing technique uses explanations from multiple independently trained models, whereas the sub-models of explanation ensembles are trained together (albeit with different seeds) - this is also likely to introduce much more variance to the method.

Furthermore, one must consider the quality of the resulting explanations - it is

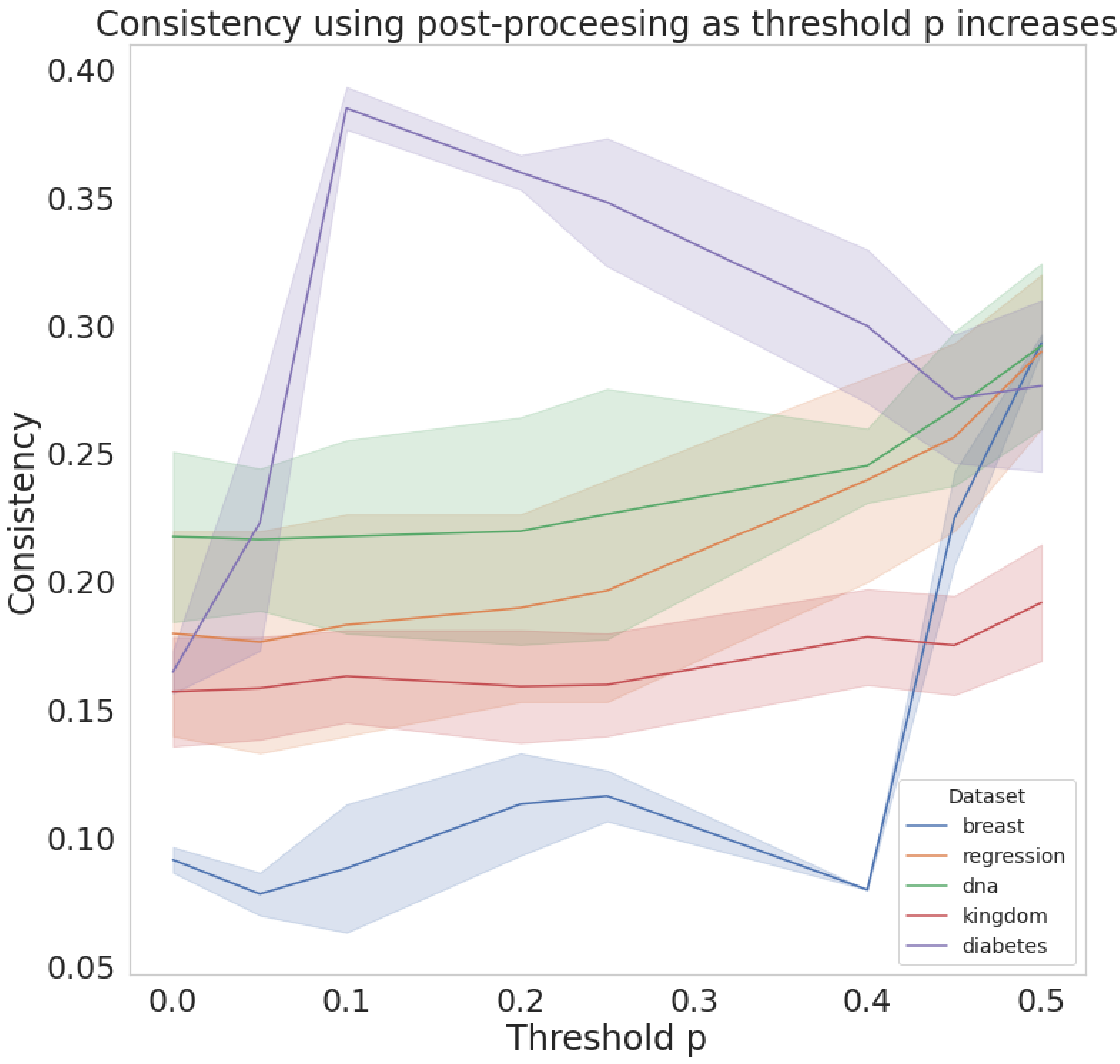


Figure 5.4: Graph showing the increase in explanation consistency (± standard error) as the threshold p for the post-processing technique is increased (i.e. as more of the original explanation is subject to thresholding)

not necessarily the case that explanations with high consistency are good explanations, nor faithful to the underlying predictive model. This is particularly the case with this technique as, unlike in the explanation ensemble method, we are modifying the final explanations without any consideration of the underlying models nor their predictive power. This post-processing technique has a significant affect on how well a human is able to perceive the final explanations - although higher threshold values p result in higher explanation consistency, they also result in much more of the explanation being removed. This necessarily affects one's ability to interpret the explanation, and in extreme cases will make the explanation useless. The explanation ensemble technique does not suffer from this problem, as it does not rely in the thresholding/removing of feature attributions and instead trains multiple models to learn features consistent to one another.

5.2.4 Explanation Consistency Results

Table 5.4 reports the consistency, C (Equation (4.2)), on all tasks tested - C is calculated for all training variations and architectures. Table 5.4 shows that the proposed Deep Explanation Ensemble architecture significantly improves the consistency of the produced explanations.

Table 5.4 also shows that the degree to which explanation consistency improves varies greatly on the dataset/task - for example, the Codon Usage Kingdom classification task sees an increase of only 0.07167 whereas the KAIMRC classification task sees and increase of 0.35417. I hypothesise that this is due both to differences in the dataset and differences in the baseline model (and thus also the explanation ensemble sub-models) architectures. The KAIMRC dataset consists of only 2 classes and 24 features, whereas the Codon Usage Kingdom classification task has 5 classes and 64 features; intuitively, one would expect it would be easier for the explanation ensemble models to learn consistent features for the smaller, simpler KAIMRC dataset than the Codon Usage dataset.

Figure 5.2c demonstrates the difference in spread of the mean separability, $S_{(a,b)} = 2 * |M_{(a,b)} - 0.5|$, between each individual training variation tested. This allows for a more fine-grained analysis of the explanation consistency than the high-level summary that explanation consistency C provides, noting that the higher the separability the worse the results. Figure 5.2c shows that the mean separability of explanation

Dataset (Task)	Base Model C	Explanation Ensemble C	Base Model C$_{\mathrm{JSD}}$	Explanation Ensemble C$_{\mathrm{JSD}}$
BCW	0.12282	**0.4450 (262%)**	0.24682	**0.273065 (11%)**
Diabetes (Classification)	0.58550	**0.93697 (60%)**	0.51667	**0.543646 (5%)**
Diabetes (Regression)	0.52691	**0.600067 (13%)**	0.35389	**0.65568 (85%)**
Codon Usage (DNA)	0.34279	**0.5564 (62%)**	0.28114	**0.340558 (21%)**
Codon Usage (Kingdom)	0.22220	**0.29387 (32%)**	0.34702	**0.39391 (14%)**
MIMIC-IV	0.02433	**0.10111 (315%)**	0.15518	**0.17912 (15%)**

Table 5.4: Explanation Consistency (C) and JSD Explanation Consistency (C_{JSD}) for the baseline models and explanation ensembles across all tasks tested. The percentage increase from baseline C (C_{JSD}) to explanation ensemble C is shown in brackets.

ensembles is lower than that of the baselines across all datasets, and that the separability is also spread across a lower range of values than both the baseline models and baseline ensemble models. These figures confirm that the discriminator portion of the explanation ensemble architecture is successfully encouraging each ensemble sub-model to learn similar features, and that this is in turn successfully forces models with different training hyperparameters to learn similar features.

As also reported in Table 5.4, these explanation consistency results are verified by also calculating the JSD consistency, C_{JSD} (Equation (5.2)), for each dataset. These results conclusively confirm the results of the original consistency measure C, with the baseline models having low C_{JSD} and explanation ensembles having higher C_{JSD} values. Figure 5.2b showcases these difference in JSD values across the baseline and ensemble models - the similarity to Figure 5.2c further confirms the results.

		BCW	**Diabetes (Class.)**	**Diabetes (Regr.)**	**CU (DNA)**	**CU (Kingdom)**	**MIMIC-IV**
Baseline Models	**Checkpoint Averaging**	0.2117	0.75322	0.53356	0.06585	0.06378	0.1527
Normal Ensemble Models	**Checkpoint Averaging**	0.2497	0.2790	0.5604	0.0007	0.2264	n/a
	Random Submodel	0.1392	0.3062	0.5075	0.0440	0.01722	n/a
	Submodel Averaging	0.1952	0.4597	0.4713	0.3882	0.1738	n/a
	CA-SA	0.2906	0.551	0.5193	0.5654	0.1638	n/a
Explanation Ensemble Models	**Checkpoint Averaging**	0.2485	0.0175	0.5322	0.2510	0.2695	0.2954
	Random Submodel	0.1365	0.2641	0.0625	0.2939	0.0193	0.01333
	Submodel Averaging	0.2983	0.0355	0.8389	0.0953	0.0330	0.2080
	CA-SA	0.3964	0.89222	0.8529	0.6462	0.3481	0.1784

Table 5.5: Explanation consistency, C, of checkpoint averaging, submodel averaging and random submodel picking on baseline models and both normal and explanation ensembles. CA-SA is checkpoint averaging followed by ensemble submodel averaging, CU the Codon Usage dataset, class. is classification and regr. regression.

Table 5.5 reports the results of checkpoint averaging, submodel averaging, random submodel picking and checkpoint averaging followed by submodel averaging. The results are consistent across all architectures: neither checkpoint averaging, submodel averaging nor random submodel picking improves explanation consistency. When compared to the results of the baseline techniques in Table 5.4, explanation consistency decreases when these extra steps are added, confirming that the proposed method produces the best results. In the case of submodel and checkpoint averaging, I hypothesise that this is the result of the averaged model using noisy features from all of the models used in the averaging process, whereas the deep explanation ensemble technique is designed to instead encourage *all* models to learn to use similar features *before* the averaging takes place. Conversely, the explanation ensemble technique is not powerful enough to force each of the submodels to learn *exactly* the same set of features, with this explaining why using one of the trained explanation ensemble submodels (at random) doesn't work as well; the averaging out of the (smaller than normal) set of noisy features across each of the submodels in the explanation ensemble plays a large part in the generation of consistent explanations.

This hypothesis is further verified by analysing the results of the checkpoint-averaging-followed-by-submodel-averaging (CA-SA) experiments reported in Table 5.5. By analysing the results in the normal ensemble models one sees that this combination of techniques increases the explanation consistency of the models, implying that averaging at both stages of the model is required. The results of the same experiment on explanation ensembles back this up, with the proposed architecture improving again upon the results of the normal ensemble CA-SA experiments. Thus, the benefits of explanation ensembles followed by CA-SA are two-fold: **1)** it improves explanation consistency even further, and **2)** it results in a much smaller model that can be run at inference time, significantly reducing computational costs whilst adding very little to the (one-time) training cost.

5.2.5 Explanation Ensemble Size Results

Research suggests that larger ensembles result in improved performance [25]. Experiments find this also transfers to explanation ensembles with Figure 5.5 showing how, in general, explanation consistency increases as the number of sub-models increases. Intuitively, this is to be expected - the more sub-models present in an ensemble, the wider the range of parameters available for the ensemble to "average out" over.

It is important to note, however, that as the number of sub-models increase, the practicality of the model decreases due to the computational and memory requirements needed to train the model. This is particularly important to consider when the sub-model architectures themselves are also large - for example, it is difficult to train explanation ensembles of size ≥ 10 on the MIMIC-IV mortality prediction task due to the memory required by the resultant ensemble network. However, as can be seen in Figure 5.5, there does become a point across all datasets where the explanation consistency begin to plateau.

It is interesting to consider why the point at which the consistency beings to plateau differs across datasets (and even different tasks with the same dataset). I hypothesise this is due to the number of features that are causally related to target versus how many features are spuriously correlated with the target. Explanation ensembles are designed such that the spurious correlations will be "averaged out" as the sub-models gradually learn to utilise only features present across all sub-models, and so in the ideal scenario the whole set of spurious features is covered by (at least) one of the explanation ensemble sub-models. Considering this hypothetical ideal scenario, it is clear that datasets with a smaller set of spurious features will require a smaller set of sub-models to achieve the best consistency by an explanation ensemble architecture possible. This hypothesis also extends to different tasks within the same dataset - each task will have a different subset of the dataset's features, one of which will be smaller than the other.

5.2.6 Explanation Quality Metrics

Tables A.12 and A.13 report the explanation infidelity and sensitivity max on each individual baseline and explanation ensemble model tested. Across all datasets, each model has low explanation infidelity and sensitivity max - this confirms that SHAP is producing explanations that are faithful to the models. As the reported infidelity measure is the mean infidelity across the whole dataset, this also shows that the explanation methods provide global fidelity.

As Table A.13 shows, explanations generated from explanation ensembles are also high quality; the range and spread of the values is the same as the baseline models, implying that the new architecture does not affect the quality of the produced explanations. Importantly, this confirms that the explanations are also faithful to explanation ensembles, meaning that the improved explanation consistency is due to the changes in the architecture (i.e. the SHAP discriminator) rather than inconsistencies present in the explanation generation method (i.e. SHAP, in this case).

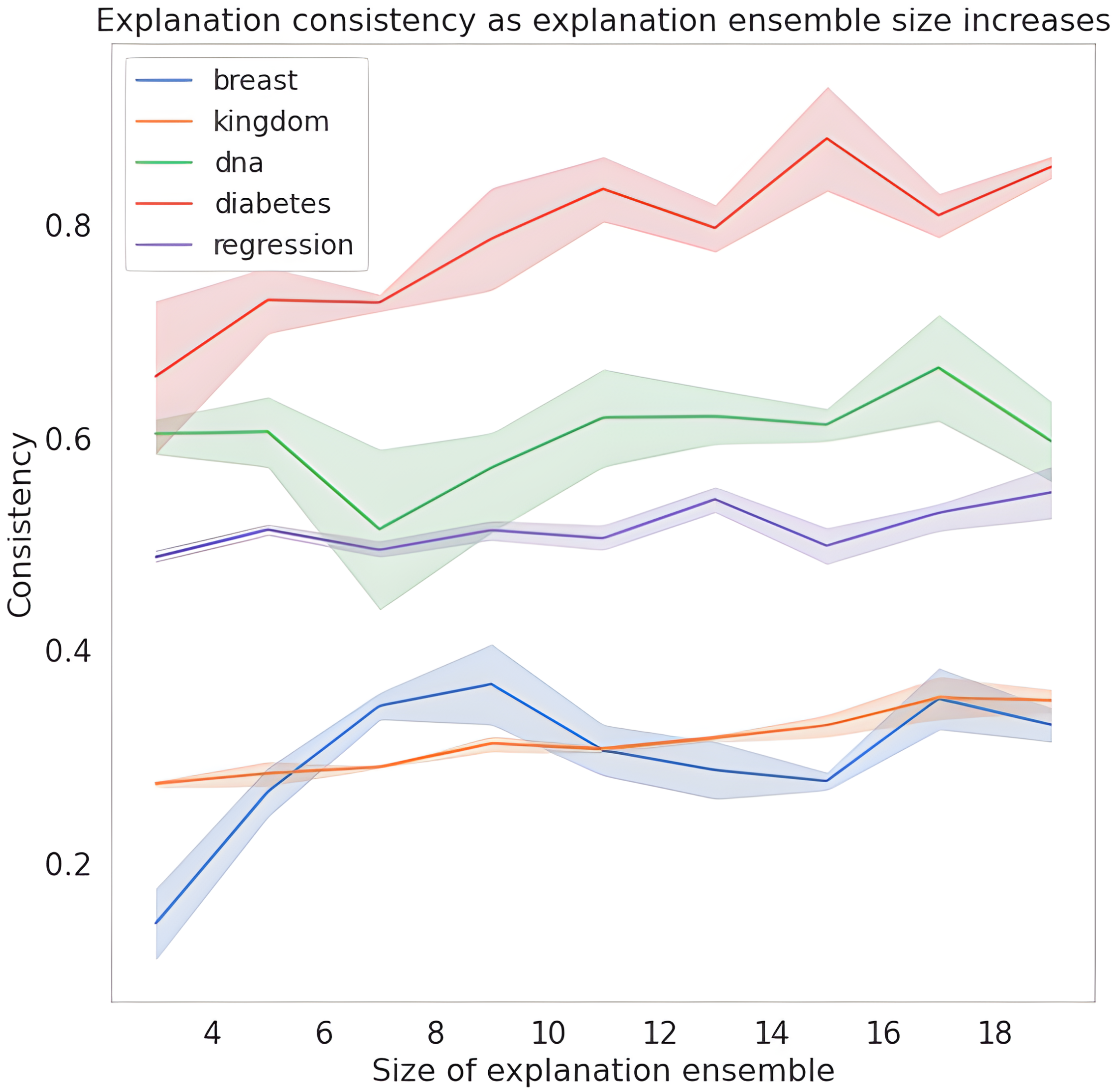


Figure 5.5: Explanation consistency (± standard error) of explanation ensembles as the number of sub-models within an ensemble increases across all datasets.

5.2.7 Statistical Significance Results

Both the test statistic U and the p-value are reported, for both the performance metric and explanation separability comparisons between the baseline and explanation ensemble models. Figures 5.2c and 5.3 also show for which datasets we report statistically significant results. Table 5.6 reports the relevant values for each dataset.

Across all datasets, the results of the Mann-Whitney tests support the conclusion that the proposed explanation ensemble architecture results in significantly improved explanation consistency C; all of the hypothesis tests result in significant results, highlighting that there is a significant difference between the results. This, coupled with the visualisation of explanation separability and JSD in Figures 5.2b and 5.2c, provide strong evidence that the proposed Deep Explanation Ensemble technique significantly increases explanation consistency.

5.3 Discussion

It is clear from both the initial consistency results on the baseline models, and from the corresponding studies carried out in Chapter 4, that the inconsistency of explanations is an important issue that is present in across a range of deep learning models; I hypothesise that it is a direct result of the stochasticity of training. Recent reports from industry [183–185] underline the importance of having explainable ML in industry (especially in sectors such as healthcare and finance), and how the lack of good quality explanations and the "unpredictable" nature of ML (which is highlighted by the inconsistency of explanations) are seen as barriers to wider adoption.

In this chapter, I have presented both a model-agnostic post-processing technique that improves explanation consistency and an entirely new architecture that can be trained specifically to learn more consistently. Through thorough experimentation on tabular data, I have shown that both of these methods are able to produce significantly better explanations (in regards to their consistency) whilst still retaining high levels of model performance and explanation quality (as measured through other, non-consistency, quantities). Through the use of a wide range of tasks we have seen that the proposed methods are able to work across both binary and multi-class

	Model Performance		Explanation Consistency	
Dataset (Task)	U **Statistic**	p**-value**	U **Statistic**	p**-value**
BCW	75	0.00249292	774	0.009378
KAIMRC (Regression)	81	0.00040946	6475	1.634×10^{-6}
KAIMRC (Classification)	51	0.04988344	3066	6.382×10^{-13}
Codon Usage (DNA)	81	0.00039825	5606.5	3.855×10^{-13}
Codon Usage (Kingdom)	0	0.00018267	11205	1.179×10^{-12}
MIMIC-IV	72	0.10397974	1350	8.226×10^{-16}

Table 5.6: U test statistic and p-values as calculated for the differences between the model performance and explanation separability $S_{(a,b)}$ of the baseline and explanation ensemble models; a two-sided test was used.

classification, as well as regression, tasks and have exhibited the usefulness of our techniques in the healthcare sector by focusing on healthcare datasets.

Through experimentation with multiple different model weight averaging techniques, I have shown that checkpoint averaging followed by ensemble submodel averaging can improve explanation consistency. Through the application of this technique to my Deep Explanation Ensemble architecture, I show that the architecture can beat the explanation consistency of current state of the art techniques even further whilst also significantly reducing the cost of running the proposed network at inference time. The final result is a comprehensive step towards creating consistent, robust models that can be deployed in sensitive domains such as healthcare and finance.

Parallels between the DEE architecture and modern feature selection algorithms (particularly self-guided algorithms such as [186]) can be drawn - the submodels of a DEE should essentially be learning to use only features which are very strongly correlated with the target. However, unlike most feature selection methods, the proposed technique does so in a self-supervised, end-to-end manner and is easily applied to any data modality. Furthermore, the DEE architecture will never *completely* remove a feature from use (unlike feature selection algorithms, where after it has been applied, some features will be completely removed from the model). This allows DEEs to still use these features in the edge-case scenarios where they may still be useful for classification purposes.

DEEs still exhibit the previously discussed explainability-performance trade-off phenomena and, while the performance differential is small (in the region of a couple of percent across all experiments), it would be prudent in the future to attempt to further address this issue. A simple solution would be to increase the number and complexity of the DEE's submodels - however, this would quickly become computationally challenging. For future work, it would be interesting to further investigate how the hyperparameters β, S can be tuned for performance (whereas this study has focused on tuning them for explanation consistency).

In Chapter 6 I explore the efficacy of DEEs when applied to different data modalities, and apply them to different sensitive applications outside of the healthcare domain. I also expand on the work done in Chapter 3 and investigate another type of malicious attack (the Membership Inference Attack), evaluating the robustness of DEEs against this type of attack and explore what makes them so robust.

CHAPTER 6

Applications of Deep Explanation Ensembles

The previous chapter introduced the concept of Deep Explanation Ensembles (DEEs), a completely novel model architecture and training algorithm that utilises model explanations during training. Through extensive evaluation, it was shown that this technique greatly improves the explanation consistency of models trained on tabular datasets. In this Chapter I explore the efficacy of DEEs on vision datasets, comparing and contrasting the agreement between model explanations from DEEs and expert's eye-gaze data. I then explore how DEEs can be applied to Federated Learning scenarios to improve user's data privacy, and discuss how these results could affect the applicability of deep learning to real-world scenarios. Specifically, this chapter:

- Evaluates the efficacy of DEEs on different data modalities (not just tabular data)
- Investigate whether DEEs improve the overlap between features used by DL models and domain experts, and discuss whether this has an impact on explanation quality
- Explores how model explanations can be used for Membership Inference At-

tacks

- Evaluates DEEs susceptibility to Membership Inference Attacks
- Investigates the usefulness of DEEs in Federated Learning settings

6.1 Applications to Medical Imaging Data

Applications of Deep Learning (DL) to healthcare have been growing rapidly in a wide range of medical scenarios; ranging from critical care [187] and diabetes risk prediction [28] to the diagnosis of chest x-rays (CXRs) [14]. This is partly driven by the rising accuracy of such models, with some beginning to achieve performance on-par with (or even exceeding) that of medical professionals [150]. However, despite these developments we are yet to see a similar growth in the number of DL models being deployed into real-world medical scenarios [16]. This is down to numerous limiting factors; most notably, before such techniques can become established in the medical field, they must be ethical in their decision-making, trustworthy, transparent and explainable [188, 189].

It is in these areas that many DL models can perform poorly. In particular, many models fail to accurately capture the causal relationships between input features and the output classification and rely instead on task irrelevant features. For example, a wide-ranging study on the use of Machine Learning (ML) and DL techniques for COVID-19 prediction from chest x-rays (CXRs) [104] has shown that many models are making spurious correlations, leading to the models being unable to accurately generalise. Furthermore, we saw in Chapter 4 that changes to training hyperparameters can greatly affect the learned features and discussed how this damages the trust between clinicians and DL techniques as it highlights just how sensitive to small changes the models are, even when those changes are independent of the medical questions the model is trying to answer.

Thus, the gold-standard for any ML model is to be able to achieve high-levels of performance whilst learning the concrete causal relationships present in the data. Unfortunately, the presence of learned causal features is extremely difficult to verify due to a lack of useful data supporting the task. Following practices in pedagogy, expert's Eye Gaze Data (EGD) can be used as a proxy for causal relationships [6,190]. The release and initial analysis of the MIMIC-CXR-EGD dataset [1] showed that even current state-of-the-art CXR classification models fail to learn the same set of features as used by radiologists in their diagnoses.

This sections evaluates the Deep Explanation Ensemble architecture presented in Chapter 5 on medical imaging data. Using the MIMIC-CXR-EGD dataset, which to the best of my knowledge is the only large-scale image dataset with accompanying expert eye-gaze data, I compare the similarity between explanations computed from DL models and the EGD from radiologists. Experiments show that there is a significant increase in overlap (increasing from -0.4634 to 0.5410 when measured by Normalised Scanpath Saliency and improving from 9.1233 to 0.8398 when measured by Kullback-Leibler Divergence) between explanations from DEEs and the EGD than there is from any other model architecture tested; including current state-of-the-art methods specifically designed to combat this issue. This section also shows that DEEs produces more consistent explanations than previous models on medical imaging data, increasing explanation consistency (Equation (4.2)) from 0.1785 to 0.5333 with no cost to model performance nor the need for specialist's EGD at inference time.

6.1.1 CXRs and Eye Gaze Data

Previous work (Chapter 2.3) has used explainability techniques to investigate the robustness and adaptability of DL models [172], finding that even small changes to the training procedure can result in significant changes to the learned features. These results, coupled with many network's susceptibility to issues such as adversarial attacks [122] and shortcut learning [18], suggest that many modern DL architectures are not necessarily learning causal relationships in the data to achieve high performance and might be relying on spurious correlations. It can be extremely difficult to verify that the learned features are indeed causal - there are only a limited number of mostly toy datasets that include descriptions of their causal relationships [191].

In the absence of such data, recent work has used EGD of experts making decisions on a visual task as a proxy for concrete causal relationships [1]. Such data can be used to determine whether models are learning features that domain experts would use in their assessment of the data - this use case has groundings from real-world applications, with similar techniques being used pedagogically in fields such as radiology [192]. The MIMIC-CXR-EGD dataset (Chapter 2.1.5) is a subset of MIMIC-CXR [45], containing 1,083 CXR images from three classes (Pneumonia, Congestive Heart Failure and Normal). Accompanying the images are aligned EGD from a trained radiologist. Both raw eye gaze information and calculated fixation points are available for this EGD - we refer readers interested in the EGD collection process to [1]. Alongside the release of the dataset the authors also show that explanations from traditional classification models do not significantly overlap with the radiologist's EGD. They propose a multi-task UNet model which uses EGD at train-time to learn to both classify the CXR image and reproduce the ground-truth EGD in order to improve the similarity between model explanations and EGD. However, the results are not very convincing and the study lacked a verifiable method of comparing their model explanations and the EGD. Additionally, this technique requires the use of expert EGD during training which is costly and difficult to collect, especially in the medical domain. This section compares the DEE techniques proposed in the previous chapter against both the baseline models and the improved UNet architecture using static EGD heatmaps proposed in [1], resulting in significantly higher degree of similarity between model explanations and EGD across all tested metrics.

6.1.2 Method

The purpose behind this section is to evaluate the DEE architecture introduced in Chapter 5 against medical imaging data. Not only does this verify that the proposed technique is able to perform well on more complex imaging data (rather than just the tabular data originally used), but the inclusion of EGD in the MIMIC-CXR-EGD dataset also allows the quality of the produced explanations to be evaluated; one would hope that the higher-quality explanations produced by DEEs have more consistent overlap with an expert's EGD.

To summarise the DEE architecture from the previous chapter, the intuition behind it is to train a discriminator D which encourages each of the S sub-models in an ensemble to learn a similar set of features. As each of the sub-models is trained with a different hyperparameter setup, they will each learn a slightly different set of features. As training progresses, D will learn to use the noisy features of each sub-model to (correctly) classify which sub-model explanations originate from - and in turn, the sub-models will learn to use different features for its classification, in order to fool D. The final result is an ensemble model that has learned to "ignore" a wide range of spurious features, with each of the sub-models only using features which all m_i agree are important. As multiple models must agree that any given feature is important for it to be used, it is more likely that these are causally related with the target, and thus is more likely to be included in an expert's eye-gaze data.

6.1.3 Experimental Setup

All experiments are carried out on the MIMIC-CXR-EGD dataset [1]. The models are trained on the same 3-label classification task: given a CXR image, predict its diagnosis (Pneumonia, Congestive Heart Failure or Normal). Three architectures are trained to compare our explanation ensemble to: **1) baseline:** a standard UNet architecture trained with a learning rate (LR) of 0.003 with Adam optimiser, batch size 32, and pre-trained EfficientNet-b0 [193] as the encoder and bottleneck layers; **2) improved UNet:** the modified UNet architecture [1] using static heatmaps during training to both classify and reproduce the EGD given a CXR using identical hyperparameters; and **3) standard ensemble:** an ensemble architecture consisting of 10 UNet architectures identical to **2)**, trained with LR=0.003 using the Adam optimiser and batch size 4 [1]. A reduced batch was used due to memory constraints. Each experiment allows for the comparison of the DEE's results against a different standard of model: **1)** is a standard classification model and used as a baseline, **2)** is the SOTA for similarity between model explanations and EGD, and **3)** confirms that DEEs are not just a result of utilising an ensemble architecture (and instead are inherent to the architecture and training procedure). UNet was used throughout to allow for direct comparison with the current state of the art model on the MIMIC-

CXR-EGD dataset in [1].

Deep Explanation Ensembles are trained using standard UNet with a classification head as their sub-models. Batch sizes of 4 and a learning rate of 0.00001 using the Adam optimiser are used. We use a CNN for our discriminator, with two convolution layers. Max pooling (with kernel size and stride of 2) and ReLU activations are used after each convolution layer. In all experiments, $\beta = 0.2$ to ensure the two parts of the main loss function are of the same order of magnitude. 10 sub-models per Explanation Ensemble are trained (see Appendix B for results on different numbers of sub-models). The accuracy (across all three labels) for all models is reported as a performance metric.

In order to allow for direct comparison with [1], the explanations for all models are computed using Grad-CAM [90] on the final convolution layer, with images being sampled from the test set for inspection. The similarity of these explanations is compared to EGD heatmaps generated from the eye-gaze fixations, which gives scalar values of importance for each pixel based on the radiologist's eye gaze [1]. To measure similarity to the EGD heatmaps standard practice of comparing saliency maps [194] is followed; specifically, both the Kullback–Leibler Divergence (KLD) as a distribution-based metric, and the Normalised Saliency Scanpath (NSS) as a location-based metric are used. KLD is an information-theoretic measure of the difference between one probability distribution and another; importantly, note that it is a *divergence* metric, meaning smaller values indicate better similarity. NSS is designed to be used to compare saliency maps with a ground-truth, and is the normalised saliency at fixed locations. Note that metrics such as Intersection over Union (IoU) are not suited to comparing EGD and saliency heatmaps [194] as one must consider how much importance is placed on each pixel (by both the model and the expert), rather than treating explanations/EGD as binary heatmaps.

It is known that NSS is sensitive to false positives, however that is desirable here - I hypothesise that the (non-explanation ensemble) models are learning many noisy features which are not necessarily causally linked to output - it is desirable to penalise the models if this is indeed the case. Negative NSS values highlight negative correlation, with chance at 0 and positive values indicating positive correlation.

As higher Explanation Consistency (Equation (4.2)) is linked to explanations more robust to spurious correlations, one would expect the Deep Explanation Ensemble model to achieve higher explanation consistency than other models tested. For each architecture, 10 models are trained with different random seeds. The Grad-CAM explanations are generated on the test set for these 10 models, with these explanations also being used to calculate the explanation consistency C for each architecture. Following the methods introduced in Chapter 4.2.2, a binary logistic regression classifier is used to measure the separability of two sets of explanations.

Furthermore, the results on Grad-CAM are confirmed by repeating these experiments with SHAP. This confirms that the results are not limited to one explanation technique; if both explainability methods agree on the outcome, then it is reasonable to conclude with increased certainty that the model is indeed learning "better" (i.e. similar, causal) features.

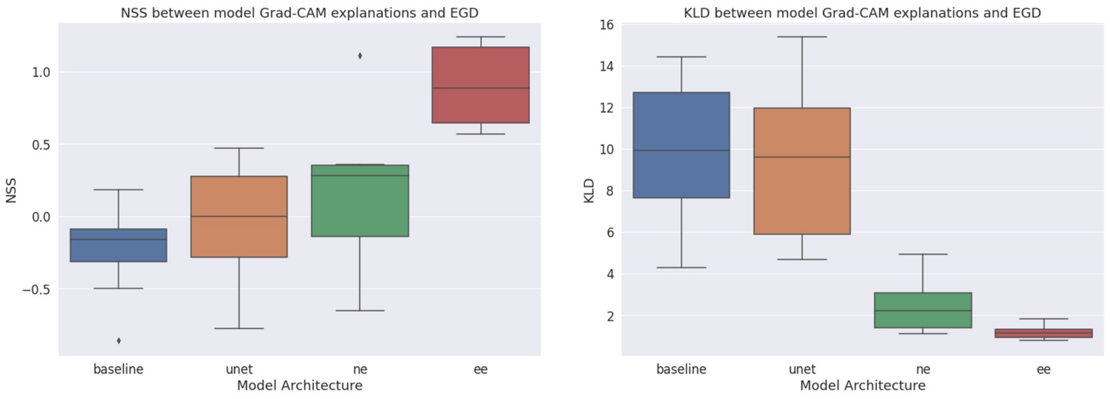
6.1.4 Results and Discussion

Table 6.1 reports the best model performance as well as summary statistics for both the KLD and NSS metrics used to compare the similarity between the model's Grad-CAM explanations and the EGD. Appendix B reports the results for each training hyperparameter setup used. The performance of both the Baseline and Improved UNet models are equal to the results reported in [1], confirming that these models are behaving as expected. Furthermore, both ensembling techniques perform better than these two models; this is to be expected given that they are ensemble architectures [195]. Importantly, the Deep Explanation Ensemble architecture is shown to improve upon the performance of the baseline models by 3.39% indicating that the models are not sacrificing model performance for improved explanations. Given that the explanations from Explanation Ensembles are shown to better align with radiologist EGD, this also suggests that features used by radiologists are better for disease classification than those learned by the baseline model.

Both Table 6.1 and Figure 6.1 report the Kullback-Leibler Divergence and Normalised Scanpath Saliency between the Grad-CAM explanations from each model architecture and the radiologist's EGD heatmaps. From Figure 6.1 one can see

Table 6.1: Table reporting the performance of the best-performing model for each architecture, alongside the similarity between the model Grad-CAM explanations and the EGD. Note that KLD is a divergence metric, and so smaller is better. Grad-CAM explanation consistency was calculated across all 10 training hyperparameter setups for each architecture.

		KLD		NSS		
Model	Accuracy	Mean (± std. dev)	Median (± IQR)	Mean (± std. dev)	Median (± IQR)	Consistency
Baseline [1]	75.55%	14.4041 ± 7.6886	13.4535 ± 10.5240	-0.8579 ± 1.2345	-1.0391 ± 1.4737	0.1785
Improved UNet [1]	76.51%	9.9371 ± 6.4179	9.1221 ± 8.4260	-0.3244 ± 1.5237	-0.4634 ± 1.9781	0.1596
Normal Ensemble	**79.86%**	3.8839 ± 3.2510	2.7740 ± 4.0799	-0.1646 ± 1.5721	-0.1307 ± 2.0840	0.3042
Explanation Ensemble (Ours)	78.94%	$\mathbf{0.8196 \pm 0.1273}$	$\mathbf{0.8398 \pm 0.1658}$	$\mathbf{0.6757 \pm 1.1178}$	$\mathbf{0.5410 \pm 1.5653}$	**0.5333**



(a) (b)

Figure 6.1: Boxplots of mean (a) NSS and (b) KLD between model Grad-CAM explanations and radiologist EGD, across each of the 10 training random seeds tested. Note that KLD is a divergence metric meaning smaller values are better.

that the Deep Explanation Ensemble model produces explanations that are more similar to the EGD than all other architectures tested, when measured by both a distribution-based measure (KLD) and a location-based metric (NSS). To confirm that these conclusions are statistically correct, a Paired t-test at the $\alpha = 0.05$ significance level is performed between the similarity metrics from the baseline and Explanation Ensemble models. The null and alternative hypotheses are the same for both KLD and NSS: $H_0 : \mu_d = 0, H_1 : \mu_d \neq 0$, where μ_d is the mean of the differences between the KLD/NSS values for the two architectures. The distributions of the differences were confirmed to be normal via simple plotting before carrying out the t-test. Table 6.2 reports both the test statistics and p-values for each of our hypothesis tests. Given that all p-values are significantly less than α, one can conclude that the Deep Explanation Ensemble architecture produces explanations that are statistically more similar to radiologist EGD than both baseline and current state-of-the-art techniques. Significantly, all models except Deep Explanation Ensembles achieve negative NSS scores, showing anti-correspondence against the EGD [194] and making the Deep Explanation Ensemble architecture the only method tested to use features that are positively correlated with those used by experts. This is further highlighted by the large reduction in KLD from our methods when compared with the baseline models tested; this underlines how significantly different the features used by current state-of-the-art models and medical experts are (and follows results suggesting that many networks suffer from shortcut learning [18] and spurious correlations [100]), and shows that the proposed method is a significant improvement. While experiments have focused on Deep Explanation Ensembles of size 10 in this chapter, the effect of changing the number of sub-models is explored in Figure 6.2. These experiments show that as the number of sub-models increase so does the agreement between model explanations and the EGD - however, it is important to note the trade-off between training cost and increased performance as the Deep Explanation Ensemble size increases.

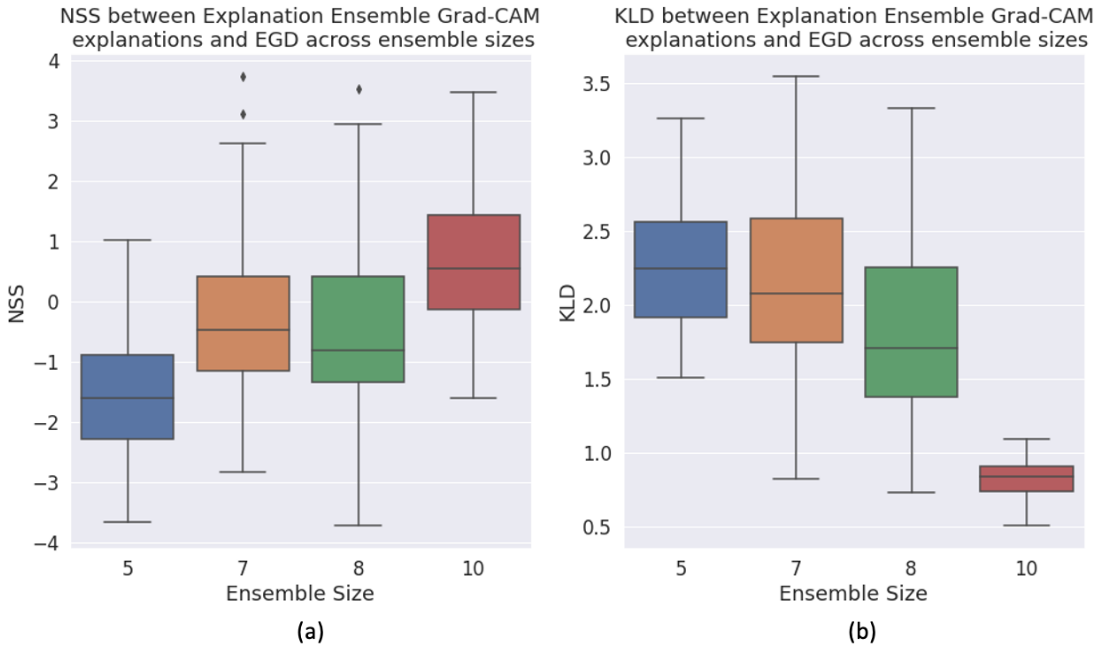


Figure 6.2: Boxplots of mean (a) NSS and (b) KLD between Grad-CAM explanations and radiologist EGD, across a range of ensemble sizes. For each ensemble size, 10 models with different random seeds were trained. Note that KLD is a divergence metric meaning smaller values are better.

In addition to improved similarity with expert EGD, explanation consistency (Table 6.1) is also significantly improved in Deep Explanation Ensemble models, verifying the results of the experiments in Chapter 5. This can also be seen by the significantly smaller range of NSS and KLD of the explanations from the explanation ensembles (as reported in Figure 6.1) when compared with other architectures tested. This inherently increases trust in the model, as it shows that our architecture is more robust than the others tested. It also further highlights how DEE networks learn "better" (i.e. similar to those in EGD) features than the baseline models - the models are learning fewer noisy/spurious features and instead placing more importance on the features that have a higher probability of being causally related to the task.

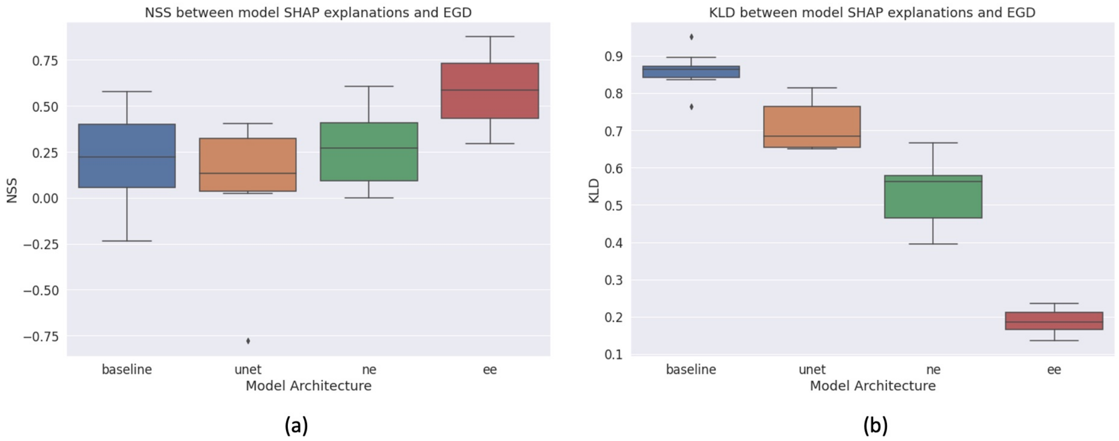


Figure 6.3: Boxplots showing the mean (a) NSS and (b) KLD between model SHAP explanations and radiologist EGD, across each of the 10 training random seeds tested. Note that KLD is a divergence metric meaning smaller values are better.

The similarity between SHAP values and the EGD data is also investigated; this is shown in Figure 6.3. Similarly to the Grad-CAM results, one sees that the proposed Deep Explanation Ensemble architecture improves the similarity upon all other model architectures tested. Similar patterns can be seen between all 4 architectures tested across the KLD and NSS values on the Grad-CAM and SHAP results, with the boxplots highlighting that the level of improvement of our explanation ensemble architecture is at the same scale regardless of the explainablility technique used. As both the results of Grad-CAM and SHAP agree, one can conclude that our proposed model is learning to use features similarly to a radiologist. These results can also be seen from a visual comparison of explanations: Figure 6.4

shows example CXRs and their corresponding EGD and explanations from all models tested, showing that Deep Explanation Ensembles places much more importance on regions similar to the expert radiologist (i.e. around the lungs and heart) than both the baseline and current state of the art models. Notice how columns 2 (baseline Grad-CAM) and 3 (Improved UNet Grad-CAM) in Figure 6.4 show how much of the feature attribution is placed in spuriously correlated features (such as the top-left corner and the image borders). On the other hand, the Deep Explanation Ensemble architecture learns a significantly different set of features (using features around the lungs and heart, with these areas much more closely matching the areas shown in the EGD heatmap in the first column), further showing that this training technique has a notable affect on the representations learned by the model. This is desirable, as it highlights how the proposed model is learning to use features similar to those used by experts, making it less likely that DEEs are over-reliant on spurious features.

Figure 6.5 shows how the learned features of our explanation ensemble model change as training progresses. Note that this figure shows only the most important pixels of each model - when showing the importance of all pixels, the heatmaps become difficult to analyse by eye. In particular, Figure 6.5 highlights how the DEE training process (i.e. the discriminator and the loss function in Equation (5.1)) encourages the sub-models of Deep Explanation Ensembles to learn similar features as training progresses, despite the sub-models starting with vastly different sets of explanations. This verifies that the intuitive understanding of our explanation ensemble architecture, and most importantly our understanding of *why* it produces explanations closer to expert's EGD, is correct.

Table 6.2: Test statistics t and p-values for the Paired t-test performed between the Explanation Ensembles and Baseline (top) and the Explanation Ensembles and Improved UNet (bottom) models.

	Test Statistic	***p*-value**
KLD	18.005	6.8698×10^{-34}
NSS	-9.9137	5.7567^{-17}
	Test Statistic	***p*-value**
KLD	14.4617	7.5950×10^{-27}
NSS	-5.8058	3.5764×10^{-8}

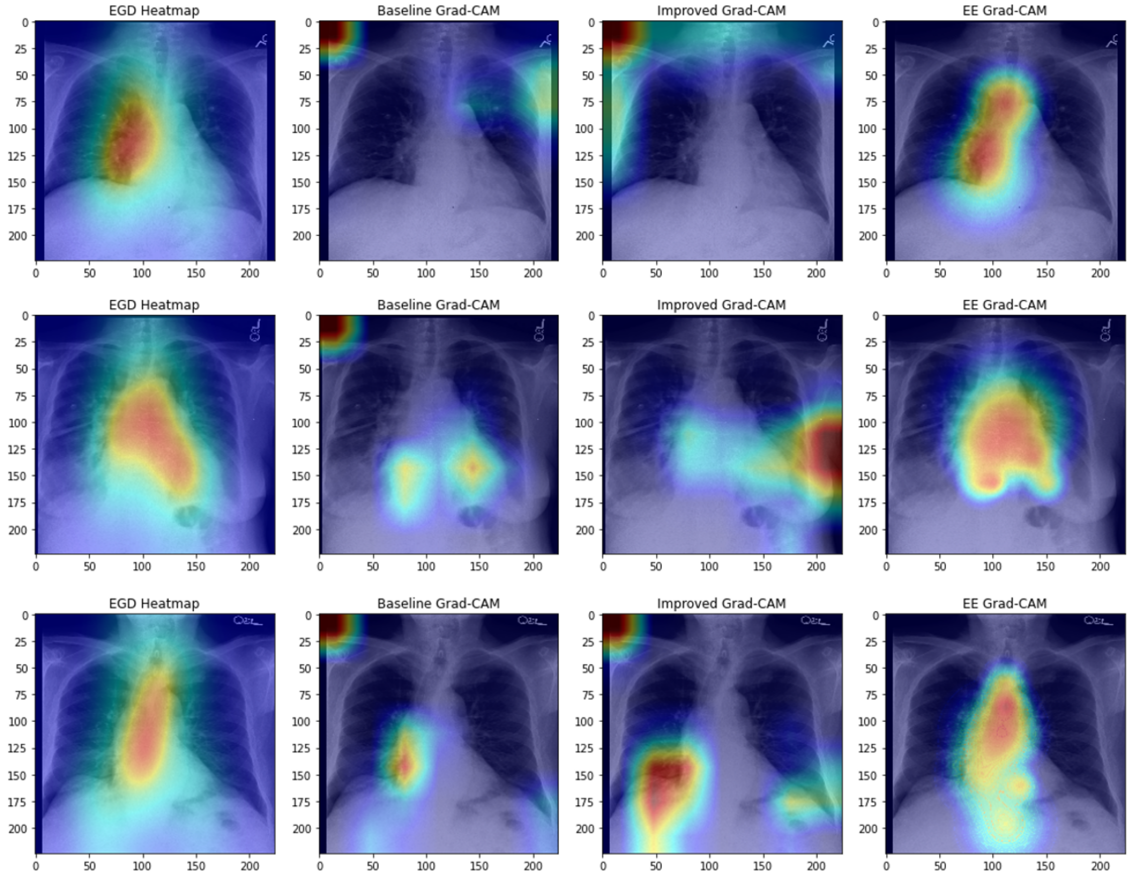


Figure 6.4: 3 samples from the MIMIC-CXR-EGD dataset, overlaid with the radiologist's EGD and Grad-CAM explanations from the baseline, improved UNet and Explanation Ensemble models.

6.1.5 Do DEEs Produce Better Quality Explanations?

Through the use of two explainability techniques and both distribution- and location-based metrics, we have shown that the Deep Explanation Ensemble technique originally proposed in Chapter 5 improves upon baseline models in both terms of performance and explanation similarity to EGD on the MIMIC-CXR-EGD dataset. Furthermore, this section has shown that the Deep Explanation Ensemble architecture also improves upon the current state-of-the-art models which share learned features with radiologist's EGD. In addition to improving agreement between model explanations and expert EGD, the proposed model architecture also improves classification performance and explanation consistency when compared with current state of the art techniques. Qualitative analysis of the results shows that our proposed architecture is a highly significant improvement upon current models, and whilst

I do not claim that these results are yet perfect they are a huge improvement in what is an extremely difficult task. Furthermore, unlike the previous state of the art [1] technique, the proposed architecture does not require EGD heatmaps during training - due to the cost of collecting EGD (especially in fields such as medicine, where expert knowledge is required), I believe this is a significant advantage over previously proposed methods.

In future work, it would be interesting to perform an in depth causal analysis of the learned features of the DEE model and compare this with a causal analysis of the learned features of baseline models, through the use of proper, theoretically-defined causal models. The improved performance, increased explanation consistency and better agreement with expert EGD suggests that DEEs may be learning more causal features than the baseline models, with the baseline models possibly relying more on spurious features. I hypothesise this as one would only expect causal features to be those that are learned consistently across multiple variations of a well-performing model. Furthermore, the increased agreement with expert radiologists (whom you would expect to use causal features in their diagnoses) further supports this conclusion. However, to fully verify this hypothesis, an extensive causal analysis of the trained models, and their learned features, must be undertaken (using techniques such as those used in [196] and [197]) and so I leave this for future work.

Due to its increased similarity with a medical professional's decision making process, I believe that more trust will be placed in our model by clinicians than current state-of-the-art techniques. I hope that these results encourage the use of our architecture in other areas of medical practice, and other sensitive fields, as well as the release of further datasets similar to MIMIC-CXR-EGD which can facilitate this type of research.

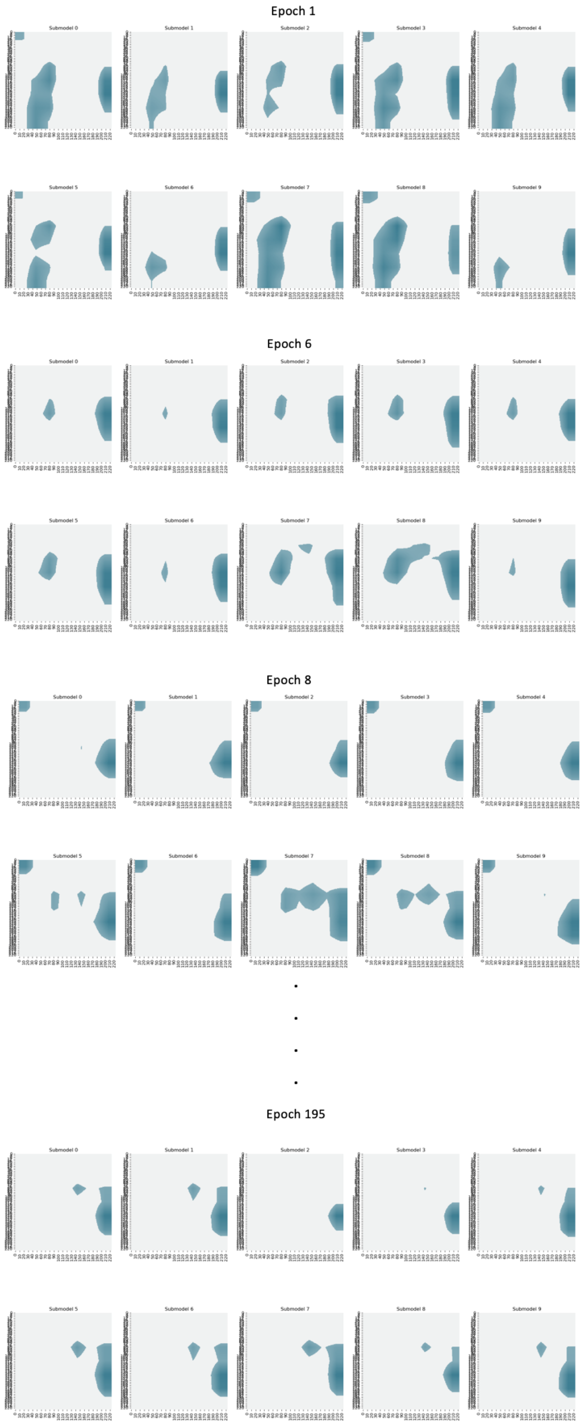


Figure 6.5: Average GradCAM values (across the validation split) of each sub-model of our Explanation Ensemble model, as training progresses. To aid with visualisation, only the most important 50% of pixels are shown. Sub-models start training with vastly different learned features, and as training progresses our training procedure encourages the sub-models to learn similar features. A fully animated version of this figure, and code to reproduce it on other models, is available at [2]

6.2 DEEs in Federated Learning Settings

Advances in modern machine learning are largely made possible by the availability of large volumes of suitable training data [198], with end-user's (often private) data being used by companies and individuals alike to create effective machine learning models [199]. As previously discussed in Chapter 2.4 this has raised concerns around the privacy of users whose data is included in these trained models, particularly where private data such as medical and financial records are concerned [200]. The European Union's General Data Protection Regulation (GDPR) [12] sets how such data may be used, and how it must be kept anonymised, private and secure. Although how exactly these regulations apply to specific ML applications is complex [77].

In an effort to combat this issue, Federated Learning (FL) was proposed as an alternative approach to the classical DL training setting (Chapter 2.4.4). In FL, a shared global model is trained through collaboration with a federation of private devices. Under this scenario, the training of the global model is controlled by a central server using the data present on the (usually large number of) private devices; crucially, this allows each device's private data to remain on their own device. FL has traditionally been applied to mobile applications (where each user's phone acts as the private devices), but it can also be used in settings such as healthcare, where it is also imperative that user data remains private [24, 77].

However, federated learning alone is not enough to alleviate privacy concerns, particularly in settings where extremely sensitive data is handled. Much like traditional machine learning methods [131, 201, 202] FL has been shown to be be susceptible to numerous types of attack [203], ranging from those that affect model performance to those that can identify data used during model training. The latter, named Membership Inference Attacks (MIA), is of particular interest from a privacy perspective. MIAs are effective against both traditional machine learning [53, 199] and federated learning settings [204], despite FL being specifically designed to keep user's data private.

Mitigating against such attacks is of importance to both the end user and ML model owner alike. Users are unlikely to want to use ML-based products if their personal data is put at risk, resulting in lack of adoption for the ML model owner (as well as the possible legal consequences of inadvertently releasing private user data). Differential Privacy (DP) when training deep learning models was used to alleviate these issues [143]. DP places limits on the influence a single data point can have on a model, and has shown to *theoretically* protect against MIAs at sufficient privacy levels [19, 145]. However, this theoretical work assumes that members and non-members are drawn independently, and from the same distribution; often, this assumption is not valid in a real-world setting [146, 205]. Indeed, in [53, 199, 206] membership inference attacks were effective against differentially-private ML models under more real-world settings.

Alongside data privacy, there are numerous other challenges facing ML models that are deployed to real-world applications. Perhaps most notably is the concept of model explainability: by their very nature, modern deep learning models are black-boxes and hard to explain, resulting in distrust from many end-users. For example, it is imperative that decisions made by ML models are interpretable by patients and their doctors for ML techniques to see wide-spread adoption [188, 189] in the healthcare domain. This has led to the development of many explainability and interpretability techniques that aim to "open up" black-box models by explaining which features of an input contributed most to the model's decision [78].

This chapter explores how explainability methods can be used as both attack and defence techniques in the membership inference domain. First, I introduce a novel membership inference attack named `ExplAttack` that utilises model explanations rather than model outputs and show that this is more effective than current state of the art MIAs in both federated and non-federated settings. I then demonstrate how the Deep Explanation Ensemble model architecture introduced in Chapter 5 can be used to mitigate against MIAs, showing that the efficacy (as measured by membership advantage) of both `ExplAttack` and existing MIAs is reduced to 0 when targeting trained Deep Explanation Ensembles in both traditional and federated settings. I also compare results against models trained with differential privacy, and highlight how the proposed DEE architecture is less susceptible to MIAs (when compared to differentially-private models in real-world scenarios) whilst keeping levels of performance at least equal to (and often greater than) both DP and non-DP models. The chapter ends by investigating why DEEs are robust to MIAs by comparing feature attributions from traditional models with those from DEEs.

6.2.1 Problem Background

As discussed in Chapter 2.4.3, it has been shown that deep learning models are susceptible to memorising training data, even if these models still achieve high levels of generalisations [130]. This memorisation results in models being prone to a number of different malicious attacks, including membership inference attacks [131,132]. These attacks are given an input (M, x) (where $M : \mathcal{X} \to \mathcal{Y}$ is a trained machine learning model and $x \in \mathcal{X}$) and attempt to infer whether x was part of the training set of M, $\mathcal{X}_{\text{train}}$. Such attacks can be used to infer information about the subject of x - for example, their relationship to the objective of the classifier M. Susceptibility to MIAs is regarded as an inherent privacy risk of machine learning models, with the US National Institute of Standards and Technology (NIST) specifically labelling successful membership inference attacks as a privacy violation [133].

Federated Learning (FL) is a training technique that is designed to improve user's data privacy (Chapter 2.4.4). Although FL allows the training of a ML model without clients explicitly sharing their data, it alone is not enough to provide sufficient privacy protection and instead must be used in conjunction with additional privacy-preserving methods [140]. For example, it has been shown that membership inference attacks are viable in the FL setting [141] as well as other attack methods that utilise unique properties of federated models [142].

One of the main defences against membership inference attacks (and many other privacy-related issues) in both traditional and federating settings is the use of Differential Privacy (DP) during training (Chapter 2.4.4). If a training algorithm satisfies DP then it is formally guaranteed that a model trained on dataset D and a model trained on dataset D', where D, D' differ by exactly one sample, will produce statistically similar results. This protects both content and output privacy of the model, though often comes at the cost of model performance (and compute cost).

This section's contributions are two-fold: **1)** it introduces a novel membership inference attack (that is named `ExplAttack`) that utilises explainability rather than model outputs that achieves state of the art attack performance, and **2)** shows that, through the utilisation of model explanations during training, it is possible to train Deep Explanation Ensemble models that are robust to both the novel and existing membership inference attacks presented in this section.

6.2.2 Explainability-based Membership Inference

In Chapter 4, we saw that due to the inconsistency of model explanations, one can easily distinguish between two models based on their produced explanations (even when these models have identical architectures, and differ only by a slight change in training hyperparameters). A natural next question to ask is: can explanations be used to infer membership inference? For such an attack to be effective one must first choose a suitable explainability technique, the choice of which will determine whether the attack is white- or black-box. For example, while GradCAM [90] requires access to the model to calculate feature attributions (making it impossible to use in a black-box attack scenario, where full access to the model is not guaranteed), SHAP [21] or Feature Ablation [17] can be computed when just model outputs are available.

The proposed `ExplAttack` follows the same process as traditional black-box MIAs, but uses explanations instead of model outputs: it trains a secondary classifier (the *attack model*) on the set of explanations, with the aim of being able to classify which explanations were calculated from the *target model's* training set samples. In order to train this secondary model, some subset of the training data for the target model must be available; to achieve this, one can utilise the idea of shadow data, and shadow models, that have been proposed as existing MIAs to generate training data for our attack model [132]. It is important to note that, if no (or few) training set members are known, and it is for some reason infeasible to generate shadow models then it is also possible to train the attack model in an unsupervised manner. When there is not enough shadow data present, an anomaly detection [207] attack model is trained on a large set of samples; as only a small subset of this data will originate from the training set of the target model, the attack model should learn a boundary between training set members and non-members.

This section focuses solely on explainability techniques that can be utilised in the black-box attack setting. Techniques that could only be used in white-box settings, such as GradCAM, typically all utilise model gradient's in their explanation computations. Attacks that utilise gradients have already been shown to be extremely effective in both traditional and federated models [141] and our proposed white-box techniques would be similar (although not identical: note that the proposed attack models only require the final explanations as input whereas previous techniques requires the model output and complete set of gradients for input features and individual hidden layers). Through the utilisation of existing explainability methods, our proposed attack is unique in that it allows the use of something *similar to* model gradients without the need for white-box access to the target model.

6.2.3 Deep Explanation Ensembles: A Defence Against MIAs

Deep Explanation Ensembles can be adapted to a number of scenarios. Firstly, they are not restrained to classification tasks - by changing the first part of Equation 5.1, DEEs can be trained on any task (e.g. regression) by choosing a suitable replacement for $\text{CELoss}(\cdot, \cdot)$. Secondly, it is possible to train DEEs in a federated setting by replacing classic Stochastic Gradient Descent with the FedAvg [138] algorithm.

6.2.4 Experiments & Results

This section first explains the experimental setup used to evaluate both the proposed novel explainability-based MIA as well as the efficacy of Deep Explanation Ensembles (DEE) as a defence mechanism against MIAs. A summary of the results of these experiments is then presented; full tables of results can be found in Appendix C.

Experimental Setup

Similar experimental setups and datasets to previous studies on MIAs [53, 199] are used, with additional experiments designed to evaluate the proposed methods on different data modalities and the federated learning scenario. Through the use of similar toy datasets to previous studies it is possible to easily compare the efficacy of `ExplAttack` and the robustness of DEEs to these attacks with state-of-the-art MIAs and models. This analysis is then extended to larger, real-world computer vision datasets to evaluate the effectiveness of the proposed techniques in real-world applications.

Datasets. Baseline experiments are carried out on the MNIST [208], COMPAS [30], Adult [52] and Texas [209] datasets. The Texas data was used following the pre-processing described in [53]. These datasets were chosen as they allow for comparison with previous MIA studies [53, 199] and cover a range of data modalities and task/data complexities. To evaluate the proposed methods on real-world data, the MIMIC-CXR-EGD [1] dataset is used, which consists of 1,083 chest x-rays across 3 different diseases. Similarly to the previous studies on MIMIC-CXR-EGD in this thesis the methods in [53] are followed to split the datasets into members, non-members and shadow data, randomly sampling from the original datasets to garner 15000 members, 25000 members and using the remaining data as shadow data; COMPAS uses 2000 and 1000 respectively. Table 6.3 reports a summary of the experiment setups used.

Federated Learning. To test the applicability of the DEE architecture to FL setups, the FEMNIST [50] and INaturalist [51] datasets are used, as well as a Synthetic federated dataset [50]. The Synthetic dataset was generated according to [50] using the default distribution with 77 features. Federated Learning is achieved via the FedAvg protocol [138] using the recommended number of clients and samples [50, 51] for each dataset.

Model Architectures. Baseline models are all trained with the Adam optimiser with a learning rate of 0.0001 and batch size 64 for at most 50 epochs, with an early stopping procedure in place to address overfitting. Tasks on tabular datasets (Synthetic, COMPAS, Adult and Texas) use MLPs with an input layer followed by two hidden layers - dropout is applied after the first two layers. MNIST and FEMNSIT both use a CNN with two convolutional layers with kernel size 3, with max-pooling and fully connected layers in between. All models use the LogSoftmax activation function. For the MIMIC-CXR-EGD dataset, the current state-of-the-art

UNet architecture from [1] is used, and for INaturalist a fine-tuned Densenet-121 model (pretrained on ImageNet). Both the ensemble and Deep Explanation Ensembles use the same MLP/CNN architectures as their sub-models, with DEEs also including a small MLP of one hidden layer as their discriminator. Each ensemble consists of 10 sub-models. For DEEs, optimally $\alpha = 2, \beta = 0.1$ after performing a hyperparameter grid search; this prevents the discriminator of the DEE from overfitting and ensure the two parts of the loss function in Equation 5.1 are of the same magnitude. For each (dataset, model) pair, 5 model variations are trained (i.e. the architecture is the same but the random seed is changed).

	Dataset Properties				DP	Model Accuracy (± std. dev.)			
Dataset	**Num. Samples**	**Num. Features**	**Num. Classes**	**Federated**	ϵ	**Baseline**	**Ensemble**	**DP**	**DEE**
MNIST	60,000	784	10	✗	2	98.80 ± 0.0462	93.00 ± 1.584	91.68 ± 1.134	$\mathbf{98.89 \pm 0.176}$
FEMNIST	805,263	784	62	✓	2	84.68 ± 2.093	$\mathbf{85.91 \pm 0.731}$	82.91 ± 1.442	85.12 ± 1.548
Synthetic	734,463	72	12	✓	4	69.87 ± 1.832	$\mathbf{81.73 \pm 3.121}$	75.04 ± 4.265	77.52 ± 1.745
INaturalist	2.7M	150528	10	✓	4	83.97 ± 1.34	83.40 ± 1.18	70.03 ± 4.58	$\mathbf{84.14 \pm 1.09}$
COMPAS	7214	466	2	✗	2	89.91	91.00 ± 1.289	75.32 ± 2.503	$\mathbf{91.87 \pm 0.693}$
Adult	32,561	205	2	✗	3	74.90 ± 1.632	75.28 ± 2.372	75.73 ± 0.823	$\mathbf{76.57 \pm 0.933}$
Texas	348,700	252	100	✗	4	83.55 ± 0.796	72.15 ± 0.686	73.55 ± 0.796	$\mathbf{83.79 \pm 1.273}$
MIMIC-CXR-EGD	1,083	150528	3	✗	4	75.57 ± 2.43	$\mathbf{76.63 \pm 2.63}$	68.43 ± 0.62	76.03 ± 1.09

Table 6.3: Statistics, privacy level (for models trained with Differential Privacy (DP)) and average model performance (± standard deviation) per dataset across all models tested. For each (dataset, model) pair, 5 models were trained.

Differential Privacy. To evaluate attack performance on differentially-private models, versions of the baseline models are trained using the DP-SGD algorithm [143, 210]. To allow for better interoperability between different datasets, the implementation of DP used calculates privacy budget based on three different hyperparameters, rather than aiming for a given ϵ. Following standard practice [143, 210] to set these hyperparameters, $\delta = \frac{1}{N}$ (where N is the size of the dataset), the maximum grad norm is set to 1.2 and the noise multiplier to 4. The achieved level of differential privacy, ϵ, that these hyperparameters achieve is reported in Table 6.3.

Membership Inference Attacks. Each model variation is tested against many membership inference attacks. Standard black-box attacks using both shadow and non-shadow datasets are used as a baseline attack, as well as a threshold attack. The current state of the art MIA, that utilises shadow datasets [132], is also tested against. For the proposed `ExplAttack`, experiments include using both Logistic Regression (LR) and Multi Layer Perceptron (MLP) models trained on the feature attributions calculated via SHAP [21]. This section focuses solely on the black-box attack scenario, assuming that the attacker only has access to the target model's outputs; as such, the explainability technique used in the explainability-based MIA must not require direct access to the model. As we have already seen that SHAP is sensitive to small changes in models and hence is suited to this task in Chapter 4; however it is important to stress that any suitable explanation technique could be used. Membership Advantage, defined as `Adv` = `TPR` − `FPR`, where `TPR` is the True Positive Rate and `FPR` is the False Positive Rate [19], is used to measure the power of a Membership Inference Attack, as previous studies have shown attack model accuracy to be a poor indicator of performance [211].

Explainability-based MIA Results

Table 6.3 and Figure 6.6 both show the accuracy of all models trained, across all tasks. All trained models match the performance of models trained in the original studies [1,30,50–52,208,209] verifying that any susceptibility to MIAs is not due to improper training.

Attack Dataset	`ExplAttack` (MLP)	`ExplAttack` (LR)	Black-Box	Rule-Based	Shadow
MNIST	**0.30 ± 0.01**	0.27 ± 0.06	0.26	0.28 ± 0.01	0.18 ± 0.01
FEMNIST	**0.46 ± 0.12**	0.29 ± 0.11	0.20 ± 0.06	0.25 ± 0.19	0.19 ± 0.05
Synthetic	**1.0**	0.41 ± 0.13	0.30 ± 0.06	0.32 ± 0.02	0.27 ± 0.06
INaturalist	**0.37 ± 0.02**	0	0.29 ± 0.02	0.31 ± 0.02	0.31 ± 0.02
COMPAS	0.21 ± 0.05	0.15 ± 0.02	0.12 ± 0.07	**0.27 ± 0.06**	0.20 ± 0.08
Adult	0.17 ± 0.04	**0.44 ± 0.10**	0	0.13 ± 0.07	0.17 ± 0.08
Texas	**0.48 ± 0.08**	0.46 ± 0.20	0.25 ± 0.03	0.34 ± 0.06	0.21 ± 0.30
MIMIC-CXR-EGD	**0.52 ± 0.02**	0.50 ± 0.02	0.20 ± 0.06	0.25 ± 0.18	0.19 ± 0.04

Table 6.4: Membership Advantage (± standard deviation across 5 model variations) of each attack tested on the baseline models on all 8 datasets.

Figure 6.7 and Table 6.4 both show the membership advantage of all attacks tested on the baseline models trained across all datasets, including the proposed `ExplAttack` method with two different attack models. Figure 6.7 shows that explainability-based attacks are more effective than baseline and state of the art attacks, although interestingly it is not always the case that the MLP-based attack always outperforms our LR based-attack (or vice-versa). For example, by inspecting Table 6.4 one can see that the MLP-based `ExplAttack` on the Adult dataset performs slightly worse than the LR-based explainability attack. I hypothesise that this may be because it

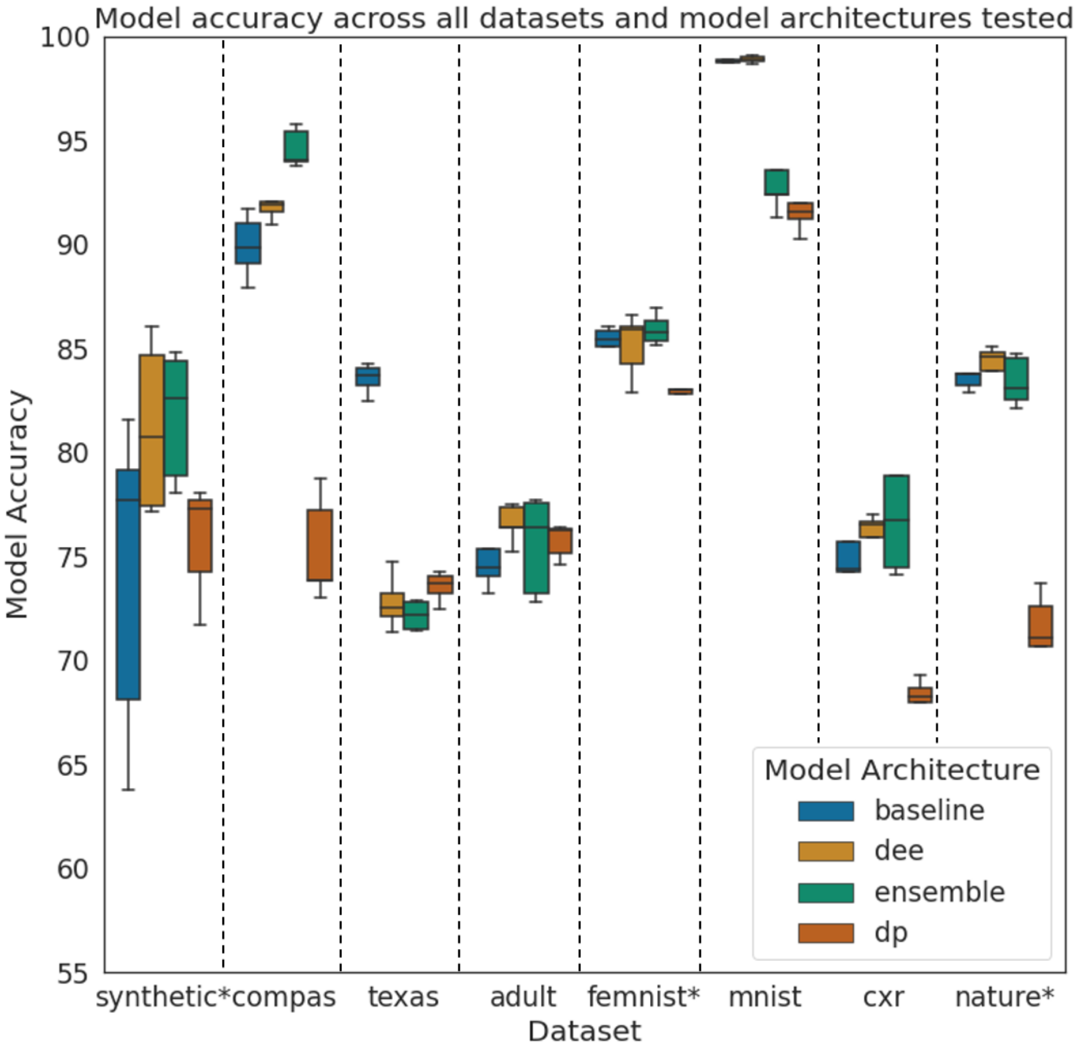


Figure 6.6: Boxplot of model accuracy on all datasets tested, across all model architectures. Datasets denoted with * are federated datasets and were trained with the `FedAvg` algorithm.

is more likely for the MLPs to begin to overfit than the LRs, even though actions have been take to attempt to prevent this from happening (e.g. dropout and early stopping as noted in Section 6.2.2). Similarly, the LR-based `ExplAttack` performs poorly on the MIMIC-CXR-EGD dataset; this is likely due to the large amount of features in a CXR image (a single image is 3x224x224), and the overall similarity of each image (CXRs are collected in highly standardised environments, resulting in a small amount of image diversity), which Logistic Regression is not very well suited to.

Overall, however, the membership advantage of the proposed `ExplAttack` is sig-

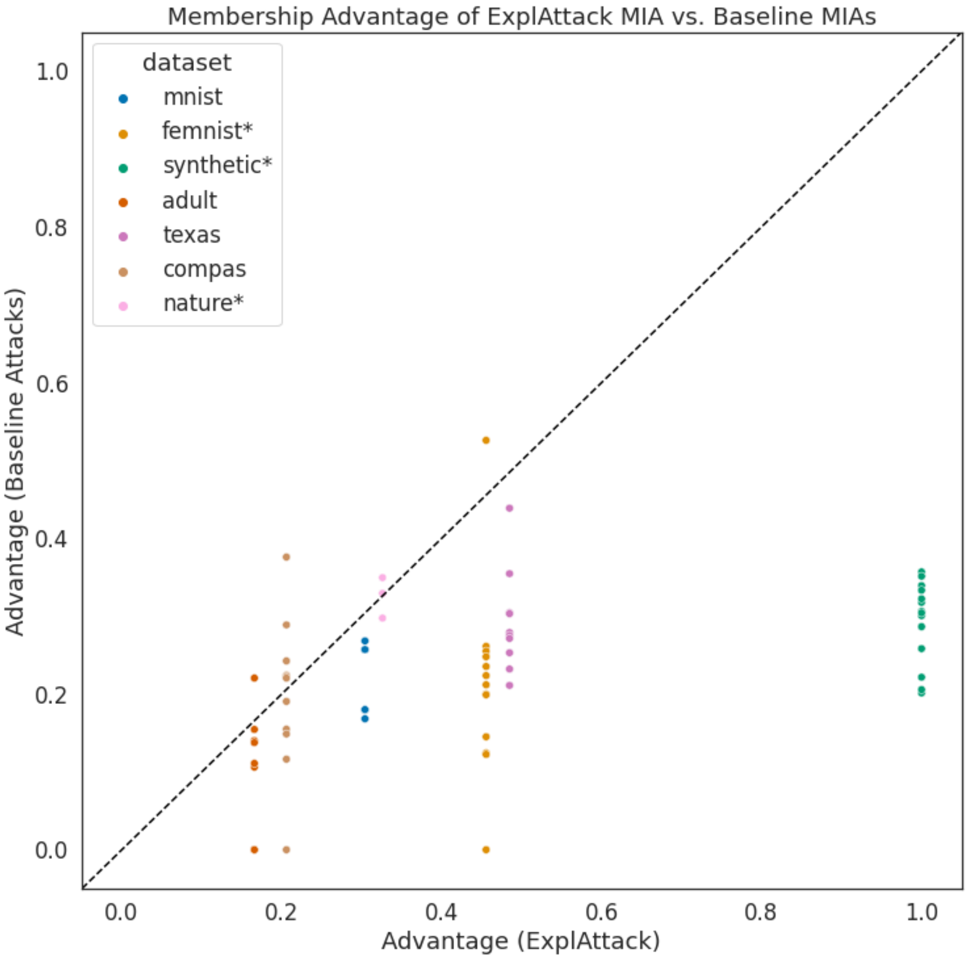


Figure 6.7: Scatterplot of membership advantage of all MIAs tested, including the proposed `ExplAttack` technique, tested on all datasets across multiple MLP models. Datasets denoted with a * are federated datasets, with the MLPs being trained with the `FedAvg` algorithm.

nificantly higher than the other state of the art attacks tested. Figure 6.8 shows that the proposed MIAs are also effective against differentially-private models trained via the DP-SGD algorithm. This highlights how model explanations can be extremely sensitive to changes in model settings, and that the information this can provide must be taken into account when analysing the privacy provided by algorithms.

Deep Explanation Ensemble Robustness

Figure 6.8, and Tables 6.4 and 6.5, highlight how DEEs provide perfect protection against all attack types and datasets tested, with all attacks being unable to accurately infer membership on DEEs. Importantly, not only are DEEs impervious to classical MIAs, but they are also resistant to our proposed explainability-based MIAs. This underlines the improvements that can be gained by improving the explanation consistency of models via the use of explanations during training; by improving the quality of explanations produced by a model, these results show that it is also possible to improve aspects of the model that are traditionally targeted by membership inference attacks.

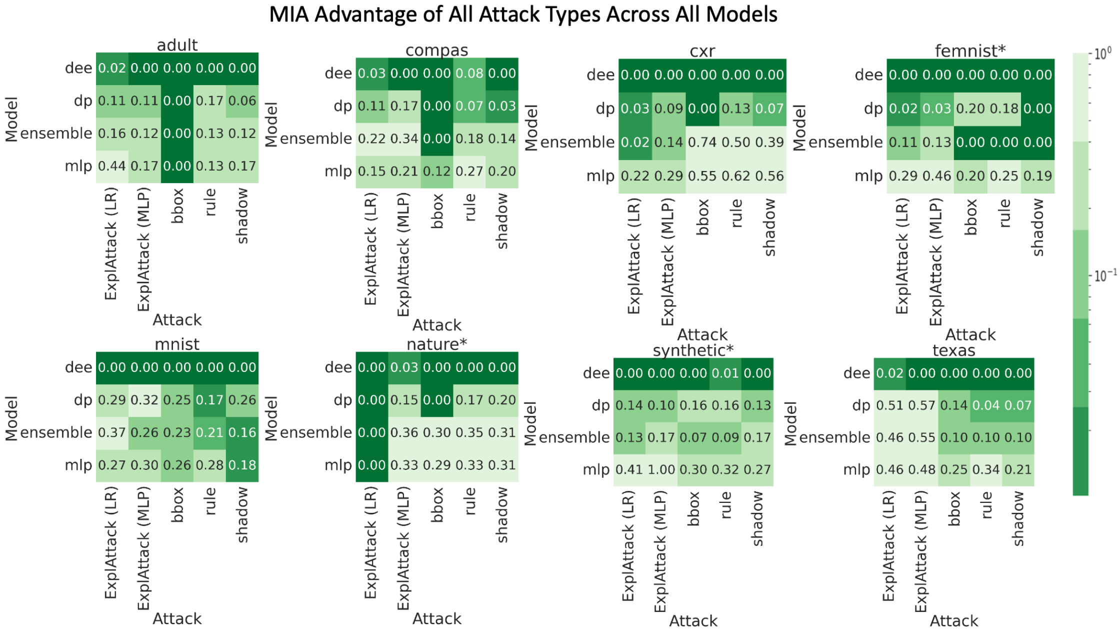


Figure 6.8: Heatmaps of the membership advantage of tested attacks on all model architectures tested across all datasets. Federated datasets are denoted by *. The values plotted are the average membership advantage across 5 trained models, rounded to 2 decimal places for the annotations. A logarithmic scale is used for visualisation purposes (although the actual membership advantage values are added as labels to each cell) due to the low spread of values.

Table 6.3 and Figure 6.6 show that the DEE architecture performs comparably to baseline models on all datasets, outperforming the baselines on most datasets. This is an indication that DEEs are learning to use a more insightful set of features, and are perhaps less reliant on noise, than other architectures - a repeat of

Attack Dataset	ExplAttack (MLP)	ExplAttack (LR)	Black-Box	Rule-Based	Shadow
MNIST	0	0	0	0	0
FEMNIST	0	0	0	0	0
Synthetic	0	0	0	0.01 ± 0.01	0
INaturalist	0	0.03 ± 0.01	0	0	0
COMPAS	0	0.03 ± 0.04	0	0.08 ± 0.02	0
Adult	0	0.02 ± 0.04	0	0	0
Texas	0	0.02 ± 0.03	0	0.01 ± 0.01	0
MIMIC-CXR-EGD	0	0	0	0	0

Table 6.5: Membership Advantage (± standard deviation across 5 model variations) of each attack tested on Deep Explanation Ensemble models on all 8 datasets.

the results of Chapter 5 and Chapter 6.1. Notably, DEEs vastly outperform differential privacy-based models, despite choosing DP hyperparameters such that the privacy/performance trade-off is not too severe. Combined with Figure 6.9, these results show that DEEs are able to provide better privacy than models trained with the DP-SGD algorithm whilst also avoiding the inherent privacy-performance trade-off that is present in differentially-private networks.

Federated Learning Results

Data privacy is a fundamental advantage of Federated Learning algorithms - a user's data can remain on their own device whilst still contributing to a global model. However, Figure 6.10 shows that these models are still susceptible to membership inference attacks which, in a lot of situations, may be considered a privacy violation. Furthermore, the proposed ExplAttack is still highly effective on federated models, highlighting the need for further defences against such MIAs in the federated setting.

Figure 6.8 demonstrates that DEEs address this issue in the federated setting. Similarly to the results on baseline datasets shown in Section 6.2.4, DEEs are resistant to both ExplAttack and traditional MIAs, achieving near-perfect membership advantage scores across all datasets. Importantly, this also holds for the real-world, complex INaturalist dataset, which shows that DEEs are also practical in real-world scenarios and not just on toy datasets such as FEMNIST and the Synthetic dataset.

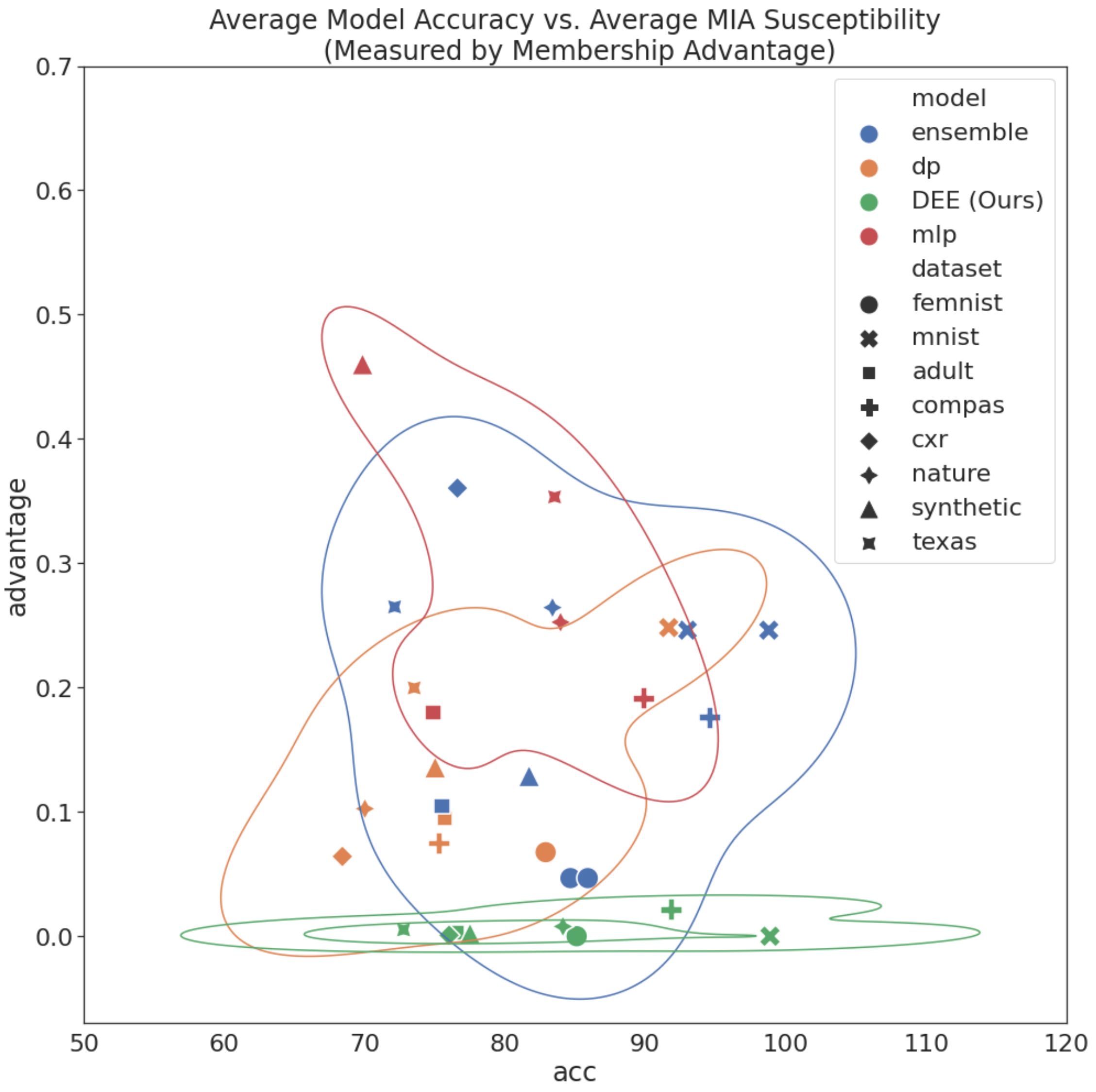


Figure 6.9: Model accuracy plot (with KDE) against membership inference on all (model, dataset) pairs tested. Note that models trained on the FEMNIST, Synthetic and Nature datasets were trained with the FedAvg algorithm.

6.2.5 Discussion

These results conclusively show that DEEs are extremely robust to membership inference attacks, under both regular (results on MNIST, COMPAS, Adult, Texas and MIMCI-CXR-EGD) and federated training regimes (results on FEMNIST, Synthetic and INaturalist). I hypothesise that this is due to Deep Explanation Ensembles learning to use a "better" set of features than other techniques, with much of the noise learned by the separate sub-models being averaged out - this leaves the final model placing the most importance on only features which are related to the

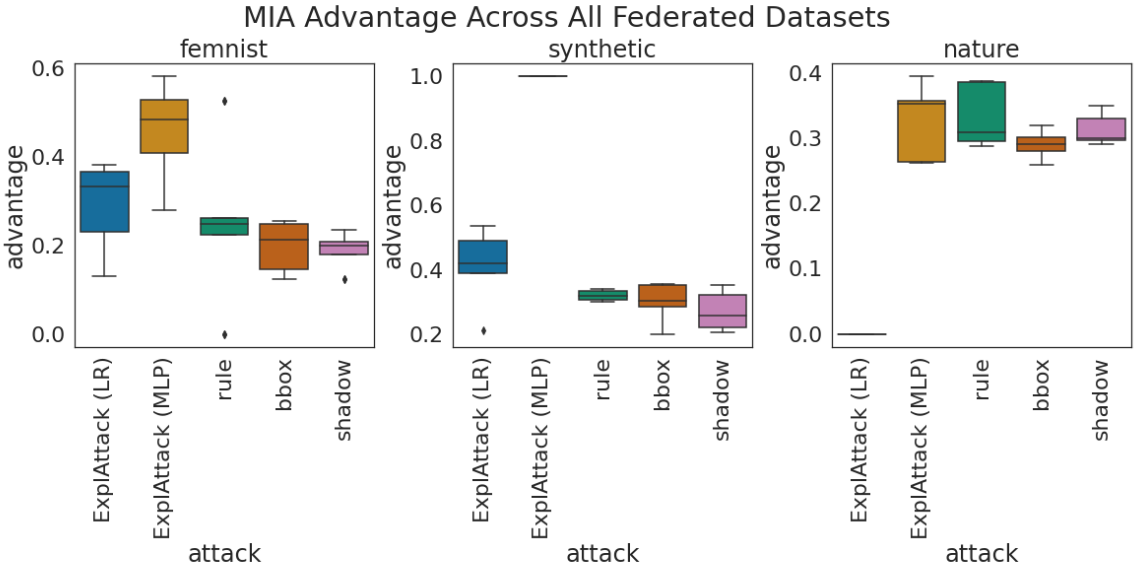


Figure 6.10: Boxplots of membership advantage of all MIAs tested, including our proposed `ExplAttack` technique, as tested on federated datasets.

task. This can be seen by inspecting Figure 6.11 - the SHAP values from Deep Explanation Ensembles trained on MNIST show much less emphasis on pixels that are not related to the number in the image, and instead more importance is placed on pixels in the numbers themselves. Hence, DEEs are less reliant on these noisy, spurious features and hence results in higher performance on the downstream task.

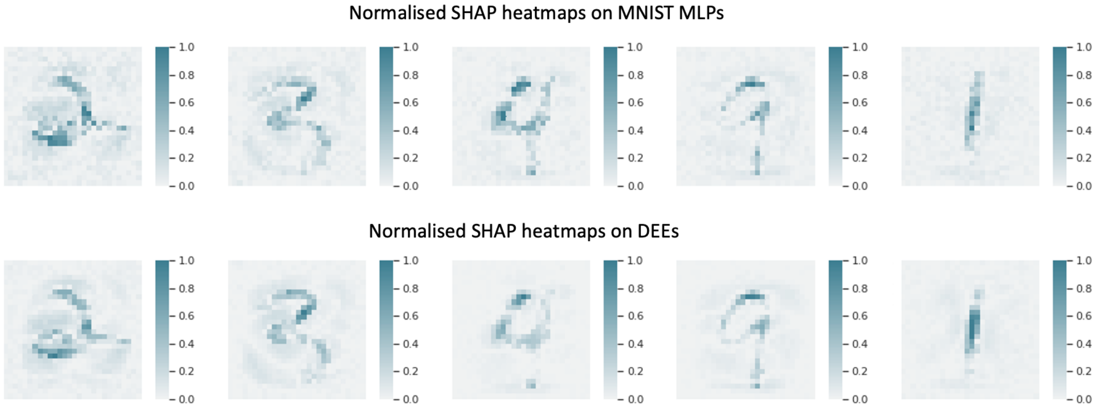


Figure 6.11: Normalised SHAP values on the first 5 MNIST samples from MLPs (top) and DEEs (bottom).

By inspecting SHAP values from `ExplAttack` attack model trained on MNIST it is possible to find the features the attack places the most importance on (i.e. the features differ the most between members and non-members). Figure 6.12 highlights

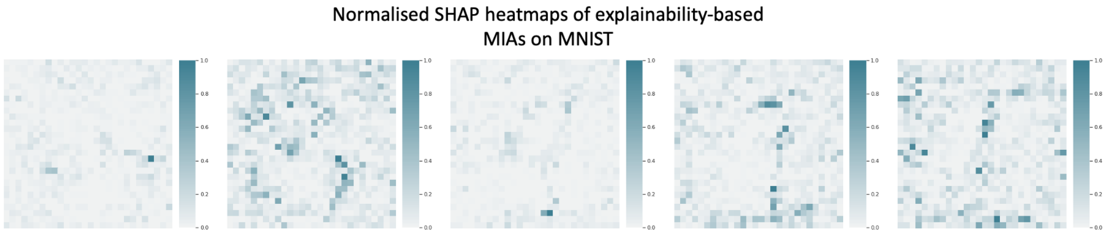

Normalised SHAP heatmaps of explainability-based MIAs on MNIST

Figure 6.12: Normalised SHAP values from the explainability-based membership inference attack on a baseline MLP, on 5 random MNIST samples.

how the feature importance of the MIA is spread across the entire feature space, suggesting that the MIA is successful as it is able to use differences in the noisy features learned by the target model to discern between members and non-members. As denoted in Figure 6.11, DEEs are less reliant on these features and so this explains why they are so much more resilient against MIAs; noisy, spurious correlations are no longer used by the target models and hence membership inference attacks can no longer use them to aid classification.

6.2.6 Conclusion

This chapter has shown that model explanations can be both used to more accurately infer train set membership than existing MIAs, and that they can be utilised during training to create more robust models. Through inspection of the learned features of Deep Explanation Ensembles on the MNIST dataset I have confirmed that they successfully reduce the number of spurious and noisy features used during classification, and we hypothesise that it is this that increases the robustness of Deep Explanation Ensembles whilst keeping high levels of accuracy (in contrast to other techniques such as Differential Privacy). Through a similar inspection of the features used by explainability-based MIAs I have confirmed that it is this through this improved learning of features that results in the reduced susceptibility to membership inference attacks. In future work it would be interesting to inspect this further, perhaps from a causal perspective, by taking a deeper look at how DEE's learned features differ from traditional architectures and how this affects the model's robustness; this would be a large study, and is outside the scope of this focused work.

CHAPTER 7

Discussion & Conclusion

Each chapter of this thesis has seen the introduction of a novel technique designed to address one of the three main problems facing Deep Learning models deployed in practice that were identified in Chapter 2. Here I discuss these methods from as a whole, examining how they can be used together rather than as individual methods, not just to improve DL applications in sensitive settings such as healthcare, but to further the field as a whole. In order to promote their use in actual applications, this chapter then gives concrete suggestions as to how, where and why this methods should be applied in practice. It then discusses some of the limitations of the work, proposes some new open questions that are direct results of the work in this thesis, and suggests numerous avenues for future work based on this.

7.1 Model Explainability: A Holistic View

As originally identified in Chapter 1.1, whilst modern machine learning techniques have repeatedly proven themselves to perform outstandingly well on a wide range of tasks, there are still numerous barriers that must be overcome to be able to successfully apply these techniques to real-world settings. As seen throughout this thesis, these issues are especially prevalent in high-risk domains such as healthcare and finance, where the consequences of an incorrect decision can be severe. Many current approaches to address these issues have been disjoint: indeed, the techniques reviewed in Chapter 2 all aim to tackle one specific weakness such as model transparency or privacy. However, I argue that a more holistic approach needs to be taken; much like how one cannot expect to fix your health via diet alone (and instead also considering fun things such as exercise and sleep), you should not expect to improve deep learning models by solving one problem at a time. This can also be seen intuitively - for example, it would be a reasonable assumption to think that a model that is less susceptible to spurious correlations is also likely less prone to bias, and more likely to produce quality explanations.

This thesis has focused on showing this through the lens of deep learning explainability. Although the all of the techniques proposed in this work are focused on improving and utilising model explanations, their overall, overarching aim is to also improve model privacy, transparency and robustness such that deep learning models can begin to be used in high-risk settings. This is initially demonstrated in Chapter 3, where the proposed adversarial sample detection technique shows that model explanations have uses beyond "opening up the black-box" and instead can also be used to improve model robustness. Perhaps most interestingly, Chapter 3 also offers a first glimpse into the utility model explanations as a tool for understanding the inner workings of a model beyond per-instance feature importance. Whereas previous studies simply utilise explainability techniques to understand which features of an input are most important - the model detects a cat in an image via its whiskers and ears, for example - the fact that these same explanations can be used to detect adversarial samples hints that they are perhaps picking up even more details on the inner workings of a model than is clear upon a first glance.

This idea is more fully explored in Chapter 4. These results conclusively show that explanations are inherently linked to a model's quality and robustness, and due to this it is possible to use them to uncover deep-rooted problems with modern deep learning architectures. The inconsistency of model explanations shown in Chapter 4.4 is a significant barrier, particularly to those who are already unsure of deep learning. The fact that *all* model architectures tested exhibited the same inconsistency problem - even those which are designed to be more robust, such as hyperensembles - highlights how important it is to take this holistic approach to model design. I argue that these results show it is no longer possible to design issue-specific models (such as those designed to improve, say, model robustness), as they inevitably succumb to one of the other problems such as a lack of transparency or susceptibility to bias.

Chapter 5 does this by looking at the bigger picture and developing techniques which are more consistent, and hence also more robust, private and transparent. By utilising the knowledge gleaned from the previous chapters that, by design, explanations are tightly linked to the inner workings of a model, the proposed DEE architecture is shown to be able to learn a much better set of features than classical models across all tasks tested; this is shown by the increased accuracy and explanation consistency. Indeed, one can ascertain that by focusing on improving the quality of the explanations you also improve the quality of the learned features and, hence, the model.

To properly consider how this affects the whole model, Chapter 6 follows this principle of looking at a model as a whole rather than its constituent parts and features by investigating how DEEs can improve a model's privacy and explanation quality. By extending the analysis of Chapter 5 to different data modalities, Chapter 6 definitively shows that DEEs improve the whole model. The comparison of the explanations from DEEs to that from domain experts highlights how, by simply focusing on learning more consistent features, models can more closely mimic decisions made by experts - something that I argue is imperative for a model to be used and trusted by domain experts in high risk settings, and yet is something that has largely been ignored up until quite recently. Furthermore, Chapter 6.2 supports this idea that, by using explanations to inspect and improve the features learned by a model, it is possible to improve a wide range of model qualities, including the level of data privacy provided.

The original aim of this thesis was to improve explainability techniques to produce better and more consistent explanations for healthcare-based machine learning models. However, through the study of this problem a much more important hypothesis was discovered, explored and eventually confirmed: it's not necessarily the explainability techniques that are wrong, its the models that are learning the wrong features. Through the idea of using model explanations to investigate the *whole* model, rather than how it behaves on specific instances, this thesis has gone on to suggest novel ways of improving model training, and how these techniques improve multiple facets of deep neural networks. I hope that this underlines the importance of taking this holistic approach to model development, wherein one looks at improving model training as a whole rather than attempting to solve one issue at a time.

7.2 DEEs: Recommendations for Applications

Throughout this thesis, the experimental focus has been on highly sensitive applications such as healthcare, biology and finance in which errors produced by any machine learning technique could have severe consequences. Such domains are of particular interest to machine learning practitioners as they are yet to see large-scale adoption of ML techniques despite high levels of model performance; this is likely due to concerns around how trustworthy these models are, as well as a general lack of understanding of how they works (see Chapter 2.1.8), and is why the majority of experiments throughout this thesis have focused on datasets and tasks from one of these domains. However, whilst earlier chapters have given theoretical suggestions for improving the training of neural networks, little thought has thus far been given to how and where these could be applied in practice. A general lack of such guidance is also another reason why many modern DL models do not get used in production, and so in this section aims to provide some ideas of when, where and how the techniques presented in this thesis should be used.

The explainability-based adversarial attack detection technique presented in Chapter 3 can be used as a post-processing method in any setting where the risk of malicious inputs to the model is high; for example, in finance or insurance settings. While the results of Chapter 6.2 suggest that DEEs may overcome the problem of adversarial attacks altogether, rendering this detection technique unnecessary, practitioners may wish to use this detection method in applications where computational power is at a premium and as such the training of a DEE is impractical. Additionally, while Chapter 6.2 does suggest that DEEs are robust to known attacks, it may still be possible for an adversary to develop an attack that is at least somewhat successful against them. As such, it may still be prudent to use the adversarial sample detection technique from Chapter 3 in scenarios where security is of the utmost importance.

I argue that the explanation consistency metric proposed in Chapter 4 should be used in any and all machine learning applications. Although named explanation consistency, this metric does not just measure the quality of model explanations; Chapter 6.1 in particular shows how it measures the quality of learned features. Thus, it would be suitable to be used as an indicator of model quality and should be considered (alongside classical metrics such as performance and loss) when choosing the best model architecture and (hyper)parameter setup for any task; models with higher explanation consistency should learn a better set of features and be the better performing model. This should be of importance to any ML practitioner, no matter their field of application, but is especially important where the model's reasoning is of paramount importance (i.e. end users will be interested in *why* the decision was made). In almost all applications, the original definition for explanation consistency that uses Logistic Regression (Equation (4.2)) will be the most suitable. However, if it is known that accurate Jensen-Shannon Divergence estimations can be quickly and easily computer for your task then one may also consider using the definition in Equation (5.2).

In an ideal world, the DEE architecture and training technique suggested in Chapter 5 would be used in any setting, as Chapters 5 and 6 show they provide superior performance, explainability, transparency and privacy to current state of the art techniques on all of the datasets tested. However, as is explained in Chapter 5 and Chapter 7.3, DEEs in their current incarnation do have some limitations. Most notably, they are extremely computationally expensive to train and as such may not be worth the trade-off in applications where traditional ML models provide adequate performance and explainability/transparency is not as important (there are many such domains, but examples may include recommender systems or business intelligence applications). Despite this limitation, DEEs have a wide scope for use in the sensitive scenarios that have been focused on throughout this work. In domains such as healthcare and biology, where an incorrect decision can have significant consequences and end users are particularly interested in a model's explanation, the time required to train a DEE is worth it to get a state of the art model in terms of performance, explainability, transparency and privacy. Moreover, any applications where data privacy is important should consider using DEEs; Chapter 6.2 shows that they are able to outperform the (already computationally and network intensive) federated learning and differential privacy techniques when it comes to both model performance and privacy, and so should be suitable as a drop in replacement for settings where these techniques are already commonplace.

As DEEs introduce some new hyperparameters that can be tuned, as well as the need to choose a suitable architecture for the discriminator, it is important to give some guidance on how these can be set. As previously suggested in Chapter 5.1.2, the task of the discriminator is simple and it is important to choose its architecture appropriately, ensuring it is not able to overfit. For most data modalities, small MLPs (or even a Logistic Regression) model should be sufficient, although for large image datasets it may be more useful to use a small CNN. The discriminator update rate, n, should be chosen similarly: as the discriminator's task is relatively simple, there is no need to update it at every epoch. Instead, setting $n = 2$ should be ample for most tasks however, if the data is extremely small and/or simple, then higher values such as $n = 4, 6$ may be more appropriate. The value of β should be set such that the loss of the discriminator and the loss of the downstream ensemble are of the same order of magnitude: this will vary by task, but can easily be chosen through simple inspection of the losses. Of course, both n and β can be (and when DEEs are being used in production, should) included in any hyperparameter searches that are carried out.

All of the techniques proposed throughout this thesis have possible applications. Hopefully, the suggestions in this section have made it easier for ML practitioners to understand when and where they may be applied and, most importantly, how they should be used in order to gain the most use out of them.

7.3 Limitations & Future Work

Although all of the methods presented in this thesis show promising results on the datasets they were evaluated on, there are still some areas they could be improved. As each chapter ends with a brief discussion of its limitations, this section consolidates these limitations and considers them as a whole, suggesting avenues for future work that could answer some of the open questions that have been left exposed by this work.

Firstly, throughout this work I have focused on experimenting with as many real-world datasets from sensitive domains as possible. However, by their very nature this data is extremely hard to come by, with not many such datasets being publicly available (in fact, this thesis has used most, if not all, of suitable, publicly-accessible data). It would be prudent, if the opportunity arises, to test all of the methods in this thesis on a wider range of datasets and data modalities where possible; this would give practitioners a better understanding of where each technique is best applied. In particular, it would be extremely interesting to confirm that the EGD experiments in Chapter 6.1 are reproducible on data from other settings, and of different modalities.

Although the DEE architecture introduced in Chapter 5 is shown throughout the rest of the thesis to have many advantageous properties, it is extremely costly to train. During training, a large enough number of sub-models S must be used, and feature attributions must be calculated for every training instance, for each of the S sub-models, at each epoch. This makes it both memory- (one must choose S large enough that there are enough sub-models to provide sufficient variety, whilst ensuring it is small enough such that all model parameters fit into memory) and time-inefficient (almost all suitable explainability techniques are costly to compute). These limitations may make DEEs impossible to train on large, complex datasets, although it should be noted that the compute cost at inference-time is the same as any traditional ensemble model and so this is a one-time issue. Future work may include investigating ways to either: increase the efficiency of the explainability techniques used during training, or remove the need for DEEs to be of an ensemble architecture. A possible future direction for this may be to enforce some form of statistical measure on the set of explanations (from a single model) during training: for example, reduce the amount of noise in the explanations by measuring its divergence from a Gaussian distribution.

It would also be interesting to study the relationship between the proposed DEE architecture and feature selection methods. In particular, one could design a set of experiments to investigate whether the discriminator part of the network ever encourages the sub-models to completely ignore some set of features. Of course, these experiments should also be extended to examine the effect feature selection methods have on the explanation consistency of a model - it may be possible to combine both feature selection and the DEE architecture to address the performance trade-off present (with the idea being that, as the DEE would need to work on fewer features, it could place more of its power on the downstream task rather than on increasing explanation consistency).

Many of the experiments in this thesis point to an issue with traditional models learning spurious features: this would explain the explanation inconsistency results in Chapter 4, as I hypothesise that if a model were learning truly causally-related features, they would remain largely the same each run. It could also be argued that the results of Chapter 6.1 suggests that DEEs are more likely to be learning causal features, as they better align with the expert's EGD. However, these conclusions are very strong statements, and I have not formally proven them; this task could be the topic of a whole other thesis, as it is not even clear if the mathematical tools needed for such an analysis have been developed. It would be extremely interesting for future work to analyse explanation consistency and DEEs from a causal perspective, experimenting on datasets with known underlying causal graphs to examine whether or not DEEs are indeed better at learning causal features. However, this work would be need to be extremely rigorous and it may even be the case that the correct tools to do this type of analysis have not yet been developed.

The overarching aim of this thesis was to develop techniques that can be used in highly critical applications, such as those in healthcare and finance. Whilst the experiments throughout this thesis have been designed to evaluate the technique's usefulness in such scenarios as much as possible, it would be very valuable to also get a domain expert's opinion. For example, it would be interesting to ask a radiologist if the DEE's model explanations from Chapter 6.1 does indeed increase the trust they would place in the model. This type of human evaluation can be extremely difficult to collect (due to the time cost for everyone involved), but is of paramount importance should we want to use the ideas presented in this thesis in practice.

An intriguing avenue for future work, which is not directly related to but rather inspired by the ideas explored in this thesis, would be to investigate how explainability techniques can be used in the generation of synthetic data. The results in Chapter 3 suggest that it may be possible to detect synthetic data via feature attributions - it is likely that generated features differ enough from real features that explainability techniques will pick up on them - which would mean that models trained on synthetic data would be less likely to generalise to real data distributions, and perhaps have even worse explanation consistency. If one could take the ideas of Chapter 5 and Chapter 6.1 and apply them to generative models such as Generative Adversarial Networks, it may be possible to create new data samples which downstream classification models treat the same as real data. This would be hugely beneficial in many of the high-risk applications that have been focused on throughout this thesis, as it would allow for the generation of a nearly infinite amount of high-quality data from only a small amount of actual data.

Finally, there is always the opportunity to continue using explainability to aid our understanding of black-box deep learning models. This thesis has shown that by focusing on creating models that learn high-quality features, it is possible to address numerous issues with traditional deep neural networks at once. Namely, Chapter 4 shows that explainability can be used to uncover previously unknown issues with the training of deep models, and Chapter 5 shows that explanations can be a solution to this (and other) issues. It would be encouraging to see future work take the general approaches proposed throughout this thesis (of using explainability to aid our understanding of the inner workings of machine learning, rather than relying on instance-based explainability) to further investigate ML training. For example: can explanations be used to understand what makes a model fail? This could be extended further - can explanations uncover *why* a model fails? This thesis has shown that explanations are somewhat of an untapped resource, and I hope this work encourages more in-depth work with them in the future.

7.4 Conclusion

Throughout this thesis, we have seen how advances in deep learning explainability can be used to gain insight into neural network model training, as well as how these techniques can be used to improve the robustness and trustworthiness of deep learning techniques. Each chapter can be seen as a step towards understanding, and often improving, why DL models face barriers when being applied to sensitive scenarios such as healthcare, bioinformatics and finance. In this Chapter, I review the contributions of the thesis and summarise how each novel technique introduced can be used to address one or more of the barriers identified in Chapter 2.1.8.

Chapter 3 uses off-the-shelf explainability techniques to create a novel detection method for adversarial attacks that is able to out-perform current state-of-the-art techniques. Importantly, through the use of the proposed (V)AE detection method, one is able to create a model for any dataset that is able to protect against even unseen attack types. Through thorough experimentation on a wide variety of datasets, it was shown that the proposed techniques provide extremely good attack detection accuracy whilst also being extremely computationally-light at inference time. This makes it a viable approach to be used as part of a deep learning pipeline in sensitive scenarios: for example, one could imagine it being used in a financial fraud detection system to detect malicious attacks before they are passed to the final deep learning classification model.

The results of Chapter 3 raise an intriguing point, in that it shows that DL model explanations are extremely sensitive to even small changes to a model's input. Chapter 4 takes this idea and extends it even further, investigating whether model feature attributions can be also be used to uncover changes to the model itself. Through this novel use of explainability techniques, I show that not only can explanations be used to differentiate between two otherwise identical architectures, but that changing model hyperparameters that are orthogonal to the downstream task (such as the random seed, or order of the training data) results in vastly different model explanations (even when the model's final output and classification are extremely similar). I confirm that this is the result across numerous different data modalities, data and architecture complexities, and explainability techniques. Through verification of the explainability method's faithfulness to the underlying model via the use of established metrics such as explanation sensitivity and infidelity, I confirm that this is indeed uncovering inherent differences in the model's learned parameters and not an issue with the explainability techniques themselves.

The results of these experiments lead to the development of a new quality metric for neural networks, Explanation Consistency. Chapter 4 presents both a general framework for Explanation Consistency, allowing it to be adapted to a wide range of scenarios, as well as a concrete implementation for general use. Thorough experimentation of both toy and real-world tasks shows that traditional deep learning models have extremely low Explanation Consistency and that even current state-of-the-art architectures such as Hyperparameter Ensembles, which are designed precisely to address the issue of model robustness and generalisability, suffer from the same issues. As experiments on kernel-based methods such as SVMs show that they do not suffer from the same issues, I hypothesise that explanation inconsistency is a result of the stochastic nature of neural network training. By exploring the similarity of layer parameters between two trained models (of identical architectures but trained with different random seeds) through the use of SVCCA, I corroborate the explanation consistency results: the final layers of a network are similar (and hence they produce similar outputs) whilst the middle layers are significantly different, which is resulting in widely different feature attributions.

The low explanation consistency of modern neural networks highlights one of the key barriers facing models when they are applied to sensitive scenarios: trustworthiness. The lack of explanation consistency suggests that models may not be as robust as they seem, and people are less likely to trust a model if they believe that they are converging to use significantly different features each time they are trained; one would expect a model to use the same set of causally-related features each time. This inspires the methods presented in Chapter 5, where I aim to address the explanation inconsistency issue. Firstly, motivated by my findings in Chapter 4 that ensemble architectures have slightly higher explanation consistency than traditional methods, I propose a post-processing technique that can be applied to the explanations of any machine learning algorithm. Extensive evaluation on tabular data shows that this algorithm, which works by removing the least important features from an ensemble of models, achieves significantly higher explanation consistency than baseline models. However, as explained in Chapter 5.1.1, there are a number of issues with this method. Most notably it does not actually change the features learned by the underlying model, only the resulting explanations, and so one can argue that the models themselves are still extremely inconsistent, and so the problem persists.

Chapter 5.1.2 addresses this issue by presenting an entirely new deep learning model architecture and training algorithm that attempts to encapsulate the previous post-processing technique inside model training. Motivated by the success of using multiple explanations, as exhibited by the post-processing technique, the suggested Deep Explanation Ensemble (DEE) is the first such method to use model explanations during training to encourage the final model to "average out" noisy features. Extensive testing of this new technique on tabular datasets from a wide range of sensitive domains shows that not only do DEEs drastically improve explanation consistency, but they do so without significantly affecting model performance as well. A thorough ablation study confirmed that these improvements are due to the use of explanations during training, and cannot be replicated through simple techniques such as checkpoint averaging, sub-model averaging or a combination of the two. The efficacy of DEEs make them a prime candidate for use in sensitive scenarios: the greatly increased explanation consistency means that the models are inherently more trustworthy, and I hypothesise that it may be a result of an increased reliance on causal features (and consequently reduced reliance on spurious correlations).

The properties of DEEs are further explored in Chapter 6. Firstly, in Chapter 6.1, we broaden the experiments to a second data modality: images. Specifically, I further show the usefulness of the technique in a real-world sensitive application by evaluating model performance on chest x-ray (CXR) diagnosis. By comparing model explanations to an expert's Eye-Gaze Data (EGD), it is shown that DEEs learn a set of features which are much closer to those that would be used by clinical experts. This, alongside the increased performance and explanation consistency when compared with even state-of-the-art models, suggests that DEEs are perhaps learning to use a better set of features and that traditional architectures are over-reliant on spurious correlations. As the DEE's feature attributions align so closely with the expert's, this will likely increase a clinician's trust in our model, addressing one of the big issues facing DL models in healthcare (and any other sensitive application). Importantly, DEEs even out perform models that use EGD during training - which is extremely expensive and time-consuming to collect - further solidifying their usefulness in practice.

Finally, inspired by the success of the success of the explainability-based adversarial attack detection techniques originally presented in Chapter 3, Chapter 6.2 investigates how explainability can be applied to Membership Inference Attacks (MIA). I present a novel black-box attack method that can be applied to any deep learning model, without the need for access to any of its training data, that utilises model explanations to infer membership inference. Comparison with existing black-box MIAs shows that this proposed attack outperforms attacks that use a model's outputs only, a result that may be expected having seen the results of Chapter 3, and how sensitive explanations are to a model's inner workings in Chapter 4. Importantly, I show that even differentially-private models are susceptible to this type of attack, highlighting the importance of including the information that is contained in model explanations when analysing the privacy provided by a deep learning algorithm.

I then show that DEEs are not only robust to this type of explainability-based MIA (which may be expected due to its utilisation of explanations during training), but also that they are robust to traditional MIAs as well. Compellingly, DEEs achieve this level of privacy without the same privacy-performance trade-off that is present in differentially-private networks, making DEEs a much more viable proposition in applications where a user's data privacy is of paramount importance. Chapter 6.2 also explains how DEEs can be applied to Federated Learning (FL) settings, and evaluates their efficacy under FL assumptions, showing that they provide better data privacy whilst keeping model performance the same.

The result of all of this work, then, is that I have shown how explainability techniques can be used to address the three main barriers facing DL models in sensitive scenarios that were identified in Chapter 2.1.8: trustworthiness (Chapter 5 and Chapter 6.1), robustness (Chapters 4 and 5) and user privacy (Chapter 3 and Chapter 6.2). Throughout the thesis, all of the proposed methods have purposely been evaluated on datasets from sensitive domains (such as healthcare and bioinformatics) to show how they can be applied to real-world applications. I hope that the results contained in this thesis encourage DL practitioners and domain experts to continue to explore how models can be applied to real-world settings, and to continue to "open up" the black-box of neural networks and make them more approachable to end-users and non-DL experts alike.

Bibliography

[1] A. Karargyris, S. Kashyap, I. Lourentzou, J. T. Wu, A. Sharma, M. Tong, S. Abedin, D. Beymer, V. Mukherjee, E. A. Krupinski, and M. Moradi, "Creation and validation of a chest x-ray dataset with eye-tracking and report dictation for ai development," *Scientific Data*, vol. 8, p. 92, Mar 2021. (document), 2.1, 2.1.5, 2.1, 2.2.6, 6.1, 6.1.1, 6.1.3, 6.1.4, 6.1, 6.1.5, 6.2.4, 6.2.4

[2] M. Watson, "Learning to mimic: Supporting code." `https://github.com/mattswatson/learning-to-mimic`, 2022. (document), 6.5

[3] S. P. Singh, A. Kumar, H. Darbari, L. Singh, A. Rastogi, and S. Jain, "Machine translation using deep learning: An overview," in *2017 international conference on computer, communications and electronics (comptelix)*, pp. 162–167, IEEE, 2017. 1

[4] H. Fujiyoshi, T. Hirakawa, and T. Yamashita, "Deep learning-based image recognition for autonomous driving," *IATSS research*, vol. 43, no. 4, pp. 244–252, 2019. 1

[5] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022. 1

[6] A. van der Gijp, C. J. Ravesloot, H. Jarodzka, M. F. van der Schaaf, I. C. van der Schaaf, J. P. J. van Schaik, and T. J. Ten Cate, "How visual search relates to visual diagnostic performance: a narrative systematic review of eye-tracking research in radiology," *Adv. Health Sci. Educ. Theory Pract.*, vol. 22, pp. 765–787, Aug. 2017. 1, 2, 6.1

[7] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: Threats and solutions," *IEEE Security & Privacy*, vol. 17, no. 2, pp. 49–58, 2019. 1, 2.4

[8] S. Yucer, S. Akçay, N. Al-Moubayed, and T. P. Breckon, "Exploring racial bias within face recognition via per-subject adversarially-enabled data augmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 18–19, 2020. 1, 2.3

[9] C. J. Kelly, A. Karthikesalingam, M. Suleyman, G. Corrado, and D. King, "Key challenges for delivering clinical impact with artificial intelligence," *BMC Medicine*, vol. 17, p. 195, Oct 2019. 1, 1.1, 5.2.1

[10] A. D'Amour, K. A. Heller, D. Moldovan, B. Adlam, B. Alipanahi, A. Beutel, C. Chen, J. Deaton, J. Eisenstein, M. D. Hoffman, F. Hormozdiari, N. Houlsby, S. Hou, G. Jerfel, A. Karthikesalingam, M. Lucic, Y. Ma, C. Y. McLean, D. Mincu, A. Mitani, A. Montanari, Z. Nado, V. Natarajan, C. Nielson, T. F. Osborne, R. Raman, K. Ramasamy, R. Sayres, J. Schrouff, M. Seneviratne, S. Sequeira, H. Suresh, V. Veitch, M. Vladymyrov, X. Wang, K. Webster, S. Yadlowsky, T. Yun, X. Zhai, and D. Sculley, "Underspecification presents challenges for credibility in modern machine learning," *CoRR*, vol. abs/2011.03395, 2020. 1, 2.3.1, 4

[11] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021. 1, 2.3.1

[12] European Commission, "2018 reform of eu data protection rules," May 2018. https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf. 1, 1.1, 2.1.8, 6.2

[13] J. Morley and I. Joshi, "Artificial intelligence: How to get it right. putting policy into practice for safe data-driven innovation in health and care.," *NHS*, 2019. 1, 1.1, 2, 2.1.8

[14] E. Çallı, E. Sogancioglu, B. van Ginneken, K. G. van Leeuwen, and K. Murphy, "Deep learning for chest x-ray analysis: A survey," *Medical Image Analysis*, vol. 72, p. 102125, 2021. 1, 2.1.4, 6.1

[15] C. M. Jones, L. Danaher, M. R. Milne, C. Tang, J. Seah, L. Oakden-Rayner, A. Johnson, Q. D. Buchlak, and N. Esmaili, "Assessment of the effect of a comprehensive chest radiograph deep learning model on radiologist reports and patient outcomes: a real-world observational study," *BMJ Open*, vol. 11, no. 12, 2021. 1

[16] S. Benjamens, P. Dhunnoo, and B. Meskó, "The state of artificial intelligence-based fda-approved medical devices and algorithms: an online database," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–8, 2020. 1, 2, 2.1.8, 6.1

[17] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Interpretable.* Morisville, North Carolina: Lulu, 2019. 1, 2.2, 2.2.1, 2.2.6, 2.2.6, 2.3.2, 6.2.2

[18] R. Geirhos, J. Jacobsen, C. Michaelis, R. S. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *CoRR*, vol. abs/2004.07780, 2020. 1, 2.3, 2.3.1, 6.1.1, 6.1.4

[19] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 268–282, 2018. 1, 2.4.4, 6.2, 6.2.4

[20] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio, "Fantastic generalization measures and where to find them," in *8th International Conference on Learning Representations, ICLR 2020*, OpenReview.net, 2020. 1, 2.3.1, 5.1.2

[21] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA* (I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, eds.), pp. 4765–4774, 2017. 1, 1.1, 5.1.3, 6.2.2, 6.2.4

[22] V. Jahmunah, E. Ng, R.-S. Tan, S. L. Oh, and U. R. Acharya, "Explainable detection of myocardial infarction using deep learning models with grad-cam technique on ecg signals," *Computers in Biology and Medicine*, vol. 146, p. 105550, 2022. 1, 2.2.5

[23] R. Gaudel, L. Galárraga, J. Delaunay, L. Rozé, and V. Bhargava, "s-lime: Reconciling locality and fidelity in linear explanations," *arXiv*, 2022. 1, 2.2.2

[24] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of Healthcare Informatics Research*, vol. 5, pp. 1–19, Mar 2021. 1, 2.4.4, 6.2

[25] F. Wenzel, J. Snoek, D. Tran, and R. Jenatton, "Hyperparameter ensembles for robustness and uncertainty quantification," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020. 1, 1.1, 2.3.2, 4.3, 5.1.2, 5.2.5

[26] J. Kaddour, A. Lynch, Q. Liu, M. J. Kusner, and R. Silva, "Causal machine learning: A survey and open problems," 2022. 1

[27] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. L. Ball, K. S. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," *CoRR*, vol. abs/1901.07031, 2019. 1.1, 2.1, 2.1.2, 3.1.1, 4.3

[28] Z. Alhassan, M. Watson, D. Budgen, R. Alshammari, A. Alessa, and N. Al Moubayed, "Improving current glycated hemoglobin prediction in

adults: Use of machine learning algorithms with electronic health records," *JMIR Med Inform*, vol. 9, p. e25237, May 2021. 1.1, 2, 2.1, 2.1.7, 5.2.1, 6.1

[29] N. Woodruff, A. Enshaei, and B. A. S. Hasan, "Fully-automatic pipeline for document signature analysis to detect money laundering activities," *CoRR*, vol. abs/2107.14091, 2021. 1.1

[30] J. Larson and J. Angwin, "Machine bias," May 2016. 1.1, 2.1, 6.2.4, 6.2.4

[31] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *J. Artif. Intell. Res.*, vol. 70, pp. 245–317, 2021. 1.1

[32] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016* (B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, eds.), pp. 1135–1144, ACM, 2016. 1.1, 2.2.2

[33] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, 2020. 1.1

[34] P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," *CoRR*, vol. abs/1803.05407, 2018. 1.1, 5.1.7

[35] P. Saranrittichai, C. K. Mummadi, C. Blaiotta, M. Munoz, and V. Fischer, "Overcoming shortcut learning in a target domain by generalizing basic visual factors from a source domain," in *European Conference on Computer Vision*, pp. 294–309, Springer, 2022. 1.1, 2.3.1

[36] P. Nakkiran, B. Neyshabur, and H. Sedghi, "The deep bootstrap framework: Good online learners are good offline generalizers," *arXiv preprint arXiv:2010.08127*, 2020. 1.1, 2.3.1

[37] S. Yucer, F. Tektas, N. Al Moubayed, and T. P. Breckon, "Measuring hidden bias within face recognition via racial phenotypes," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 995–1004, 2022. 1.1

[38] W. E, C. Ma, S. Wojtowytsch, and L. Wu, "Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't," *CoRR*, vol. abs/2009.10713, 2020. 1.1, 2.3.1

[39] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in Bioinformatics*, vol. 19, pp. 1236–1246, 05 2017. 2

[40] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Y. Ding, A. Bagul, C. Langlotz, K. S. Shpanskaya, M. P. Lungren, and A. Y. Ng, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *CoRR*, vol. abs/1711.05225, 2017. 2, 3.1.1, 4.3

[41] W. Zhu, L. Xie, J. Han, and X. Guo, "The application of deep learning in cancer prognosis prediction," *Cancers*, vol. 12, no. 3, 2020. 2

[42] H. Haenssle, C. Fink, F. Toberer, J. Winkler, W. Stolz, T. Deinlein, R. Hofmann-Wellenhof, A. Lallas, S. Emmert, T. Buhl, M. Zutt, A. Blum, M. Abassi, L. Thomas, I. Tromme, P. Tschandl, A. Enk, A. Rosenberger, C. Alt, M. Bachelerie, S. Bajaj, A. Balcere, S. Baricault, C. Barthaux, Y. Beckenbauer, I. Bertlich, A. Blum, M.-F. Bouthenet, S. Brassat, P. Marcel Buck, K. Buder-Bakhaya, M.-L. Cappelletti, C. Chabbert, J. De Labarthe, E. DeCoster, T. Deinlein, M. Dobler, D. Dumon, S. Emmert, J. Gachon-Buffet, M. Gusarov, F. Hartmann, J. Hartmann, A. Herrmann, I. Hoorens, E. Hulstaert, R. Karls, A. Kolonte, C. Kromer, A. Lallas, C. Le Blanc Vasseux, A. Levy-Roy, P. Majenka, M. Marc, V. M. Bourret, N. Michelet-Brunacci, C. Mitteldorf, J. Paroissien, C. Picard, D. Plise, V. Reymann, F. Ribeaudeau, P. Richez, H. Roche Plaine, D. Salik, E. Sattler, S. Schäfer, R. Schneiderbauer, T. Secchi, K. Talour, L. Trennheuser, A. Wald, P. Wölbing, and P. Zukervar, "Man against machine reloaded: performance of a market-approved convolutional neural network in classifying a broad spectrum of skin lesions in comparison with 96 dermatologists working under less artificial conditions," *Annals of Oncology*, vol. 31, no. 1, pp. 137–143, 2020. 2

[43] A. Jacovi, A. Marasović, T. Miller, and Y. Goldberg, "Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai," 2020. 2

[44] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. http://www.deeplearningbook.org. 2

[45] A. E. W. Johnson, T. J. Pollard, S. J. Berkowitz, N. R. Greenbaum, M. P. Lungren, C.-y. Deng, R. G. Mark, and S. Horng, "MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports," *Scientific Data*, vol. 6, p. 317, Dec. 2019. 2.1, 2.1.4, 3.1.1, 6.1.1

[46] A. E. W. Johnson, T. J. Pollard, S. J. Berkowitz, N. R. Greenbaum, M. P. Lungren, C. Deng, R. G. Mark, and S. Horng, "MIMIC-CXR: A large publicly available database of labeled chest radiographs," *CoRR*, vol. abs/1901.07042, 2019. 2.1, 2.1.3

[47] B. B. Khomtchouk, "Codon usage bias levels predict taxonomic identity and genetic composition," *bioRxiv*, 2020. 2.1, 2.1, 2.1.6, 5.2.1

[48] M. Ghassemi and S. Mohamed, "Machine learning and health need better values," *npj Digital Medicine*, vol. 5, p. 51, Apr 2022. 2.1

[49] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 1058–1066, PMLR, 17–19 Jun 2013. 2.1

[50] S. Caldas, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, "LEAF: A benchmark for federated settings," *CoRR*, vol. abs/1812.01097, 2018. 2.1, 6.2.4, 6.2.4

[51] F. Lai, Y. Dai, S. S. Singapuram, J. Liu, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Fedscale: Benchmarking model and system performance of federated learning at scale," in *International Conference on Machine Learning (ICML)*, 2022. 2.1, 6.2.4, 6.2.4

[52] D. Dua and C. Graff, "UCI machine learning repository," 2017. 2.1, 6.2.4, 6.2.4

[53] T. Humphries, S. Oya, L. Tulloch, M. Rafuse, I. Goldberg, U. Hengartner, and F. Kerschbaum, "Investigating membership inference attacks under data dependencies," *arXiv e-prints*, pp. arXiv–2010, 2020. 2.1, 2.4.4, 6.2, 6.2.4

[54] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. F. Stewart, "RETAIN: an interpretable predictive model for healthcare using reverse time attention mechanism," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pp. 3504–3512, 2016. 2.1, 3.1.1

[55] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis," in *Biomedical Image Processing and Biomedical Visualization* (R. S. Acharya and D. B. Goldgof, eds.), vol. 1905, pp. 861 – 870, International Society for Optics and Photonics, SPIE, 1993. 2.1.1, 5.2.1

[56] J. W. Smith, J. E. Everhart, W. Dickson, W. C. Knowler, and R. S. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the annual symposium on computer application in medical care*, p. 261, American Medical Informatics Association, 1988. 2.1.1

[57] L. P. Joseph, E. A. Joseph, and R. Prasad, "Explainable diabetes classification using hybrid bayesian-optimized tabnet architecture," *Computers in Biology and Medicine*, p. 106178, 2022. 2.1.1

[58] R. Sauber-Cole and T. M. Khoshgoftaar, "The use of generative adversarial networks to alleviate class imbalance in tabular data: a survey," *Journal of Big Data*, vol. 9, p. 98, Aug 2022. 2.1.1

[59] A. Johnson, M. Lungren, Y. Peng, Z. Lu, R. Mark, S. Berkowitz, and S. Horng, "Mimic-cxr-jpg - chest radiographs with structured labels (version 2.0.0)," *PhysioNet*, 2019. 2.1.4, 4.3

[60] B. Chu, V. Madhavan, O. Beijbom, J. Hoffman, and T. Darrell, "Best practices for fine-tuning visual classifiers to new domains," in *European conference on computer vision*, pp. 435–442, Springer, 2016. 2.1.4

[61] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. 2.1.4, 3.1.1

[62] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. 2.1.4

[63] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. 2.1.4

[64] I. Kumar, S. P. Singh, and Shivam, "Chapter 26 - machine learning in bioinformatics," in *Bioinformatics* (D. B. Singh and R. K. Pathak, eds.), pp. 443–456, Academic Press, 2022. 2.1.6

[65] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, pp. 583–589, Aug 2021. 2.1.6

[66] D. Azouri, S. Abadi, Y. Mansour, I. Mayrose, and T. Pupko, "Harnessing machine learning to guide phylogenetic-tree search algorithms," *Nature Communications*, vol. 12, p. 1983, Mar 2021. 2.1.6

[67] K. Abouelmehdi, A. Beni-Hessane, and H. Khaloufi, "Big healthcare data: preserving security and privacy," *Journal of Big Data*, vol. 5, p. 1, Jan 2018. 2.1.7

[68] J. H. Thorpe and E. A. Gray, "Big data and public health: navigating privacy laws to maximize potential," *Public Health Rep.*, vol. 130, pp. 171–175, Mar. 2015. 2.1.7

[69] M. Elliot, K. O'hara, C. Raab, C. M. O'Keefe, E. Mackey, C. Dibben, H. Gowans, K. Purdam, and K. McCullagh, "Functional anonymisation: personal data and the data environment," *Computer Law & Security Review*, February 2018. 2.1.7

[70] Z. Alhassan, D. Budgen, R. Alshammari, T. Daghstani, A. S. McGough, and N. Al Moubayed, "Stacked denoising autoencoders for mortality risk prediction using imbalanced clinical data," in *2018 17th IEEE International Conference*

on Machine Learning and Applications (ICMLA), pp. 541–546, IEEE, 2018. 2.1.7

[71] Z. Alhassan, A. S. McGough, R. Alshammari, T. Daghstani, D. Budgen, and N. Al Moubayed, "Type-2 diabetes mellitus diagnosis from time series clinical data using deep learning models," in *International Conference on Artificial Neural Networks*, pp. 468–478, Springer, 2018. 2.1.7

[72] C. J. Kelly, A. Karthikesalingam, M. Suleyman, G. Corrado, and D. King, "Key challenges for delivering clinical impact with artificial intelligence," *BMC Medicine*, vol. 17, p. 195, Oct 2019. 2.1.8, 2.2, 3

[73] E. Harwich and K. Laycock, *Thinking on its own: AI in the NHS.* Reform, 2018. 2.1.8

[74] E. Vayena, A. Blasimme, and I. G. Cohen, "Machine learning in medicine: Addressing ethical challenges," *PLOS Medicine*, vol. 15, pp. 1–4, 11 2018. 2.1.8

[75] B. M. Li, F. Corponi, G. Anmella, A. Mas, M. Sanabra, I. Pacchiarotti, M. Valentí, A. Giménez-Palomo, M. Garriga, I. Agasi, A. Bastidas, T. Fernández-Plaza, N. Arbelo, M. Cavero, C. García-Rizo, M. Bioque, N. Verdolini, S. Madero, A. Murru, I. Grande, S. Amoretti, V. Ruiz, G. Fico, M. De Prisco, V. Oliva, E. Vieta, and D. Hidalgo-Mazzei, "Can machine learning with data from wearable devices distinguish disease severity levels and generalise across patients? a pilot study in mania and depression," *medRxiv*, 2022. 2.1.8

[76] J. Hoeksma, "The nhs's care.data scheme: what are the risks to privacy?," *BMJ*, vol. 348, 2014. 2.1.8

[77] Z. Zuo, M. Watson, D. Budgen, R. Hall, C. Kennelly, and N. Al Moubayed, "Data anonymization for pervasive health care: Systematic literature mapping study," *JMIR Med Inform*, vol. 9, p. e29871, Oct 2021. 2.1.8, 6.2

[78] V. Belle and I. Papantonis, "Principles and practice of explainable machine learning," *Frontiers in Big Data*, vol. 4, p. 39, 2021. 2.2, 2.2, 6.2

[79] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell, "What do we need to build explainable AI systems for the medical domain?," *CoRR*, vol. abs/1712.09923, 2017. 2.2, 2.2, 2.2.1, 4

[80] F. Harder, M. Bauer, and M. Park, "Interpretable and differentially private predictions," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 4083–4090, AAAI Press, 2020. 2.2

[81] J. Mingers, "An empirical comparison of pruning methods for decision tree induction," *Machine learning*, vol. 4, no. 2, pp. 227–243, 1989. 2.2

[82] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020. 2.2.1

[83] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, and H. Müller, "Causability and explainability of artificial intelligence in medicine," *WIREs Data Mining and Knowledge Discovery*, vol. 9, no. 4, p. e1312, 2019. 2.2.1

[84] G. Visani, E. Bagli, F. Chesani, A. Poluzzi, and D. Capuzzo, "Statistical stability indices for lime: Obtaining reliable explanations for machine learning models," *Journal of the Operational Research Society*, vol. 73, no. 1, pp. 91–101, 2022. 2.2.2

[85] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA* (I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, eds.), pp. 4765–4774, 2017. 2.2.3, 2.2.3, 3.1.3, 3.2.3, 4.3

[86] S. Lipovetsky and M. Conklin, "Analysis of regression in game theory approach," *Applied Stochastic Models in Business and Industry*, vol. 17, no. 4, pp. 319–330, 2001. 2.2.3

[87] B. Dimanov, U. Bhatt, M. Jamnik, and A. Weller, "You shouldn't trust me: Learning models which conceal unfairness from multiple explanation methods," in *Proceedings of the Workshop on Artificial Intelligence Safety, SafeAI@AAAI 2020*, vol. 2560, pp. 63–73, CEUR-WS.org, 2020. 2.2.3

[88] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, vol. 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328, PMLR, 2017. 2.2.4, 4.3

[89] P. Sturmfels, S. Lundberg, and S.-I. Lee, "Visualizing the impact of feature attribution baselines," *Distill*, vol. 5, no. 1, p. e22, 2020. 2.2.4

[90] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017. 2.2.5, 6.1.3, 6.2.2

[91] R. L. Draelos and L. Carin, "Use hirescam instead of grad-cam for faithful explanations of convolutional neural networks," 2020. 2.2.5

[92] T. Viering, Z. Wang, M. Loog, and E. Eisemann, "How to manipulate cnns to make them lie: the gradcam case," *arXiv preprint arXiv:1907.10901*, 2019. 2.2.5

[93] N. I. Papandrianos, A. Feleki, S. Moustakidis, E. I. Papageorgiou, I. D. Apostolopoulos, and D. J. Apostolopoulos, "An explainable classification method of spect myocardial perfusion images in nuclear cardiology using deep learning and grad-cam," *Applied Sciences*, vol. 12, no. 15, p. 7592, 2022. 2.2.5

[94] C. Yeh, C. Hsieh, A. S. Suggala, D. I. Inouye, and P. Ravikumar, "On the (in)fidelity and sensitivity of explanations," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pp. 10965–10976, 2019. 2.2.6, 2.2.6, 4, 4.2

[95] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3681–3688, Jul. 2019. 2.2.6

[96] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, "Evaluating the quality of machine learning explanations: A survey on methods and metrics," *Electronics*, vol. 10, no. 5, 2021. 2.2.6

[97] Z. Buçinca, P. Lin, K. Z. Gajos, and E. L. Glassman, "Proxy tasks and subjective measures can be misleading in evaluating explainable AI systems," in *IUI '20: 25th International Conference on Intelligent User Interfaces, Cagliari, Italy, March 17-20, 2020* (F. Paternò, N. Oliver, C. Conati, L. D. Spano, and N. Tintarev, eds.), pp. 454–464, ACM, 2020. 2.2.6

[98] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA*, vol. 81 of *Proceedings of Machine Learning Research*, pp. 77–91, PMLR, 2018. 2.3

[99] V. Nagarajan, A. Andreassen, and B. Neyshabur, "Understanding the failure modes of out-of-distribution generalization," *CoRR*, vol. abs/2010.15775, 2020. 2.3, 2.3.1, 2.4.4, 4

[100] Y.-Y. Yang and K. Chaudhuri, "Understanding rare spurious correlations in neural networks," 2022. 2.3, 6.1.4

[101] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study," *PLOS Medicine*, vol. 15, pp. 1–17, 11 2018. 2.3

[102] H. B. Syeda, M. Syed, K. W. Sexton, S. Syed, S. Begum, F. Syed, F. Prior, and F. Yu, "Role of Machine Learning Techniques to Tackle the COVID-19 Crisis: Systematic Review," *JMIR Med Inform*, vol. 9, p. e23811, Jan 2021. 2.3

[103] I. von Borzyskowski, A. Mazumder, B. Mateen, and M. Wooldridge, *Data science and AI in the age of COVID-19*. The Alan Turing Institute, 2021. 2.3

[104] M. Roberts, D. Driggs, M. Thorpe, J. Gilbey, M. Yeung, S. Ursprung, A. I. Aviles-Rivero, C. Etmann, C. McCague, L. Beer, J. R. Weir-McCall, Z. Teng, E. Gkrania-Klotsas, A. Ruggiero, A. Korhonen, E. Jefferson, E. Ako, G. Langs, G. Gozaliasl, G. Yang, H. Prosch, J. Preller, J. Stanczuk, J. Tang, J. Hofmanninger, J. Babar, L. E. Sánchez, M. Thillai, P. M. Gonzalez, P. Teare, X. Zhu, M. Patel, C. Cafolla, H. Azadbakht, J. Jacob, J. Lowe, K. Zhang, K. Bradley, M. Wassin, M. Holzer, K. Ji, M. D. Ortet, T. Ai, N. Walton, P. Lio, S. Stranks, T. Shadbahr, W. Lin, Y. Zha, Z. Niu, J. H. F. Rudd, E. Sala, C.-B. Schönlieb, and AIX-COVNET, "Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans," *Nature Machine Intelligence*, vol. 3, pp. 199–217, Mar 2021. 2.3, 6.1

[105] X. Ma, Y. Niu, L. Gu, Y. Wang, Y. Zhao, J. Bailey, and F. Lu, "Understanding adversarial attacks on deep learning based medical image analysis systems," *CoRR*, vol. abs/1907.10456, 2019. 2.3, 2.4.2, 3, 3.2, 3.2.3

[106] B. Neal, S. Mittal, A. Baratin, V. Tantia, M. Scicluna, S. Lacoste-Julien, and I. Mitliagkas, "A modern take on the bias-variance tradeoff in neural networks. arxiv 2018," *arXiv preprint arXiv:1810.08591*, 2018. 2.3.1

[107] Z. Yang, Y. Yu, C. You, J. Steinhardt, and Y. Ma, "Rethinking bias-variance trade-off for generalization of neural networks," in *International Conference on Machine Learning*, pp. 10767–10777, PMLR, 2020. 2.3.1

[108] M. Geiger, A. Jacot, S. Spigler, F. Gabriel, L. Sagun, S. d'Ascoli, G. Biroli, C. Hongler, and M. Wyart, "Scaling description of generalization with number of parameters in deep learning," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2020, p. 023401, feb 2020. 2.3.1

[109] Y. Dar, V. Muthukumar, and R. G. Baraniuk, "A farewell to the bias-variance tradeoff? an overview of the theory of overparameterized machine learning," 2021. 2.3.1

[110] M. Nauta, R. Walsh, A. Dubowski, and C. Seifert, "Uncovering and correcting shortcut learning in machine learning models for skin cancer diagnosis," *Diagnostics (Basel)*, vol. 12, p. 40, Dec. 2021. 2.3.1

[111] G. K. Dziugaite, A. Drouin, B. Neal, N. Rajkumar, E. Caballero, L. Wang, I. Mitliagkas, and D. M. Roy, "In search of robust measures of generalization," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, (Red Hook, NY, USA), Curran Associates Inc., 2020. 2.3.1, 4

[112] J. Bai, Y. Zeng, Y. Zhao, and F. Zhao, "Training a v1 like layer using gabor filters in convolutional neural networks," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2019. 2.3.2, 4.3

[113] R. Mehrotra, K. Namuduri, and N. Ranganathan, "Gabor filter-based edge detection," *Pattern Recognition*, vol. 25, no. 12, pp. 1479–1494, 1992. 2.3.2

[114] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, 2018. 2.3.2

[115] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving adversarial robustness via promoting ensemble diversity," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, vol. 97 of *Proceedings of Machine Learning Research*, pp. 4970–4979, PMLR, 2019. 2.3.2, 2.3.2, 4, 4.3

[116] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?," *Journal of Machine Learning Research*, vol. 15, no. 90, pp. 3133–3181, 2014. 2.3.2

[117] M. A. Arbib, *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 2nd ed., 2002. 2.3.2

[118] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and systems magazine*, vol. 6, no. 3, pp. 21–45, 2006. 2.3.2

[119] G. Perin, Ł. Chmielewski, and S. Picek, "Strength in numbers: Improving generalization with ensembles in machine learning-based profiled side-channel analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 337–364, 2020. 2.3.2

[120] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," *arXiv*, 2009. 2.3.2

[121] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2014. 2.4.1

[122] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2.4.1, 6.1.1

[123] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, 2019. 2.4.1

[124] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017. 2.4.1, 3.1.2

[125] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 39–57, IEEE Computer Society, 2017. 2.4.1, 3.1.2

[126] S. An, C. Xiao, W. F. Stewart, and J. Sun, "Longitudinal adversarial attack on electronic health records data," in *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pp. 2558–2564, ACM, 2019. 2.4.1, 3, 3.1.2

[127] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017. 2.4.2

[128] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *CoRR*, vol. abs/1703.00410, 2017. 2.4.2, 3.1.3, 3.2, 3.2.3

[129] P. Yang, J. Chen, C. Hsieh, J. Wang, and M. I. Jordan, "ML-LOO: detecting adversarial examples with feature attribution," *CoRR*, vol. abs/1906.03499, 2019. 2.4.2, 3.2.3

[130] N. Carlini, C. Liu, J. Kos, Úlfar Erlingsson, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *USENix*, 2019. 2.4.3, 6.2.1

[131] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," 2021. 2.4.3, 6.2, 6.2.1

[132] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," 2016. 2.4.3, 6.2.1, 6.2.2, 6.2.4

[133] E. Tabassi, K. J. Burns, M. Hadjimichael, A. D. Molina-Markham, and J. T. Sexton, "A taxonomy and terminology of adversarial machine learning," *NIST IR*, pp. 1–29, 2019. 2.4.3, 6.2.1

[134] Y. Long, L. Wang, D. Bu, V. Bindschaedler, X. Wang, H. Tang, C. A. Gunter, and K. Chen, "A pragmatic approach to membership inferences on machine learning models," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 521–534, IEEE, 2020. 2.4.3

[135] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 1964–1974, PMLR, 18–24 Jul 2021. 2.4.3

[136] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021. 2.4.4

[137] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. 2.4.4

[138] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629*, vol. 2, 2016. 2.4.4, 6.2.3, 6.2.4

[139] A. Hard, C. M. Kiddon, D. Ramage, F. Beaufays, H. Eichner, K. Rao, R. Mathews, and S. Augenstein, "Federated learning for mobile keyboard prediction," 2018. 2.4.4

[140] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020. 2.4.4, 6.2.1

[141] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*, pp. 739–753, IEEE, 2019. 2.4.4, 6.2.1, 6.2.2

[142] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 2512–2520, IEEE, 2019. 2.4.4, 6.2.1

[143] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016. 2.4.4, 2.4.4, 6.2, 6.2.4

[144] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*, pp. 265–284, Springer, 2006. 2.4.4

[145] B. Jayaraman, L. Wang, K. Knipmeyer, Q. Gu, and D. Evans, "Revisiting membership inference under realistic assumptions," 2020. 2.4.4, 6.2

[146] M. M. Kamani, S. Farhang, M. Mahdavi, and J. Z. Wang, "Targeted data-driven regularization for out-of-distribution generalization," in *KDD '20*, KDD '20, (New York, NY, USA), p. 882–891, Association for Computing Machinery, 2020. 2.4.4, 6.2

[147] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pp. 193–204, 2011. 2.4.4

[148] S. G. Finlayson, I. S. Kohane, and A. L. Beam, "Adversarial attacks against medical deep learning systems," *CoRR*, vol. abs/1804.05296, 2018. 3, 3.2.3, 3.3

[149] P. E. Kalb, "Health Care Fraud and Abuse," *JAMA*, vol. 282, pp. 1163–1168, 09 1999. 3

[150] E. J. Topol, “High-performance medicine: the convergence of human and artificial intelligence,” *Nature Medicine*, vol. 25, pp. 44–56, Jan 2019. 3, 6.1

[151] V. Prasad and S. Mailankody, “Research and Development Spending to Bring a Single Cancer Drug to Market and Revenues After Approval,” *JAMA Intern Med*, vol. 177, pp. 1569–1575, 11 2017. 3

[152] H. H. Pien, A. J. Fischman, J. H. Thrall, and A. G. Sorensen, “Using imaging biomarkers to accelerate drug development and clinical trials,” *Drug Discovery Today*, vol. 10, no. 4, pp. 259 – 266, 2005. 3

[153] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, “MIMIC-III, a freely accessible critical care database,” *Scientific Data*, vol. 3, p. 160035, May 2016. 3.1.1

[154] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 2261–2269, IEEE Computer Society, 2017. 3.1.1

[155] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2014. 3.1.3

[156] N. Sharma, V. Jain, and A. Mishra, “An analysis of convolutional neural networks for image classification,” *Procedia Computer Science*, vol. 132, pp. 377 – 384, 2018. International Conference on Computational Intelligence and Data Science. 3.2.2

[157] K. Xu, M. Lam, J. Pang, X. Gao, C. Band, P. Mathur, F. Papay, A. K. Khanna, J. B. Cywinski, K. Maheshwari, P. Xie, and E. P. Xing, “Multimodal machine learning for automated icd coding,” in *Proceedings of the 4th Machine Learning for Healthcare Conference* (F. Doshi-Velez, J. Fackler, K. Jung, D. Kale, R. Ranganath, B. Wallace, and J. Wiens, eds.), vol. 106 of *Proceedings of Machine Learning Research*, (Ann Arbor, Michigan), pp. 197–215, PMLR, 09–10 Aug 2019. 3.3

[158] D. S. Char, M. D. Abràmoff, and C. Feudtner, “Identifying ethical considerations for machine learning healthcare applications,” *The American Journal of Bioethics*, vol. 20, no. 11, pp. 7–17, 2020. 4

[159] J. J. Hatherley, “Limits of trust in medical ai,” *Journal of Medical Ethics*, vol. 46, no. 7, pp. 478–481, 2020. 4

[160] A. Chattopadhyay, P. Manupriya, A. Sarkar, and V. N. Balasubramanian, “Neural network attributions: A causal perspective,” in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 981–990, PMLR, 09–15 Jun 2019. 4

[161] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, 1990. 4

[162] G. Plumb, D. Molitor, and A. Talwalkar, "Model agnostic supervised local explanations," *arXiv preprint arXiv:1807.02910*, 2018. 4.2

[163] E. Parzen, "On Estimation of a Probability Density Function and Mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065 – 1076, 1962. 4.2.2

[164] J. W. Pratt and J. D. Gibbons, *Kolmogorov-Smirnov Two-Sample Tests*, pp. 318–344. New York, NY: Springer New York, 1981. 4.2.2

[165] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. 4.3

[166] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "SVCCA: singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 6076–6085, 2017. 4.4

[167] W. van Drongelen, "4 - signal averaging," in *Signal Processing for Neuroscientists* (W. van Drongelen, ed.), pp. 55–70, Burlington: Academic Press, 2007. 5.1.1

[168] N.-B. Heidenreich, A. Schindler, and S. Sperlich, "Bandwidth selection for kernel density estimation: a review of fully automatic selectors," *AStA Advances in Statistical Analysis*, vol. 97, pp. 403–433, Oct 2013. 5.1.1

[169] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015. 5.1.1

[170] D. M. Bashtannyk and R. J. Hyndman, "Bandwidth selection for kernel conditional density estimation," *Computational Statistics & Data Analysis*, vol. 36, no. 3, pp. 279–298, 2001. 5.1.1, 5.2.3

[171] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018. 5.1.2

[172] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016. 5.1.2, 6.1.1

[173] M. Watson, B. A. S. Hasan, and N. A. Moubayed, "Agree to disagree: When deep learning models with identical architectures produce distinct explanations," *CoRR*, vol. abs/2105.06791, 2021. 5.1.2, 5.1.4, 5.2.3

[174] K. Schulz, L. Sixt, F. Tombari, and T. Landgraf, “Restricting the flow: Information bottlenecks for attribution,” in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020. 5.1.3

[175] C. D. Manning and H. Schutze, *Foundations of statistical natural language processing.* MIT, 2008. 5.1.4

[176] X. Wang, D. Kondratyuk, K. M. Kitani, Y. Movshovitz-Attias, and E. Eban, “Multiple networks are more efficient than one: Fast and accurate models via ensembles and cascades,” *CoRR*, vol. abs/2012.01988, 2020. 5.1.7

[177] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, “Deep ehr: A survey of recent advances in deep learning techniques for electronic health record (ehr) analysis,” *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1589–1604, 2018. 5.2.1

[178] L. Koumakis, “Deep learning models in genomics; are we there yet?,” *Computational and Structural Biotechnology Journal*, vol. 18, pp. 1466–1473, 2020. 5.2.1

[179] A. F. M. Agarap, “On breast cancer detection: An application of machine learning algorithms on the wisconsin diagnostic dataset,” in *Proceedings of the 2nd International Conference on Machine Learning and Soft Computing*, ICMLSC ’18, (New York, NY, USA), p. 5–9, Association for Computing Machinery, 2018. 5.2.1

[180] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. Celi, and R. Mark, “Mimic-iv (version 1.0),” 2020. 5.2.1

[181] J. Deasy, P. Liò, and A. Ercole, “Dynamic survival prediction in intensive care units from heterogeneous time series without the need for variable selection or curation,” *Scientific Reports*, vol. 10, p. 22129, Dec 2020. 5.2.1

[182] J. Deasy, P. Liò, and A. Ercole, “flexible-ehr implementation.” `https://github.com/jacobdeasy/flexible-ehr`. Accessed: 2022-10-04. 5.2.1

[183] J. Chen and V. Storchan, “Seven challenges for harmonizing explainability requirements,” *CoRR*, vol. abs/2108.05390, 2021. 5.3

[184] K. Gade, S. C. Geyik, K. Kenthapadi, V. Mithal, and A. Taly, “Explainable ai in industry,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD ’19, (New York, NY, USA), p. 3203–3204, Association for Computing Machinery, 2019. 5.3

[185] S. Kundu, “Ai in medicine must be explainable,” *Nature Medicine*, vol. 27, pp. 1328–1328, Aug 2021. 5.3

[186] M. You, A. Yuan, D. He, and X. Li, “Unsupervised feature selection via neural networks and self-expression with adaptive graph constraint,” *Pattern Recognition*, vol. 135, p. 109173, 2023. 5.3

[187] A. Vellido, V. Ribas, C. Morales, A. Ruiz Sanmartín, and J. C. Ruiz Rodríguez, “Machine learning in critical care: state-of-the-art and a sepsis case study,” *BioMedical Engineering OnLine*, vol. 17, p. 135, Nov 2018. 6.1

[188] D. S. Char, M. D. Abràmoff, and C. Feudtner, “Identifying Ethical Considerations for Machine Learning Healthcare Applications,” *Am J Bioeth*, vol. 20, pp. 7–17, 11 2020. 6.1, 6.2

[189] J. J. Hatherley, “Limits of trust in medical ai,” *Journal of Medical Ethics*, vol. 46, no. 7, pp. 478–481, 2020. 6.1, 6.2

[190] S. Sindhwani, G. Minissale, G. Weber, C. Lutteroth, A. Lambert, N. Curtis, and E. Broadbent, “A multidisciplinary study of eye tracking technology for visual intelligence,” *Education Sciences*, vol. 10, no. 8, 2020. 6.1

[191] B. Butcher, V. S. Huang, C. Robinson, J. Reffin, S. K. Sgaier, G. Charles, and N. Quadrianto, “Causal datasheet for datasets: An evaluation guide for real-world data analysis and data collection design using bayesian networks,” *Frontiers in Artificial Intelligence*, vol. 4, 2021. 6.1.1

[192] S. Waite, A. Grigorian, R. G. Alexander, S. L. Macknik, M. Carrasco, D. J. Heeger, and S. Martinez-Conde, “Analysis of perceptual expertise in radiology – current knowledge and a new perspective,” *Frontiers in Human Neuroscience*, vol. 13, 2019. 6.1.1

[193] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*, pp. 6105–6114, PMLR, 2019. 6.1.3

[194] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, “What do different evaluation metrics tell us about saliency models?,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 740–757, 2018. 6.1.3, 6.1.4

[195] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, “A survey on ensemble learning,” *Frontiers of Computer Science*, vol. 14, no. 2, pp. 241–258, 2020. 6.1.4

[196] S. Singla, S. Wallace, S. Triantafillou, and K. Batmanghelich, “Using causal analysis for conceptual deep learning explanation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 519–528, Springer, 2021. 6.1.5

[197] Y. Goyal, A. Feder, U. Shalit, and B. Kim, “Explaining classifiers with causal concept effect (cace),” *arXiv preprint arXiv:1907.07165*, 2019. 6.1.5

[198] X. Gu and A. Easwaran, “Towards safe machine learning for cps: Infer uncertainty from training data,” in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, ICCPS ’19, (New York, NY, USA), p. 249–258, Association for Computing Machinery, 2019. 6.2

[199] S. Truex, L. Liu, M. Gursoy, L. Yu, and W. Wei, "Demystifying membership inference attacks in machine learning as a service," *IEEE Transactions on Services Computing*, vol. 14, pp. 2073–2089, nov 2021. 6.2, 6.2.4

[200] T. Moberly, "Should we be worried about the nhs selling patient data?," *BMJ*, vol. 368, 2020. 6.2

[201] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014. 6.2

[202] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," 2020. 6.2

[203] P. Liu, X. Xu, and W. Wang, "Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives," *Cybersecurity*, vol. 5, p. 4, Feb 2022. 6.2

[204] O. Zari, C. Xu, and G. Neglia, "Efficient passive membership inference attack in federated learning," 2021. 6.2

[205] S. Sagawa, A. Raghunathan, P. W. Koh, and P. Liang, "An investigation of why overparameterization exacerbates spurious correlations," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 8346–8356, PMLR, 13–18 Jul 2020. 6.2

[206] M. A. Rahman, T. Rahman, R. Laganière, N. Mohammed, and Y. Wang, "Membership inference attack against differentially private deep learning model.," *Trans. Data Priv.*, vol. 11, no. 1, pp. 61–79, 2018. 6.2

[207] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021. 6.2.2

[208] Y. LeCun and C. Cortes, "MNIST handwritten digit database," *Online*, 2010. 6.2.4, 6.2.4

[209] T. D. of State Health Services, "Hospital discharge data public use data file," 2015. 6.2.4, 6.2.4

[210] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao, G. Cormode, and I. Mironov, "Opacus: User-friendly differential privacy library in PyTorch," *arXiv preprint arXiv:2109.12298*, 2021. 6.2.4

[211] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914, IEEE, 2022. 6.2.4

APPENDIX A

Deep Explanation Ensemble Hyperparameter Results

The experiments carried out in Chapter 5 required numerous models to be trained, each with different training hyperparameter setups. While all of the results in the tables below have been summarised in Chapter 5.2, for completeness and reproducibility I have included in the individual results from each model trained below.

Dataset (Task)	**Random Seed**	**Shuffle**	**Performance Metric**
Diabetes (Classification)	18829	FALSE	82.33
Diabetes (Classification)	20526	FALSE	82.93
Diabetes (Classification)	44392	FALSE	83.02
Diabetes (Classification)	7021	FALSE	82.71
Diabetes (Classification)	93864	FALSE	81.89
Diabetes (Classification)	17884	TRUE	83.08
Diabetes (Classification)	26549	TRUE	83.58
Diabetes (Classification)	42609	TRUE	83.53
Diabetes (Classification)	52732	TRUE	83.42
Diabetes (Classification)	53291	TRUE	83.33
Diabetes (Classification)	58075	TRUE	83.43
Diabetes (Classification)	65452	TRUE	83.27
Diabetes (Classification)	66701	TRUE	**83.67**
Diabetes (Classification)	7495	TRUE	83.33
Diabetes (Classification)	81189	TRUE	83.43
Diabetes (Classification)	9937	TRUE	83.12
Diabetes (Regression)	1	FALSE	0.578
Diabetes (Regression)	17351	FALSE	0.601
Diabetes (Regression)	35397	FALSE	0.579
Diabetes (Regression)	39419	FALSE	0.595
Diabetes (Regression)	42290	FALSE	0.601
Diabetes (Regression)	51363	FALSE	**0.602**
Diabetes (Regression)	54867	TRUE	0.582
Diabetes (Regression)	64500	TRUE	0.593
Diabetes (Regression)	66703	TRUE	0.569
Diabetes (Regression)	83349	TRUE	0.560
Diabetes (Regression)	95279	TRUE	0.584
Diabetes (Regression)	96047	TRUE	0.586

Table A.1: Performance and hyperparameters of the baseline MLPs trained on the KAIMRC dataset. The performance metric for the classification task is accuracy and adjusted R^2 for the regression task.

Dataset	**Seed**	**Shuffle**	**AUROC**
BCW	22323	FALSE	93.86
BCW	25197	FALSE	93.86
BCW	29698	FALSE	93.86
BCW	30135	FALSE	89.47
BCW	30938	FALSE	93.86
BCW	39325	FALSE	92.98
BCW	41292	FALSE	90.35
BCW	53050	FALSE	92.11
BCW	61455	FALSE	89.47
BCW	78827	FALSE	92.11
BCW	81960	FALSE	92.11
BCW	19191	TRUE	88.60
BCW	23087	TRUE	91.23
BCW	24735	TRUE	89.47
BCW	43842	TRUE	**95.61**
BCW	47506	TRUE	86.42
BCW	57075	TRUE	92.11
BCW	62605	TRUE	90.25
BCW	63612	TRUE	93.86
BCW	67425	TRUE	90.35
BCW	92747	TRUE	93.86
BCW	97704	TRUE	94.74

Table A.2: Accuracy and hyperparameters for the baseline MLPs on the Breast Cancer Wisconsin (BCW) dataset.

Dataset (Task)	Seed	Shuffle	Accuracy	F1 Score
Codon Usage (DNA)	22402	FALSE	99.31	99.30
Codon Usage (DNA)	24402	FALSE	99.19	99.19
Codon Usage (DNA)	39126	FALSE	**99.38**	**99.39**
Codon Usage (DNA)	44437	FALSE	99.31	99.31
Codon Usage (DNA)	55833	FALSE	99.15	99.16
Codon Usage (DNA)	58236	TRUE	99.19	99.19
Codon Usage (DNA)	6160	TRUE	99.31	99.31
Codon Usage (DNA)	64119	TRUE	99.11	99.12
Codon Usage (DNA)	64390	TRUE	99.07	99.09
Codon Usage (DNA)	71650	TRUE	99.15	99.15
Codon Usage (Kingdom)	17094	FALSE	85.62	0.8504
Codon Usage (Kingdom)	19709	FALSE	**86.65**	**0.8638**
Codon Usage (Kingdom)	29559	FALSE	86.31	0.8616
Codon Usage (Kingdom)	3440	FALSE	85.67	0.8488
Codon Usage (Kingdom)	39406	FALSE	82.15	0.8103
Codon Usage (Kingdom)	51088	TRUE	83.30	0.8231
Codon Usage (Kingdom)	63023	TRUE	85.84	0.8506
Codon Usage (Kingdom)	74147	TRUE	85.79	0.8547
Codon Usage (Kingdom)	84013	TRUE	85.75	0.8530
Codon Usage (Kingdom)	92214	TRUE	86.22	0.8580

Table A.3: Accuracy, F1 score and hyperparameters of the baseline MLPs trained on the Codon Usage dataset on both the kingdom and DNA multi-class classification tasks.

Dataset	**Seed**	**Shuffle**	**AUROC**
MIMIC-IV	7321	FALSE	0.7642
MIMIC-IV	1163	FALSE	0.7247
MIMIC-IV	3193	FALSE	0.7813
MIMIC-IV	7429	FALSE	0.7563
MIMIC-IV	8433	FALSE	0.7916
MIMIC-IV	22321	TRUE	**0.8166**
MIMIC-IV	32283	TRUE	0.7748
MIMIC-IV	69432	TRUE	0.7794
MIMIC-IV	77973	TRUE	0.8071
MIMIC-IV	82342	TRUE	0.8089

Table A.4: Model performance and hyperparameters for the LSTM-based baseline models trained on MIMIC-IV mortality prediction.

Dataset (Task)	**Seed**	**Shuffle**	**Accuracy**
BCW	1621	FALSE	72.42
BCW	3063	FALSE	84.62
BCW	3309	FALSE	87.25
BCW	7159	FALSE	78.57
BCW	8163	FALSE	86.59
BCW	2602	TRUE	72.64
BCW	3233	TRUE	**88.79**
BCW	6922	TRUE	88.24
BCW	7797	TRUE	76.15
BCW	8332	TRUE	72.86

Table A.5: Model performance of normal ensemble models, of 10 sub-models each, on the BCW dataset.

Dataset (Task)	**Seed**	**Shuffle**	**Accuracy**
KAIMRC (Classification)	1621	FALSE	83.28
KAIMRC (Classification)	3063	FALSE	83.15
KAIMRC (Classification)	3309	FALSE	83.07
KAIMRC (Classification)	7159	FALSE	**83.64**
KAIMRC (Classification)	8163	FALSE	83.08
KAIMRC (Classification)	2602	TRUE	83.15
KAIMRC (Classification)	3233	TRUE	83.08
KAIMRC (Classification)	6922	TRUE	83.28
KAIMRC (Classification)	7797	TRUE	83.28
KAIMRC (Classification)	8332	TRUE	83.27
KAIMRC (Regression)	1621	FALSE	0.52
KAIMRC (Regression)	3063	FALSE	0.49
KAIMRC (Regression)	3309	FALSE	**0.54**
KAIMRC (Regression)	7159	FALSE	0.51
KAIMRC (Regression)	8163	FALSE	0.51
KAIMRC (Regression)	2602	TRUE	0.51
KAIMRC (Regression)	3233	TRUE	0.52
KAIMRC (Regression)	6922	TRUE	0.50
KAIMRC (Regression)	7797	TRUE	0.53
KAIMRC (Regression)	8332	TRUE	0.51

Table A.6: Model performance of normal ensemble models, of 10 sub-models each, on the KAIMRC dataset for both the classification and regression tasks.

Dataset (Task)	Seed	Shuffle	Accuracy
Codon Usage (DNA)	1621	FALSE	99.07
Codon Usage (DNA)	3063	FALSE	99.42
Codon Usage (DNA)	3309	FALSE	99.23
Codon Usage (DNA)	7159	FALSE	98.84
Codon Usage (DNA)	8163	FALSE	99.19
Codon Usage (DNA)	2602	TRUE	**99.46**
Codon Usage (DNA)	3233	TRUE	99.00
Codon Usage (DNA)	6922	TRUE	99.04
Codon Usage (DNA)	7797	TRUE	98.88
Codon Usage (DNA)	8332	TRUE	99.23
Codon Usage (Kingdom)	1621	FALSE	90.39
Codon Usage (Kingdom)	3063	FALSE	87.23
Codon Usage (Kingdom)	3309	FALSE	90.82
Codon Usage (Kingdom)	7159	FALSE	91.24
Codon Usage (Kingdom)	8163	FALSE	90.99
Codon Usage (Kingdom)	2602	TRUE	91.20
Codon Usage (Kingdom)	3233	TRUE	91.29
Codon Usage (Kingdom)	6922	TRUE	88.63
Codon Usage (Kingdom)	7797	TRUE	89.96
Codon Usage (Kingdom)	8332	TRUE	**91.76**

Table A.7: Model performance of normal ensemble models, of 10 sub-models each, on the Codon Usage dataset for both the DNA and kingdom multi-class classification tasks.

Dataset (Task)	Seed	Shuffle	Accuracy
BCW	15671	FALSE	80.59
BCW	19353	FALSE	89.47
BCW	26628	FALSE	**90.53**
BCW	45386	FALSE	88.90
BCW	56945	FALSE	89.63
BCW	58245	TRUE	89.24
BCW	59288	TRUE	86.84
BCW	92627	TRUE	83.33
BCW	99734	TRUE	87.72

Table A.8: Model performance of our explanation ensemble models, each of 10 sub-models each, on the BCW dataset.

Dataset (Task)	**Seed**	**Shuffle**	**Accuracy**
KAIMRC (Classification)	3294	FALSE	81.86
KAIMRC (Classification)	32259	FALSE	81.82
KAIMRC (Classification)	45556	FALSE	82.37
KAIMRC (Classification)	56208	FALSE	82.64
KAIMRC (Classification)	61300	TRUE	81.61
KAIMRC (Classification)	78867	TRUE	**83.27**
KAIMRC (Classification)	80154	TRUE	82.28
KAIMRC (Classification)	83464	TRUE	82.53
KAIMRC (Regression)	1540	FALSE	0.5493
KAIMRC (Regression)	4881	FALSE	0.5152
KAIMRC (Regression)	33097	FALSE	0.5514
KAIMRC (Regression)	43716	FALSE	0.5529
KAIMRC (Regression)	45016	TRUE	0.5254
KAIMRC (Regression)	62778	TRUE	0.5572
KAIMRC (Regression)	72795	TRUE	0.5561
KAIMRC (Regression)	91774	TRUE	0.5076
KAIMRC (Regression)	97880	TRUE	**0.5578**

Table A.9: Model performance of our explanation ensemble models, each of 10 sub-models each, on the KAIMRC dataset for both the classification and regression tasks.

Dataset (Task)	**Seed**	**Shuffle**	**Accuracy**
Codon Usage (DNA)	7009	FALSE	**98.51**
Codon Usage (DNA)	20624	FALSE	98.26
Codon Usage (DNA)	37971	FALSE	98.49
Codon Usage (DNA)	41030	FALSE	97.51
Codon Usage (DNA)	43356	FALSE	97.39
Codon Usage (DNA)	64863	TRUE	97.06
Codon Usage (DNA)	86245	TRUE	97.64
Codon Usage (DNA)	94742	TRUE	98.37
Codon Usage (DNA)	97499	TRUE	97.54
Codon Usage (Kingdom)	2107	FALSE	89.96
Codon Usage (Kingdom)	46598	FALSE	87.12
Codon Usage (Kingdom)	47329	FALSE	89.87
Codon Usage (Kingdom)	49806	FALSE	89.27
Codon Usage (Kingdom)	49951	FALSE	89.66
Codon Usage (Kingdom)	54426	TRUE	87.64
Codon Usage (Kingdom)	57058	TRUE	88.15
Codon Usage (Kingdom)	64179	TRUE	**90.26**
Codon Usage (Kingdom)	73122	TRUE	88.80
Codon Usage (Kingdom)	87606	TRUE	89.48

Table A.10: Model performance of our explanation ensemble models, with of 10 sub-models each, on the Codon Usage dataset on both the DNA and Kingdom multi-class classification tasks.

Dataset (Task)	**Seed**	**Shuffle**	**AUROC**
MIMIC-IV	1163	FALSE	0.7733
MIMIC-IV	22321	FALSE	0.7734
MIMIC-IV	3193	FALSE	0.7734
MIMIC-IV	32283	FALSE	**0.7735**
MIMIC-IV	69432	FALSE	0.7733
MIMIC-IV	7321	TRUE	0.7730
MIMIC-IV	7429	TRUE	0.7732
MIMIC-IV	77973	TRUE	0.7733
MIMIC-IV	82342	TRUE	0.7732
MIMIC-IV	8433	TRUE	0.7733

Table A.11: Model performance (accuracy under the receiver operating characteristic curve, AUROC) of our explanation ensemble models, of 10 sub-models each, on the MIMIC-IV mortality prediction task.

Dataset	Seed	Shuffle	Infidelity	Sensitivity
BCW	29698	FALSE	0.000391	0.581461
BCW	25197	FALSE	0.000634	0.512061
BCW	41292	FALSE	0.000527	0.421115
BCW	22323	FALSE	0.000434	0.517406
BCW	19191	TRUE	0.000445	0.513631
BCW	24735	TRUE	0.000460	0.563020
BCW	47506	TRUE	0.000774	0.716147
BCW	57075	TRUE	0.001268	0.617298
Codon Usage (DNA)	71650	TRUE	0.008231	0.678376
Codon Usage (DNA)	58236	TRUE	0.001719	1.823636
Codon Usage (DNA)	64119	TRUE	0.003357	1.089292
Codon Usage (DNA)	64390	TRUE	0.004377	1.255262
Codon Usage (DNA)	22402	FALSE	0.001696	1.610876
Codon Usage (DNA)	6160	TRUE	0.002804	1.356076
Codon Usage (DNA)	39126	FALSE	0.000410	3.270923
Codon Usage (DNA)	55833	FALSE	0.005762	0.776691
Codon Usage (DNA)	44437	FALSE	0.005224	0.997720
KAIMRC (Classification)	1621	FALSE	0.001360	0.591762
KAIMRC (Classification)	3063	FALSE	0.004201	0.453467
KAIMRC (Classification)	3309	FALSE	0.001295	0.509114
KAIMRC (Classification)	7159	FALSE	0.000000	0.000000
KAIMRC (Classification)	8163	FALSE	0.003186	0.427264
KAIMRC (Classification)	2602	TRUE	0.003995	0.397451
KAIMRC (Classification)	3233	TRUE	0.001077	0.542822
KAIMRC (Classification)	6922	TRUE	0.000547	0.000000
KAIMRC (Classification)	7797	TRUE	0.003540	0.597366
KAIMRC (Classification)	8332	TRUE	0.000997	0.527902
Codon Usage (Kingdom)	1621	FALSE	0.002073	0.700954
Codon Usage (Kingdom)	3063	FALSE	0.006445	0.968512
Codon Usage (Kingdom)	7159	FALSE	0.009926	0.754319
Codon Usage (Kingdom)	8163	FALSE	0.004835	0.688471
Codon Usage (Kingdom)	2602	TRUE	0.002424	0.673015
Codon Usage (Kingdom)	3233	TRUE	0.039926	0.573023
Codon Usage (Kingdom)	6922	TRUE	0.012041	0.718585
Codon Usage (Kingdom)	7797	TRUE	0.003896	1.015589

Table A.12: Explanation infidelity and explanation sensitivity max of the baseline model architectures on all classification datasets.

Dataset	**Seed**	**Shuffle**	**Infidelity**	**Sensitivity**
BCW	1621	FALSE	0.000039834	1.114579
BCW	3063	FALSE	0.000075637	0.835891
BCW	3309	FALSE	0.000071457	1.704849
BCW	7159	FALSE	0.000081874	**0.633781**
BCW	8163	FALSE	0.000085646	0.998697
BCW	2602	TRUE	0.000022444	**0.633781**
BCW	3233	TRUE	0.000422726	0.995961
BCW	6922	TRUE	0.000442698	0.738602
BCW	7797	TRUE	**0.000010963**	1.175704
BCW	8332	TRUE	0.001441245	0.894858
Codon Usage (DNA)	1621	FALSE	0.00062983	**0.656578**
Codon Usage (DNA)	3063	FALSE	**0.00000066**	1.467080
Codon Usage (DNA)	3309	FALSE	0.00000144	0.739519
Codon Usage (DNA)	7159	FALSE	0.00006802	1.894022
Codon Usage (DNA)	8163	FALSE	0.00000257	0.780092
Codon Usage (DNA)	2602	TRUE	0.00001721	1.000341
Codon Usage (DNA)	3233	TRUE	0.00003930	1.979980
Codon Usage (DNA)	6922	TRUE	0.00000357	1.071276
Codon Usage (DNA)	7797	TRUE	0.00001502	0.761258
Codon Usage (DNA)	8332	TRUE	0.00144648	2.045398
Codon Usage (Kingdom)	1621	FALSE	0.00002163	0.930669
Codon Usage (Kingdom)	3063	FALSE	0.00003154	1.205581
Codon Usage (Kingdom)	3309	FALSE	0.00002493	0.951259
Codon Usage (Kingdom)	7159	FALSE	0.00003060	0.981180
Codon Usage (Kingdom)	8163	FALSE	**0.00001940**	1.083869
Codon Usage (Kingdom)	2602	TRUE	0.00003436	**0.789647**
Codon Usage (Kingdom)	3233	TRUE	0.00004075	1.104336
Codon Usage (Kingdom)	6922	TRUE	0.00003798	1.003424
Codon Usage (Kingdom)	7797	TRUE	0.00003327	1.121576
Codon Usage (Kingdom)	8332	TRUE	0.00002583	0.981720
KAIMRC (Classification)	1621	FALSE	0.00045833	0.483609
KAIMRC (Classification)	3063	FALSE	0.00090531	**0.464702**
KAIMRC (Classification)	3309	FALSE	**0.00024935**	0.484267
KAIMRC (Classification)	7159	FALSE	0.00043942	0.538328
KAIMRC (Classification)	8163	FALSE	0.00220850	0.471807
KAIMRC (Classification)	2602	TRUE	0.00334645	0.474118
KAIMRC (Classification)	3233	TRUE	0.00046485	0.563551
KAIMRC (Classification)	6922	TRUE	0.00100100	0.472233
KAIMRC (Classification)	7797	TRUE	0.00141688	0.516746
KAIMRC (Classification)	8332	TRUE	0.00079866	0.581355

Table A.13: Explanation infidelity and explanation sensitivity of each individual explanation ensemble (of size 10) tested across each classification dataset.

APPENDIX B

Medical Imaging Applications Hyperparameter Results

As part of the extensive experimentation carried out for Chapter 6.1, the methods were run across models with many hyperparameter setups. While these results are summarised in Chapter 6.1.4, for completeness and reproducibility, the exact results for each hyperparameter results are reported below.

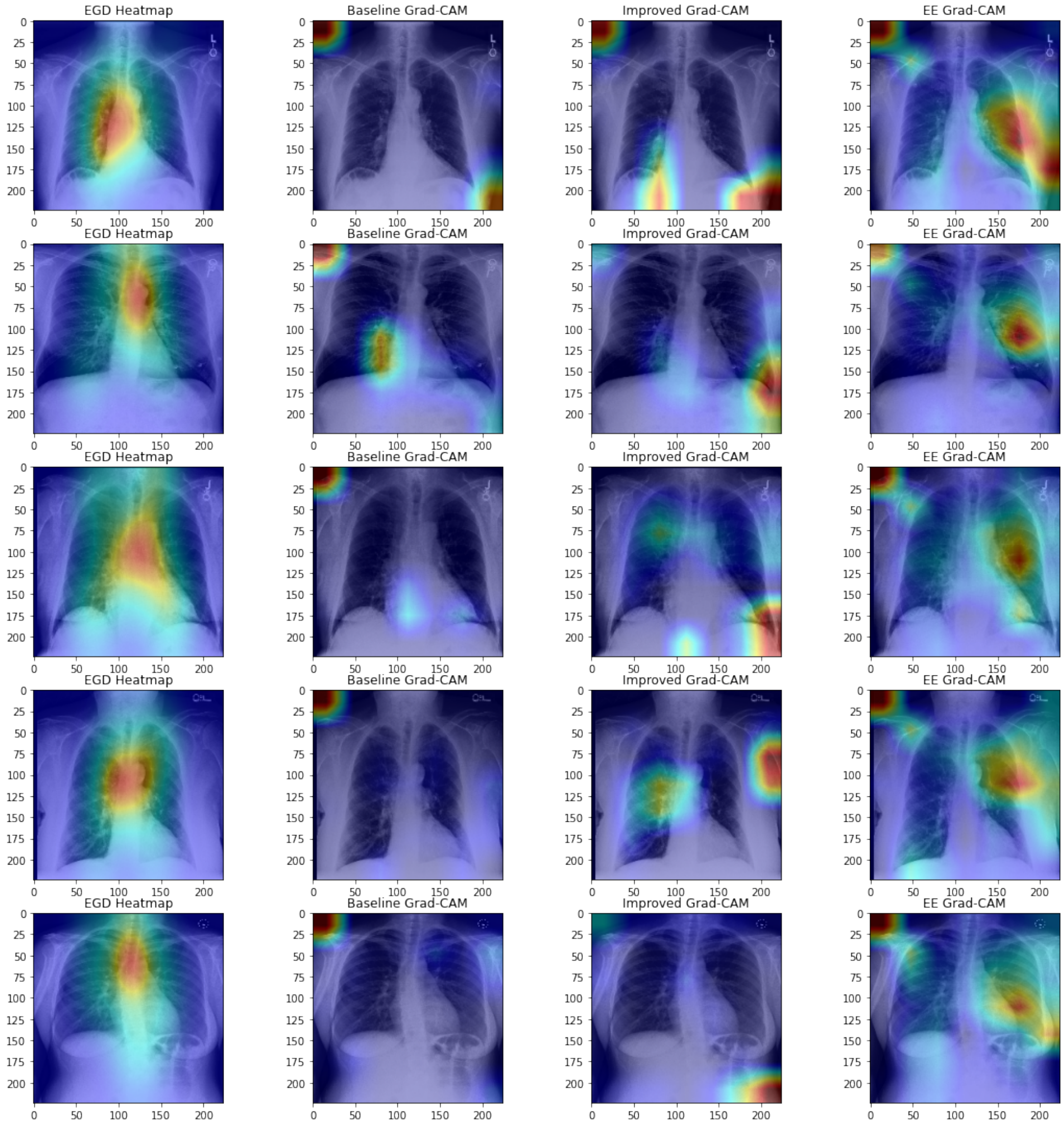


Figure B.1: 5 random samples from the MIMIC-CXR-EGD dataset overlaid with the eye gaze data heatmaps and GradCAM explanations from the baseline, improved UNet and explanation ensemble models.

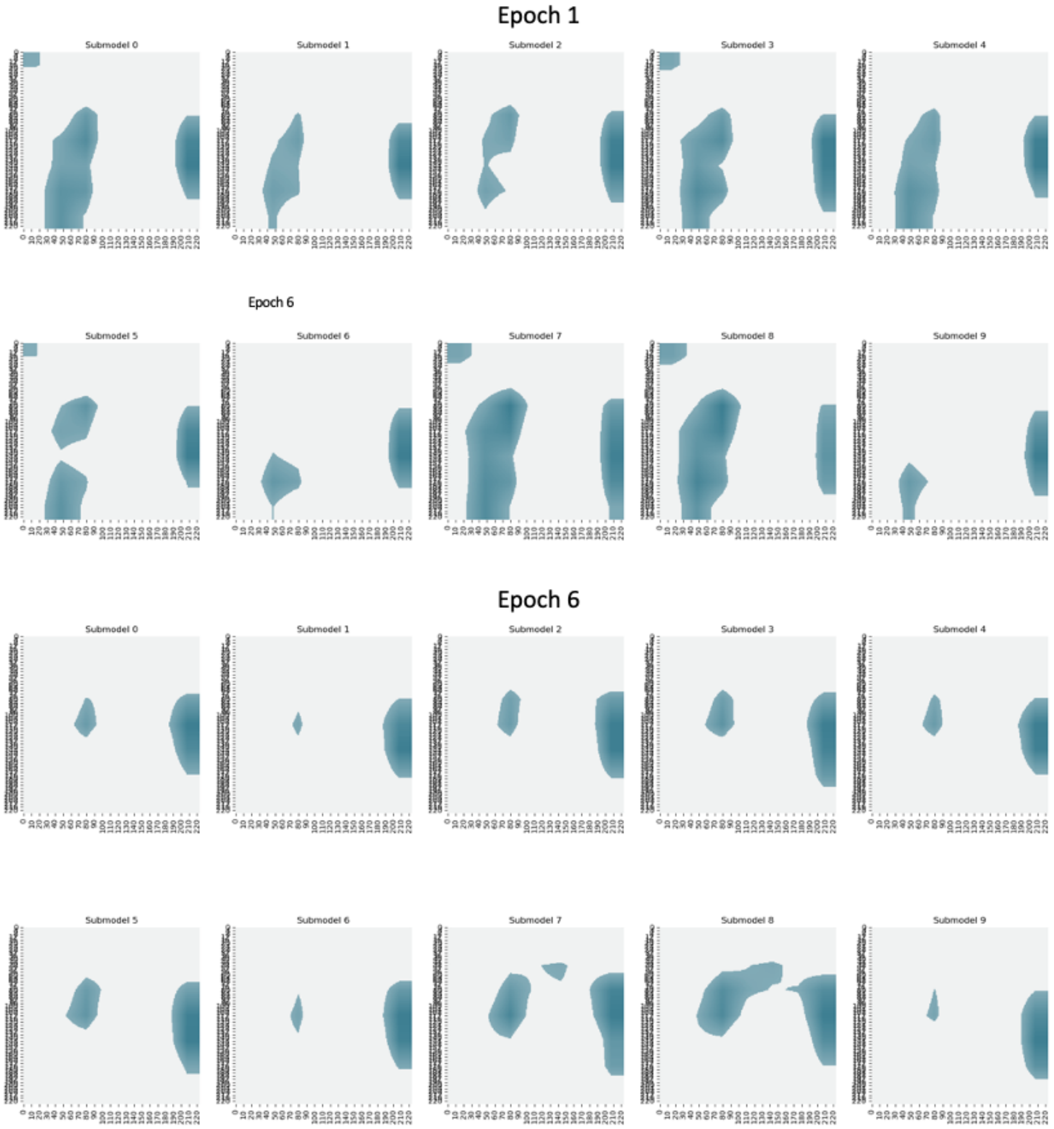


Figure B.2: Average GradCAM values (across the validation split) of each sub-model of our Explanation Ensemble model, as training progresses over epochs 1 and 6. To aid with visualisation, only the most important 50% of pixels are shown. Sub-models start training with vastly different learned features, and as training progresses our training procedure encourages the sub-models to learn similar features. Joint with Figure B.3, this is a larger version of Figure 6.5.

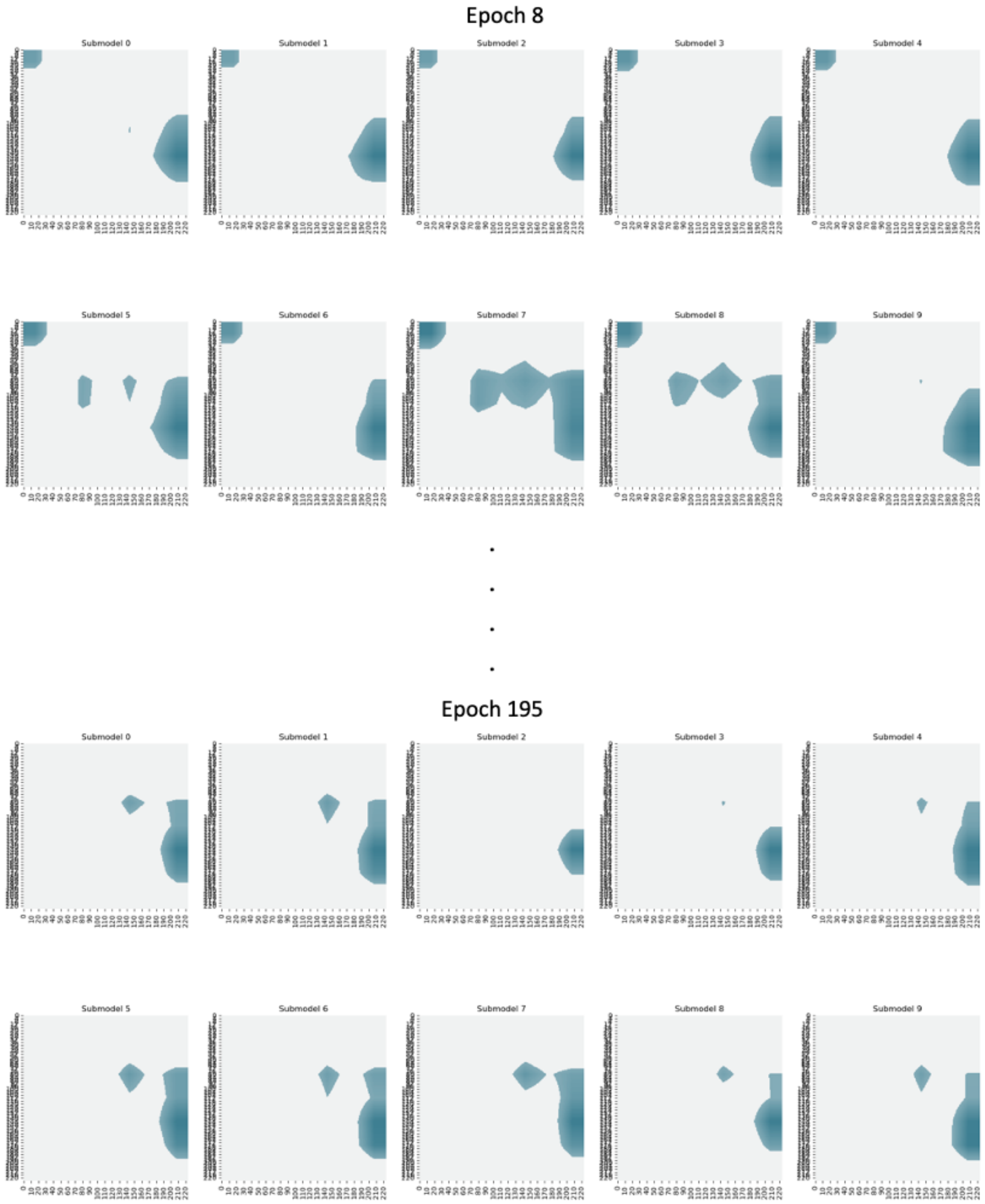


Figure B.3: Average GradCAM values (across the validation split) of each sub-model of our Explanation Ensemble model, as training progresses over epochs 8 and 195. To aid with visualisation, only the most important 50% of pixels are shown. Sub-models start training with vastly different learned features, and as training progresses our training procedure encourages the sub-models to learn similar features. Joint with Figure B.2, this is a larger version of Figure 6.5.

Table B.1: Table reporting model accuracy and mean KLD/NSS similarity between GradCAM explanations and radiologist eye-gaze data.

Model Architecture	Seed	Accuracy	KLD	NSS
Baseline	1735	72.17	10.744	-0.497
	2948	74.34	6.114	0.174
	4235	72.61	8.902	-0.331
	4582	69.00	9.555	-0.272
	4678	74.00	13.349	-0.145
	5682	73.81	4.288	0.183
	7624	75.55	14.404	-0.858
	7626	73.69	14.064	-0.113
	9374	69.85	10.289	-0.078
	9576	73.09	7.197	-0.173
Improved UNet (current SOTA)	1735	75.29	5.363	0.469
	2948	70.58	13.031	-0.032
	4235	75.57	5.429	0.096
	4582	76.51	9.937	-0.324
	4678	75.77	7.266	-0.169
	5682	69.25	12.195	0.336
	7624	74.93	11.257	0.405
	7626	75.85	4.992	0.025
	9374	74.59	4.672	-0.777
	9576	72.97	9.265	-0.386
Normal Ensemble	1735	74.59	1.221	-0.070
	2948	78.90	2.287	0.359
	4235	78.90	3.285	0.360
	4582	74.11	2.378	0.324
	4678	**79.86**	3.884	-0.165
	5682	78.42	4.944	-0.653
	7624	76.51	2.117	0.269
	7626	75.12	1.2688	0.290
	9374	73.64	1.099	-0.249
	9576	76.99	1.740	1.111
Expl. Ensemble (Ours)	1735	74.11	1.340	0.640
	2948	76.51	1.025	0.577
	4235	76.99	**0.786**	**1.237**
	4582	76.51	0.967	1.157
	4678	75.60	1.808	0.758
	5682	73.16	1.388	0.666
	7624	73.16	1.263	1.170
	7626	77.46	1.267	0.566
	9374	78.94	0.820	1.176
	9576	76.03	0.908	1.011

Seed	Ensemble Size	KLD	NSS	Seed	Ensemble Size	KLD	NSS
1467	5	1.983	-2.515	1467	8	1.485	0.837
3942	5	2.785	-1.013	3942	8	1.701	-1.16
4635	5	1.936	0.279	4635	8	2.175	-1.145
8304	5	2.694	-2.151	8304	8	2.321	-1.163
5305	5	2.292	-2.302	5305	8	1.266	-0.842
5439	5	1.833	-1.489	5439	8	1.471	0.831
6395	5	2.302	-1.853	6395	8	6.55	-1.491
7098	5	1.811	-1.586	7098	8	2.503	0.556
2089	5	2.472	-2.995	2089	8	1.25	-1.045
3104	5	2.441	1.021	3104	8	1.559	-0.954
1467	7	2.193	0.298	1467	10	0.786	1.237
3942	7	1.991	-0.081	3942	10	0.967	1.157
4635	7	2.372	0.23	4635	10	1.808	0.758
8304	7	1.975	-2.031	8304	10	1.388	0.666
5305	7	2.382	0.023	5305	10	1.263	1.170
5439	7	2.476	3.741	5439	10	1.267	0.566
6395	7	1.313	-1.64	6395	10	0.820	1.176
7098	7	2.004	2.298	7098	10	0.908	1.011
2089	7	1.608	-0.069	2089	10	1.340	0.640
3104	7	1.541	1.22	3104	10	1.025	0.577

Table B.2: Table reporting mean KLD/NSS similarity between GradCAM explanations and radiologist eye-gaze data of our Explanation Ensemble architecture with differing numbers of sub-models (i.e. ensemble size).

APPENDIX C

Federated Learning Hyperparameter Results

As part of the extensive experimentation carried out for Chapter 6.2, the methods were run across models with many hyperparameter setups. While these results are summarised in Chapter 6.2.4, for completeness and reproducibility, the exact results for each hyperparameter results are reported below.

Model	Seed	Attack Type	Advantage
dee	6	lr	0.0
dee	7	lr	0.0
dee	8	lr	0.0
dee	1	lr	0.0
dee	6	mlp	0.0
dee	7	mlp	0.0
dee	8	mlp	0.0
dee	1	mlp	0.0
dee	6	svm	0.0
dee	7	svm	0.0
dee	8	svm	0.0
dee	1	svm	0.0
mlp	1	mlp	0.1
mlp	11	mlp	0.1
mlp	12	mlp	0.1
mlp	13	mlp	0.1
mlp	14	mlp	0.1
mlp	1	lr	0.5353
mlp	11	lr	0.3895
mlp	12	lr	0.4200
mlp	13	lr	0.2109
mlp	14	lr	0.4916
mlp	1	rule	0.3062
mlp	11	rule	0.3392
mlp	12	rule	0.3006
mlp	13	rule	0.3333
mlp	14	rule	0.3176
mlp	1	bbox	0.2865
mlp	11	bbox	0.2013
mlp	12	bbox	0.3042
mlp	13	bbox	0.3523
mlp	14	bbox	0.3568
mlp	1	shadow	0.3222
mlp	11	shadow	0.3512
mlp	12	shadow	0.2216
mlp	13	shadow	0.2583
mlp	14	shadow	0.2056
dee	6	rule	0.01792
dee	8	rule	0.0117
dee	9	rule	0.0008
dee	6	bbox	0.0
dee	8	bbox	0.0
dee	9	bbox	0.0
dee	6	shadow	0.0
dee	8	shadow	0.0136
dee	9	shadow	0.0
dp	6	rule	0.1276
dp	8	rule	0.1637
dp	9	rule	0.1890
dp	6	bbox	0.1496
dp	8	bbox	0.1977
dp	9	bbox	0.1457
dp	6	shadow	0.1315
dp	8	shadow	0.1243
dp	9	shadow	0.1460
dp	8	lr	0.1112
dp	9	lr	0.1810
dp	6	lr	0.1227
dp	7	lr	0.1311
dp	8	mlp	0.0965
dp	9	mlp	0.0869
dp	6	mlp	0.0991
dp	7	mlp	0.1060
ensemble	8	lr	0.1028
ensemble	9	lr	0.1134
ensemble	7	lr	0.1362
ensemble	6	lr	0.1151
ensemble	1	lr	0.1881
ensemble	8	mlp	0.1189
ensemble	9	mlp	0.1548
ensemble	7	mlp	0.1766
ensemble	6	mlp	0.2135
ensemble	1	mlp	0.1850
ensemble	1	rule	0.1175
ensemble	6	rule	0.1179
ensemble	8	rule	0.1170
ensemble	9	rule	0.0
ensemble	1	bbox	0.1334
ensemble	6	bbox	0.1633
ensemble	8	bbox	0.0
ensemble	9	bbox	0.0
ensemble	1	shadow	0.1666
ensemble	6	shadow	0.1159
ensemble	8	shadow	0.1999
ensemble	9	shadow	0.1999

Table C.1: Table of membership advantage on the Synthetic dataset across all (model, seed, attack) combinations.

Model	Seed	Attack Type	Advantage
model	split	attack	advantage
mlp	1	lr	0.4056
mlp	11	lr	0.5934
mlp	12	lr	0.4313
mlp	13	lr	0.3333
mlp	14	lr	0.4246
mlp	1	mlp	0.1485
mlp	11	mlp	0.1989
mlp	12	mlp	0.1059
mlp	13	mlp	0.2132
mlp	14	mlp	0.1621
dee	1	mlp	0.0
dee	11	mlp	0.0
dee	12	mlp	0.0
dee	13	mlp	0.0
dee	14	mlp	0.0
dee	1	lr	0.0800
dee	11	lr	0.0
dee	12	lr	0.0
dee	13	lr	0.0
dee	14	lr	0.0
dee	1	rule	0.0
dee	11	rule	0.0
dee	12	rule	0.0
dee	13	rule	0.0
dee	14	rule	0.0
dee	1	bbox	0.0
dee	11	bbox	0.0
dee	12	bbox	0.0
dee	13	bbox	0.0
dee	14	bbox	0.0
dee	1	shadow	0.0
dee	11	shadow	0.0
dee	12	shadow	0.0
dee	13	shadow	0.0
dee	14	shadow	0.0
mlp	1	rule	0.1401
mlp	11	rule	0.1545
mlp	12	rule	0.1061
mlp	13	rule	0.1376
mlp	14	rule	0.1109
mlp	1	bbox	0.0
mlp	11	bbox	0.0
mlp	12	bbox	0.0
mlp	13	bbox	0.0
mlp	14	bbox	0.0

Model	Seed	Attack Type	Membership Advantage
mlp	1	shadow	0.0644
mlp	11	shadow	0.1354
mlp	12	shadow	0.2663
mlp	13	shadow	0.1537
mlp	14	shadow	0.2205
dp	13	mlp	0.1045
dp	14	mlp	0.1248
dp	13	rule	0.1712
dp	14	rule	0.1697
dp	13	bbox	0.0
dp	14	bbox	0.0
dp	13	shadow	0.0
dp	14	shadow	0.1233
dp	14	lr	0.1058
dp	12	lr	0.1264
dp	13	lr	0.1177
dp	11	lr	0.0914
ensemble	13	lr	0.2220
ensemble	11	lr	0.1112
ensemble	14	lr	0.1632
ensemble	1	lr	0.1524
ensemble	12	lr	0.1412
ensemble	13	mlp	0.1204
ensemble	11	mlp	0.1256
ensemble	14	mlp	0.1122
ensemble	1	mlp	0.1064
ensemble	12	mlp	0.1286
ensemble	1	rule	0.1130
ensemble	11	rule	0.1158
ensemble	12	rule	0.1542
ensemble	13	rule	0.1306
ensemble	14	rule	0.1164
ensemble	1	bbox	0
ensemble	11	bbox	0
ensemble	12	bbox	0
ensemble	13	bbox	0
ensemble	14	bbox	0
ensemble	1	shadow	0.1438
ensemble	11	shadow	0.1014
ensemble	12	shadow	0.1356
ensemble	13	shadow	0.1114
ensemble	14	shadow	0.1205

Table C.2: Table of membership advantage on the Adult dataset across all (model, seed, attack) combinations.

Model	Seed	Attack Type	Advantage
mlp	1	lr	0.1179
mlp	11	lr	0.1476
mlp	12	lr	0.1705
mlp	13	lr	0.1667
mlp	14	lr	0.1574
mlp	1	mlp	0.2015
mlp	11	mlp	0.1835
mlp	12	mlp	0.2946
mlp	13	mlp	0.1493
mlp	14	mlp	0.2001
dee	1	mlp	0.0
dee	11	mlp	0.0
dee	12	mlp	0.0
dee	13	mlp	0.0
dee	14	mlp	0.0
dee	1	lr	0.0769
dee	11	lr	0.0
dee	12	lr	0.0
dee	13	lr	0.0105
dee	14	lr	0.0588
dee	1	rule	0.09491
dee	11	rule	0.0671
dee	12	rule	0.0849
dee	13	rule	0.0942
dee	14	rule	0.0436
dee	1	bbox	0.0
dee	11	bbox	0.0
dee	12	bbox	0.0
dee	13	bbox	0.0
dee	14	bbox	0.0
dee	1	shadow	0.0
dee	11	shadow	0.0
dee	12	shadow	0.0
dee	13	shadow	0.0
dee	14	shadow	0.0
mlp	1	rule	0.2425
mlp	11	rule	0.2241
mlp	12	rule	0.2227
mlp	13	rule	0.3759
mlp	14	rule	0.2888
mlp	1	bbox	0.1548
mlp	11	bbox	0.1905
mlp	12	bbox	0.1487
mlp	13	bbox	0.0
mlp	14	bbox	0.1163
mlp	1	shadow	0.1110
mlp	11	shadow	0.1411
mlp	12	shadow	0.2784
mlp	13	shadow	0.2727
mlp	14	shadow	0.2205
dp	13	rule	0.07676
dp	14	rule	0.05799
dp	13	bbox	0.0
dp	14	bbox	0.0
dp	13	shadow	0.04939
dp	14	shadow	0.01044
dp	13	mlp	0.20000
dp	14	mlp	0.1314
dp	13	lr	0.0817
dp	14	lr	0.1385
ensemble	13	lr	0.3636
ensemble	11	lr	0.2154
ensemble	14	lr	0.1316
ensemble	1	lr	0.2628
ensemble	12	lr	0.1500
ensemble	13	mlp	0.1667
ensemble	11	mlp	0.6667
ensemble	14	mlp	0.2500
ensemble	1	mlp	0.2076
ensemble	12	mlp	0.3957
ensemble	1	rule	0.1440
ensemble	11	rule	0.1683
ensemble	12	rule	0.1951
ensemble	13	rule	0.1945
ensemble	14	rule	0.1845
ensemble	1	bbox	0
ensemble	11	bbox	0
ensemble	12	bbox	0
ensemble	13	bbox	0
ensemble	14	bbox	0
ensemble	1	shadow	0.1986
ensemble	11	shadow	0.1190
ensemble	12	shadow	0.1463
ensemble	13	shadow	0.1205
ensemble	14	shadow	0.1114

Table C.3: Table of membership advantage on the COMPAS dataset across all (model, seed, attack) combinations.

Model	Seed	Attack Type	Advantage
mlp	1	lr	0.3333
mlp	11	lr	0.6667
mlp	12	lr	0.3143
mlp	13	lr	0.6869
mlp	14	lr	0.3056
mlp	1	mlp	0.4513
mlp	11	mlp	0.5036
mlp	12	mlp	0.4015
mlp	13	mlp	0.6095
mlp	14	mlp	0.4580
dee	1	mlp	0.0
dee	11	mlp	0.0
dee	12	mlp	0.0
dee	13	mlp	0.0
dee	14	mlp	0.0
dee	1	lr	0.0
dee	11	lr	0.0
dee	12	lr	0.0527
dee	13	lr	0.0125
dee	14	lr	0.0482
dee	1	rule	0.0
dee	11	rule	0.0
dee	12	rule	0.01177
dee	13	rule	0.0
dee	14	rule	0.0021
dee	1	bbox	0.0
dee	11	bbox	0.0
dee	12	bbox	0.0
dee	13	bbox	0.0
dee	14	bbox	0.0
dee	1	shadow	0.0
dee	11	shadow	0.0
dee	12	shadow	0.0
dee	13	shadow	0.0
dee	14	shadow	0.00197
mlp	1	rule	0.3044
mlp	11	rule	0.2790
mlp	12	rule	0.3546
mlp	13	rule	0.3032
mlp	14	rule	0.4387
mlp	1	bbox	0.2321
mlp	11	bbox	0.2530
mlp	12	bbox	0.2111
mlp	13	bbox	0.2753
mlp	14	bbox	0.2714
mlp	1	shadow	0.0
mlp	11	shadow	0.6403
mlp	12	shadow	0.03899
mlp	13	shadow	0.1482
dp	13	rule	0.04592
dp	14	rule	0.04301
dp	13	bbox	0.1057
dp	14	bbox	0.1743
dp	13	shadow	0.0457
dp	14	shadow	0.09831
dp	14	mlp	0.5718
dp	14	lr	0.5131
ensemble	13	lr	0.4179
ensemble	11	lr	0.4860
ensemble	14	lr	0.4779
ensemble	1	lr	0.4522
ensemble	12	lr	0.4852
ensemble	13	mlp	0.5924
ensemble	11	mlp	0.5152
ensemble	14	mlp	0.5769
ensemble	1	mlp	0.5120
ensemble	12	mlp	0.5785
ensemble	1	rule	0.1043
ensemble	11	rule	0.1064
ensemble	12	rule	0.1084
ensemble	13	rule	0.1034
ensemble	14	rule	0.10001
ensemble	1	bbox	0.1003
ensemble	11	bbox	0.1004
ensemble	12	bbox	0.1006
ensemble	13	bbox	0.1004
ensemble	14	bbox	0.1008
ensemble	1	shadow	0.1009
ensemble	11	shadow	0.1008
ensemble	12	shadow	0.1005
ensemble	13	shadow	0.1008
ensemble	14	shadow	0.1010

Table C.4: Table of membership advantage on the Texas dataset across all (model, seed, attack) combinations.

Model	Seed	Attack Type	Advantage
dee	6	lr	0.0
dee	7	lr	0.0
dee	8	lr	0.0
dee	1	lr	0.0
dee	6	mlp	0.0
dee	7	mlp	0.0
dee	8	mlp	0.0
dee	1	mlp	0.0
dee	6	svm	0.0
dee	7	svm	0.0
dee	8	svm	0.0
dee	1	svm	0.0
mlp	1	mlp	0.4839
mlp	11	mlp	0.5812
mlp	12	mlp	0.5270
mlp	13	mlp	0.4075
mlp	14	mlp	0.2780
mlp	1	lr	0.3805
mlp	11	lr	0.3330
mlp	12	lr	0.2310
mlp	13	lr	0.1310
mlp	14	lr	0.3651
mlp	1	rule	0.5258
mlp	11	rule	0.2237
mlp	12	rule	0.0
mlp	13	rule	0.2479
mlp	14	rule	0.2612
mlp	1	bbox	0.1451
mlp	11	bbox	0.2551
mlp	12	bbox	0.2119
mlp	13	bbox	0.2479
mlp	14	bbox	0.1242
mlp	1	shadow	0.19994
mlp	11	shadow	0.19902
mlp	12	shadow	0.12242
mlp	13	shadow	0.2354
dp	13	rule	0.1795
dp	14	rule	0.1779
dp	13	bbox	0.2050
dp	14	bbox	0.1854
dp	13	shadow	0.0
dp	14	shadow	0.0005
dp	1	mlp	0.0496
dp	11	mlp	0.0338
dp	12	mlp	0.0245
dp	13	mlp	0.0237
dp	1	lr	0.0266
dp	11	lr	0.0158
dp	12	lr	0.0055
dp	13	lr	0.0180
dee	1	rule	0.0
dee	12	rule	0.0
dee	13	rule	0.0
dee	14	rule	0.0
dee	1	bbox	0.0
dee	12	bbox	0.0
dee	13	bbox	0.0
dee	14	bbox	0.0
dee	1	shadow	0.0
dee	12	shadow	0.0
dee	13	shadow	0.0
dee	14	shadow	0.0
ensemble	6	rule	0.0
ensemble	7	rule	0.0
ensemble	8	rule	0.0
ensemble	6	bbox	0.0
ensemble	7	bbox	0.0
ensemble	8	bbox	0.0
ensemble	6	shadow	0.0
ensemble	7	shadow	0.0
ensemble	8	shadow	0.0
ensemble	6	mlp	0.1220
ensemble	7	mlp	0.1366
ensemble	8	mlp	0.1269
ensemble	6	lr	0.1072
ensemble	7	lr	0.1025
ensemble	8	lr	0.1058

Table C.5: Table of membership advantage on the FEMNIST dataset across all (model, seed, attack) combinations.

Model	Seed	Attack	Advantage
dee	6	lr	0.0
dee	7	lr	0.0
dee	8	lr	0.0
dee	9	lr	0.0
dee	6	mlp	0.0
dee	7	mlp	0.0
dee	8	mlp	0.0
dee	9	mlp	0.0
dp	6	lr	0.9872
dp	9	mlp	0.9213
mlp	6	lr	0.9862
mlp	7	lr	0.9100
mlp	8	lr	0.9200
mlp	9	lr	0.8800
mlp	6	mlp	0.9124
mlp	7	mlp	0.9105
mlp	8	mlp	0.8904
mlp	9	mlp	0.9028
mlp	1	rule	0.7798
mlp	11	rule	0.7799
mlp	12	rule	0.7680
mlp	13	rule	0.7800
mlp	14	rule	0.7682
mlp	1	bbox	0.8573
mlp	11	bbox	0.8573
mlp	12	bbox	0.8572
mlp	13	bbox	0.8571
mlp	14	bbox	0.8571
mlp	1	shadow	0.7797
mlp	11	shadow	0.7682
mlp	12	shadow	0.7799
mlp	13	shadow	0.7800
dee	6	rule	0.0
dee	7	rule	0.0
dee	8	rule	0.0
dee	9	bbox	0.0
dee	6	bbox	0.0
dee	7	bbox	0.0
dee	8	shadow	0.0
dee	9	shadow	0.0
dee	10	shadow	0.0
dp	6	rule	0.7733
dp	7	rule	0.7739
dp	6	bbox	0.8585
dp	7	bbox	0.8504
dp	6	shadow	0.8501
dp	7	shadow	0.8711
ensemble	6	rule	0.5797
ensemble	7	rule	0.7750
ensemble	8	rule	0.5793
ensemble	6	bbox	0.6009
ensemble	7	bbox	0.9000
ensemble	8	bbox	0.6002
ensemble	6	shadow	0.7393
ensemble	7	shadow	0.7515
ensemble	8	shadow	0.7934
ensemble	6	mlp	0.6950
ensemble	7	mlp	0.6036
ensemble	8	mlp	0.6752
ensemble	6	lr	0.4022
ensemble	7	lr	0.4982
ensemble	8	lr	0.4990

Table C.6: Table of membership advantage on the MNIST dataset across all (model, seed, attack) combinations.

Model	Seed	Attack	Advantage	Model	Seed	Attack	Advantage
mlp	1	mlp	0.3528	ensemble	1	shadow	0.3246
mlp	11	mlp	0.2619	ensemble	11	shadow	0.2991
mlp	12	mlp	0.3572	ensemble	12	shadow	0.2886
mlp	13	mlp	0.2630	ensemble	13	shadow	0.3014
mlp	14	mlp	0.3948	ensemble	14	shadow	0.3215
mlp	1	lr	0	dp	1	rule	0.1894
mlp	11	lr	0	dp	11	rule	0.1944
mlp	12	lr	0	dp	12	rule	0.1678
mlp	13	lr	0	dp	13	rule	0.1236
mlp	14	lr	0	dp	14	rule	0.1710
mlp	1	rule	0.3861	dp	1	bbox	0
mlp	11	rule	0.3872	dp	11	bbox	0
mlp	12	rule	0.3095	dp	12	bbox	0
mlp	13	rule	0.2881	dp	13	bbox	0
mlp	14	rule	0.2955	dp	14	bbox	0
mlp	1	bbox	0.2909	dp	1	shadow	0.1909
mlp	11	bbox	0.3194	dp	11	shadow	0.1988
mlp	12	bbox	0.2596	dp	12	shadow	0.2090
mlp	13	bbox	0.3006	dp	13	shadow	0.1873
mlp	14	bbox	0.2804	dp	14	shadow	0.1899
mlp	1	shadow	0.2911	ee	1	mlp	0.0245
mlp	11	shadow	0.3005	ee	11	mlp	0.0228
mlp	12	shadow	0.3495	ee	12	mlp	0.0495
mlp	13	shadow	0.3293	ee	13	mlp	0.0379
mlp	14	shadow	0.2975	ee	14	mlp	0.0383
ee	1	rule	0	ee	1	lr	0
ee	11	rule	0.023	ee	11	lr	0
ee	12	rule	0	ee	12	lr	0
ee	13	rule	0	ee	13	lr	0
ee	14	rule	0	ee	14	lr	0
ee	1	bbox	0	ensemble	1	mlp	0.3511
ee	11	bbox	0	ensemble	11	mlp	0.4228
ee	12	bbox	0	ensemble	12	mlp	0.3664
ee	13	bbox	0	ensemble	13	mlp	0.3539
ee	14	bbox	0	ensemble	14	mlp	0.3234
ee	1	shadow	0	ensemble	1	lr	0
ee	11	shadow	0	ensemble	11	lr	0
ee	12	shadow	0	ensemble	12	lr	0
ee	13	shadow	0	ensemble	13	lr	0
ee	14	shadow	0	ensemble	14	lr	0
ensemble	1	rule	0.3572	dp	1	mlp	0.1689
ensemble	11	rule	0.3913	dp	11	mlp	0.1378
ensemble	12	rule	0.3374	dp	12	mlp	0.1441
ensemble	13	rule	0.3649	dp	13	mlp	0.1476
ensemble	14	rule	0.3141	dp	14	mlp	0.1434
ensemble	1	bbox	0.2909	dp	1	lr	0
ensemble	11	bbox	0.2984	dp	11	lr	0
ensemble	12	bbox	0.3004	dp	12	lr	0
ensemble	13	bbox	0.3193	dp	13	lr	0
ensemble	14	bbox	0.2822	dp	14	lr	0

Table C.7: Table of membership advantage on the INaturalist dataset across all (model, seed, attack) combinations.

Model	Seed	Attack	Advantage
mlp	7	rule	0.4924
mlp	8	rule	0.6864
mlp	9	rule	0.6420
mlp	0	rule	0.6593
mlp	7	bbox	0.5364
mlp	8	bbox	0.5613
mlp	9	bbox	0.5820
mlp	0	bbox	0.5302
mlp	7	shadow	0.5819
mlp	8	shadow	0.4664
mlp	9	shadow	0.6139
mlp	0	shadow	0.5927
dp	7	rule	0.0710
dp	8	rule	0.1524
dp	9	rule	0.1695
dp	7	bbox	0
dp	8	bbox	0
dp	9	bbox	0
dp	7	shadow	0.0333
dp	8	shadow	0.1259
dp	9	shadow	0.0562
ensemble	7	rule	0.5605
ensemble	8	rule	0.4265
ensemble	9	rule	0.5275
ensemble	0	rule	0.5039
ensemble	7	bbox	0.7364
ensemble	8	bbox	0.7613
ensemble	9	bbox	0.7280
ensemble	0	bbox	0.7494
ensemble	7	shadow	0.4056
ensemble	8	shadow	0.2383
ensemble	9	shadow	0.4861
ensemble	0	shadow	0.4268
ee	7	rule	0.0059
ee	8	rule	0.0072
ee	9	rule	0
ee	0	rule	0
ee	7	bbox	0
ee	8	bbox	0
ee	9	bbox	0
ee	0	bbox	0
ee	7	shadow	0
ee	8	shadow	0
ee	9	shadow	0.0026
ee	0	shadow	0
mlp	7	mlp	0.3025
mlp	8	mlp	0.2027
mlp	9	mlp	0.3083
mlp	0	mlp	0.3533
ee	7	mlp	0
ee	8	mlp	0
ee	9	mlp	0
ee	0	mlp	0
ensemble	7	mlp	0.1375
ensemble	8	mlp	0.1272
ensemble	9	mlp	0.1245
ensemble	0	mlp	0.1857
dp	7	mlp	0.1159
dp	8	mlp	0.0528
dp	9	mlp	0.1201
dp	0	mlp	0.0810
mlp	7	lr	0.2684
mlp	8	lr	0.2063
mlp	9	lr	0.1958
mlp	0	lr	0.2103
ee	7	lr	0
ee	8	lr	0
ee	9	lr	0
ee	0	lr	0
ensemble	7	lr	0.0073
ensemble	8	lr	0.0343
ensemble	9	lr	0.0132
ensemble	0	lr	0.0291
dp	7	lr	0.0203
dp	8	lr	0.0197
dp	9	lr	0.0149
dp	0	lr	0.0577

Table C.8: Table of membership advantage on the MIMIC-CXR-EGD dataset across all (model, seed, attack) combinations.

Model	Dataset	Seed	Accuracy	model	dataset	split	acc	model	dataset	split	acc
mlp	synthetic	1	81.55	dp	adult	10	76.37	unet	cxr	0	74.27
mlp	synthetic	2	77.68	dee	adult	10	77.38	unet	cxr	9	74.52
mlp	synthetic	3	75.83	dee	adult	11	76.38	unet	cxr	8	79.21
mlp	synthetic	4	79.06	dee	adult	12	75.21	unet	cxr	7	74.28
mlp	synthetic	5	68.08	dee	adult	13	77.52	dp	cxr	0	69.30
mlp	synthetic	6	79.15	dee	adult	14	76.38	dp	cxr	9	68.01
mlp	synthetic	7	63.78	dee	texas	10	72.14	dp	cxr	8	68.44
mlp	synthetic	8	87	dee	texas	11	71.36	dp	cxr	7	67.97
dee	synthetic	1	52.68	dee	texas	12	73.21	ee	cxr	0	74.11
dee	synthetic	2	86.07	dee	texas	13	72.52	ee	cxr	9	76.51
dee	synthetic	3	85.61	dee	texas	14	74.73	ee	cxr	8	76.99
dee	synthetic	4	65.04	ensemble	texas	1	72.16	ee	cxr	7	76.51
dee	synthetic	5	77.17	ensemble	texas	11	71.43	ensemble	cxr	0	74.59
dee	synthetic	6	84.78	ensemble	texas	12	72.86	ensemble	cxr	9	78.90
dee	synthetic	7	84.41	ensemble	texas	13	71.50	ensemble	cxr	8	78.90
dee	synthetic	8	78.04	ensemble	texas	14	72.81	ensemble	cxr	7	74.11
dee	synthetic	9	78.87	dee	compas	10	91.92	mlp	nature	1	83.74
dee	synthetic	10	82.56	dee	compas	11	90.96	mlp	nature	11	82.86
ensemble	synthetic	5	82.56	dee	compas	12	91.57	mlp	nature	12	86.26
ensemble	synthetic	6	78.87	dee	compas	13	92.02	mlp	nature	13	83.21
ensemble	synthetic	7	78.04	dee	compas	14	92.86	mlp	nature	14	83.79
ensemble	synthetic	8	84.41	ensemble	adult	1	76.38	ee	nature	1	82.36
ensemble	synthetic	9	84.78	ensemble	adult	11	73.18	ee	nature	11	84.83
dp	synthetic	1	78.04	ensemble	adult	12	77.69	ee	nature	12	83.88
dp	synthetic	2	76.01	ensemble	adult	13	72.81	ee	nature	13	84.57
dp	synthetic	3	77.68	ensemble	adult	14	77.58	ee	nature	14	85.06
dp	synthetic	4	77.31	dp	adult	11	76.23	dp	nature	1	62.14
dp	synthetic	5	74.26	dp	adult	12	75.15	dp	nature	11	70.689
dp	synthetic	6	71.68	dp	adult	13	76.35	dp	nature	12	71.05
dp	synthetic	7	77.58	dp	adult	14	74.57	dp	nature	13	72.58
dp	synthetic	8	77.67	cnn	mnist	10	98.75	dp	nature	14	73.72
dp	synthetic	9	65.13	cnn	mnist	11	98.86	ensemble	nature	1	82.51
mlp	compas	1	87.9	cnn	mnist	12	98.84	ensemble	nature	11	83.07
mlp	compas	11	91.04	cnn	mnist	13	98.77	ensemble	nature	12	82.13
mlp	compas	12	89.82	cnn	mnist	14	98.80	ensemble	nature	13	84.51
mlp	compas	13	89.10	ensemble	mnist	9	93.56	ensemble	nature	14	84.76
mlp	compas	14	91.68	ensemble	mnist	9c	95.44				
mlp	texas	1	82.47	ensemble	mnist	6	92.36				
mlp	texas	11	83.44	ensemble	mnist	6c	91.28				
mlp	texas	12	84.00	ensemble	mnist	62	92.37				
mlp	texas	13	84.27	dee	mnist	10	98.89				
mlp	adult	10	74.02	dee	mnist	11	99.01				
mlp	adult	11	73.23	dee	mnist	12	98.65				
mlp	adult	12	77.48	dee	mnist	13	99.10				
mlp	adult	13	74.42	dee	mnist	14	98.80				
mlp	adult	14	75.35	dp	mnist	10	90.26				
cnn	femnist	1	80.99	dp	mnist	11	91.57				
cnn	femnist	6	85.07	dp	mnist	12	93.35				
cnn	femnist	7	85.45	dp	mnist	13	92.00				
cnn	femnist	13	85.84	dp	mnist	14	91.21				
cnn	femnist	14	86.03	dee	femnist	1	86.02				
dp	femnist	1	82.98	dee	femnist	6	85.93				
dp	femnist	11	83.01	dee	femnist	7	86.58				
dp	femnist	12	84.93	dee	femnist	13	82.84				
dp	femnist	13	80.86	dee	femnist	14	84.23				
dp	femnist	14	82.77	ensemble	femnist	6	86.96				
dp	compas	1	73.82	ensemble	femnist	7	86.31				
dp	compas	11	73.81	ensemble	femnist	8	85.16				
dp	compas	12	78.73	ensemble	femnist	9	85.74				
dp	compas	13	77.22	ensemble	femnist	10	85.38				
dp	compas	14	73.02	ensemble	compas	1	95.75				
dp	texas	1	72.47	ensemble	compas	11	95.45				
dp	texas	11	73.44	ensemble	compas	12	93.80				
dp	texas	12	74.00	ensemble	compas	13	94.03				
dp	texas	13	74.27	ensemble	compas	14	93.99				

Table C.9: Model accuracy of trained models across all dataset and training hyperparameters used.