University of Glasgow | School of Computing Science

Honours Individual Project Dissertation

Development of machine learning models to detect Arrhythmia based on ECG data - Explainable Models

Hector J. Jones

April 1, 2022

Abstract

Cardiovascular diseases (CVDs) are one of the leading causes of death globally. Preventative measures like monitoring electrocardiograms (ECGs) to screen for arrhythmias is a cost effective method to mitigate the risks associated with CVDs. However, manual interpretation by cardiologists takes a lot of time and therefore automated interpretation using machine learning methods are becoming more popular. The issue with machine learning methods, particularly deep learning methods, is that their decision making process is difficult to understand for humans. This means it is difficult for them to be trusted in a medical setting, therefore emphasis on explainability in machine learning models needs to be increased.

In this project, four novel classifier models that incorporate the existing explainability methods; integrated and expected gradients, into their training and are based on a Convolutional Neural Network (CNN) architecture, are proposed. The performances of these novel models are compared quantitatively to a baseline CNN and Support Vector Machine (SVM). The integrated and expected gradient explainability methods are also qualitatively compared to the established Gradient Weighted Class Activation Maps (GradCAM) method. The model performances were evaluated using 5 repeated stratified 10-fold cross validation and statistical analysis was performed to investigate the significance of the results.

After evaluation, it was found that although one of the novel model's performances in the experiments was comparable to that of the baselines, the baseline CNN model was the best classifier overall. The integrated gradients explainability method was also found to provide the clearest explanations about a model's decision making process.

Acknowledgements

I want to thank my close friends and family for their continuous encouragement and support throughout the course of this project. I would also like to thank Dr. Fani Deligianni for all her help throughout this project, as it was my first ever exposure to machine learning. I am incredibly grateful for all the aid I have received from Dr. Deligianni and I hope to further pursue this domain of computer science with everything that I have learnt thus far.

Education Use Consent

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Requirements Analysis 2
1.3 Dissertation Structure 2

2 Background **3**
2.1 ECG specifics 3
2.2 MIT BIH Dataset 3
2.3 Automatic classification of heartbeats from ECG recordings 4
2.3.1 Support Vector Machines 4
2.3.2 Convolutional Neural Networks 5
2.4 Interpretability and Explainability 5
2.4.1 Existing Explainability Methods 6
2.4.2 Attributions Methods 7
2.4.3 Integrated Gradients 7
2.4.4 Attribution Priors 8
2.4.5 Expected Gradients 8
2.5 Building upon this previous work 8
2.6 Chapter Summary 8

3 Methods **9**
3.1 Machine Learning Models used in this Project 9
3.1.1 SVM 9
3.1.2 Convolutional Neural Network 9
3.2 Gradient Based Explainability Methods 9
3.2.1 Integrated Gradients 9
3.2.2 Expected Gradients 11
3.3 Attribution Priors 11
3.4 Technologies Used 13
3.5 Data Pre-Processing 13
3.6 Class Imbalances 13
3.7 Hyper-parameter Tuning 15
3.8 Chapter Summary 16

4 Evaluation **17**
4.1 Beat Holdout Experiment 17
4.1.1 Repeated Stratified K-Fold Cross Validation 17
4.1.2 Performance Results for Each Model 17
4.1.3 Quantitative Evaluation - Statistical Analysis 19
4.2 Patient Holdout Experiment 21
4.2.1 Performance Results for Each Model 21
4.3 Explainability Results 22
4.3.1 Beat Holdout 22
4.3.2 Patient Holdout 24

5 Discussion and Conclusion **26**
5.1 Summary 26
5.2 Beat Holdout Experiment 26
5.3 Patient Holdout Experiment 27
5.4 Explainability Methods 28
5.5 Conclusion 28
5.6 Limitations and Future Work 28

Appendices **30**

A Appendices **30**
A.1 Performance Metrics 30
A.1.1 Accuracy 30
A.1.2 Precision 30
A.1.3 Recall 30
A.1.4 F1 Macro 30

Bibliography **36**

1 Introduction

In this chapter the motivations of this thesis will be outlined, as well as presenting the structure of the rest of the chapters and a brief description about the content of each.

1.1 Motivation

Heart arrhythmias occur when there is an irregularity in your heartbeat and can be an indicator of many different problems, such as cardiovascular diseases (CVD) or thyroid disease (NHS 2019). Due to the fact that CVDs are consistently considered as the leading cause of death worldwide (WHO 2019) it is therefore clear that methods to monitor the conditions of patients' hearts are crucial to mitigate the risk factors associated with contracting such diseases. In works such as Roth et al. (2020) it has been shown that the number of cases of CVD and deaths caused by it have increased steadily between 1990 and 2019. This shows that more than ever methods to prevent deaths caused by CVD are a strong requirement today.

ECGs have been shown to be popular and cost-effective methods to monitor patients' heart activity to help diagnose CVD (Serhani et al. 2020). Since the inception of ECGs, cardiologists have been trained to manually interpret the information displayed by these heart monitoring devices, to identify if heart arrhythmias are occurring and whether further action should be taken.

In recent years there has been rapid development in machine and deep learning techniques, especially in medical domains where classification problems are required to be solved (Shen et al. 2017; Anaya-Isaza et al. 2021; Yadav and Jadhav 2019). The promising performance of these models have clear benefits in assisting medical professionals in clinical and research applications. This is due to their high accuracies achieved by analysing very large datasets at speeds unachievable by humans.

A significant concern regarding most machine and all deep learning methods is to do with their black-box nature. Due to their incredibly complex structure, it is very difficult for users of these methods to understand their inner workings and how the models arrive at certain decisions. It is therefore incredibly difficult for users of these methods, especially medical practitioners, to trust deep learning models enough to use them in high-risk decision making domains where patients' lives are at stake. Following from this it is clear why this became such an active research area in parallel to machine learning methods, in regard to reducing the black-box property of these models. This domain of research revolves around increasing the explainability of machine learning models, or better yet to make them interpretable by design. By successfully and consistently increasing the user trust in these machine and deep learning models, the benefits of their performance could have many beneficial impacts, especially in the medical domain.

Many different explainability methods applicable to machine learning models to uncover their black-box behaviour exist. These can be placed into categories such as if the method provides *global* or *local* explainability and whether the method is model *specific* or *agnostic*. This will be discussed further in Section 2.4. Integrated and expected gradients are two *local* explainability methods that are model *agnostic* and have been used to facilitate user understanding of the model

decision making process. With so many different explanation methods available and there not being a standard quantitative way to compare them, it is also very difficult to evaluate which methods are objectively better. There have been huge steps in trying to develop models that take both classification performance and explainability into account throughout model design, particularly through the use of Expected Gradients in Erion et al. (2021). At the time of writing this paper, there is a gap in the research focused on integrated or expected gradients to train models for classification of ECG timeseries data. In this paper, four novel deep learning models are proposed, that incorporate integrated and expected gradients into training and are all based on a Convolution Neural Network (CNN) architecture. These novel models will then be compared to a baseline CNN and a Support Vector Machine (SVM). All of the models under investigation were trained and tested using the MIT-BIH dataset (Moody and Mark 2001). Statistical hypothesis testing will be using to compare the performances of the different models. Additionally, to evaluate the explainability of both integrated and expected gradients, the visualisations of both of these will be compared to those produced using GradCAM (Selvaraju et al. 2017).

1.2 Requirements Analysis

Should Have: at the start of this project, there were a number of requirements that had to be achieved:

- To develop deep learning models for timeseries ECG classification that are inspired from existing state-of-the-art models.
- Evaluate and compare their performances against a shallow machine learning model, such as an SVM, as well as a baseline deep learning model.
- Ensure that this novel model has an aspect of interpretability or explainability, for example using gradient-based methods, that can reduce its black-box behaviour.

Achieved: at the end of the project all of the above were achieved as well as the items listed below:

- Investigation of whether incorporating gradient-based attribution priors in the training loss function improves the accuracy of deep learning models.
- Comparisons were made between models incorporating gradient-based attribution priors, with respect to their explainability and performance in classification.

1.3 Dissertation Structure

- Chapter 2 will cover the MIT-BIH dataset and outline the basics of the problem domain. It will also discuss existing research concerning highly performing models as well as work carried out into interpretability and explainability, all of which this project builds upon.
- Chapter 3 goes through the design and implementation of models used in this project. Starting with the shallow SVM model used, followed by the architecture of the Convolutional Neural Network model used in the classification problem. Explanations of the Integrated and Expected gradient methods used to improve performance and provide visual explanations are then presented. This chapter will also go on to describe the technologies that were used in the project, the pre-processing that was applied to the data and the explainability techniques compared.
- Chapter 4 contains the evaluation of the different models on the dataset, as well as the results produced from this evaluation.
- Chapter 5 is the conclusion of this dissertation, where the results obtained are reflected upon and possible avenues for future work are outlined.

2 Background

2.1 ECG specifics

To better understand the literature surrounding beat classification from ECG recordings it is best to start with what an ECG is. The contractions of the heart are controlled by electrical impulses initiated by the sinoatrial node. These electrical impulses can be detected using electrodes connected to the ECG and represented as an ECG recording. Different ECG machines record a different number of amplifier channels, they are either a single lead ECG or a multi-lead ECG. When recording the electrical activity of the heart, it is possible to do so by placing the electrodes of the ECG at different locations, with varying orientations on the body, in order to gain spatial information.

Visually, an ECG is a long waveform consisting of multiple heartbeats. Segmenting the beats to inspect the morphology of each is how classification is performed. The morphology of a typical normal heartbeat consists of the P wave, the QRS complex (the Q, R and S waves), and the T wave. Each of these waves correspond to the electrical impulses entering and exiting different sections of the heart. The peak of the R wave, known as the R peak, is commonly used in the standard method of segmenting each individual beat. A diagram of an ECG beat and its labelled different segments can be found in Figure 2.1.

2.2 MIT BIH Dataset

The MIT BIH dataset is a popular, publicly accessible dataset used frequently for research in timeseries ECG classification (Moody and Mark 2001). The dataset consists of 48 records from 47 subjects, where 60% of patients were inpatients and 40% of patients were outpatients. Each of the records are an approximately 30 minute long recording from a two-lead ECG, that is digitized at 360 seconds per sample for each of the two channels. The heartbeats in the dataset were

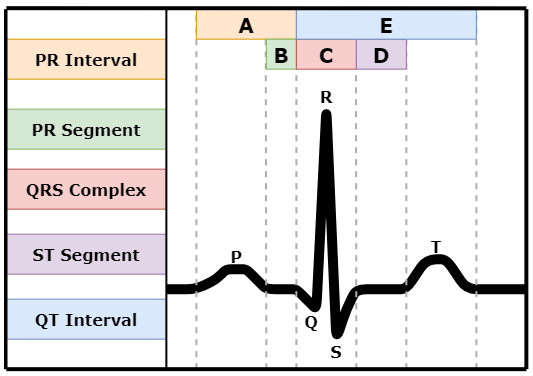


Figure 2.1: *A representation of what a single heartbeat appears like on an ECG recording, with labels describing each wave and each section of the beat.*

Table 2.1: *Heartbeat Classes as they are found in the MIT BIH dataset, along with their equivalent medical description. The ID column represents how the classes are represented as indexes throughout the project.*

ID	Heartbeat Class	Medical description
1	N	Normal beat
2	L	Left bundle branch block beat
3	R	Right bundle branch block beat
4	V	Premature ventricular contraction
5	A	Atrial premature beat
6	F	Fusion of ventricular and normal beat
7	f	Fusion of paced and normal beat
8	/	Paced beat

annotated by cardiologists, albeit there were a considerable number of heartbeats that could not be confidently classified. These annotations include the class that each heartbeat belongs to, and although this dataset contains many different classes, only eight will be considered throughout this project. The labels of these classes as they appear in the dataset, their accompanying descriptions and the numerical ID used to identify them throughout implementation is summarised in Table 2.1.

2.3 Automatic classification of heartbeats from ECG recordings

Early research into automatic classification of ECG recordings dates to as far back as 1978 in Rautaharju et al. (1978). This work discusses how utility of computer interpretation of ECG recordings was expanding. It goes onto explain how there were many benefits provided by this advancement, although the authors added that the stated advantages were subjective. Some of the advantages pointed out include the ability of computers to handle high volumes of ECG recordings, improving the quality of ECG interpretation possible by using a unified standard and most importantly the use of emergency ECG automatic interpretation when a cardiologist isn't available. This paper goes on to discuss some of the methods adopted by several existing programs at the time, which include using decision trees and maximal likelihood statistics to obtain possible classifications for interpretation. The use of automatic classification using artificial neural networks, has been prevalent since the early 1990s. Early works include that of Edenbeandt et al. (1993), Farrugia et al. (1993), Reddy et al. (1992) and Hedén et al. (1994), with networks ranging from multi-layer perceptrons (MLP) to multi-layer feed-forward networks. The common motivations of this research appeared to be comparing the accuracy of the automatic methods to that of cardiology experts. These works also agree that at the time, the use of neural networks had great potential to be used in the future to improve the performance of ECG interpretation programs.

2.3.1 Support Vector Machines

Support Vector Machines (SVMs) are a machine learning method that was proposed in Boser and Vapnik (1992) that is commonly used in classification and regression tasks. In a classification task SVMs work by creating a hyperplane that best separates data points into one of two clusters of their respective classes. This hyperplane effectively represents the classification decision boundary. Their effectiveness in the interpretation and classification of ECG heartbeats was demonstrated in early works such as Osowski et al. (2004), Kikawa and Oguri (2004) and Zhao and Zhang (2005). Particular in the problem domain of heartbeat classification using ECG signals, SVMs have proven to obtain high performances in metrics such as accuracy (Rabee and Barhumi 2012;

Jannah et al. 2021). It is for this reason that an SVM model will be used as the baseline model to compare against all other models defined in this project.

2.3.2 Convolutional Neural Networks

Convolution Neural Networks (CNNs) have become widely adopted as a deep learning technique, ever since its inception in 1987 (Atlas et al. 1987). Although CNNs are mostly used in image classification tasks due to their high performance (O'Shea and Nash 2015), research has shown them to be as equally powerful in most classification problem domains, particularly in classifying heartbeats.

CNNs are similar to typical artificial deep neural networks in their structure but stand out due to the convolutional layers that they contain. In the case of 2D CNN for image recognition, when an input is passed into a two-dimensional convolutional layer, a two-dimensional convolutional kernel or filter slides over sections of it, applying a mathematical operation to produce the output to be passed to the next layer of the network. The mathematical operation used by the convolutional kernel determines what kind of feature is extracted in this convolutional layer, in the case of image recognition this could be edge detection. As more and more convolutional layers are added to a CNN, increasingly complex features can be extracted from an input to the model. One-dimensional CNNs work in the same way as two-dimensional CNNs apart from the fact that instead one-dimensional convolutional kernels or filters are applied to inputs of each layer. There are other important layers that usually accompany convolutional layers in a CNN, such as batch normalisation layers which can be used after each convolutional layer to normalise features along a batch of the input to speed up and also stabilise training. Another important layer commonly used in CNNs is pooling layers, either max pooling or average pooling, used to downsample the output from a previous layer by using another kernel or filter to get either the max or average over a feature map produced by a convolutional layer. This downsampling method has been shown to reduce overfitting of models and help them generalise to unseen data. A final key part of a typical CNN are the fully connected dense layers, that is common in normal artificial neural networks and is where the final classification decision is produced.

The deep neural network evaluated in Hannun et al. (2019) showed that the sensitivity of the neural network had exceeded the average sensitivity achieved by a group of cardiologists. This paper suggested that the use of this automatic classification could help filter through and find more urgent conditions so that they receive medical attention quicker. The deep neural network proposed is similar to that of a CNN, and consists of many convolutional, batch normalisation and max pooling layers as described above. CNNs were also used in the works of Acharya et al. (2017) and Kiranyaz et al. (2016) whose model performance results are also outstanding. These studies boast that their models' performances are the current state-of-the-art for automatic ECG classification with respect to the screening of heart irregularities such as arrhythmia. However, they all do not address one of the biggest issues with deep learning at this current point in time, which is the lack of explainability or interpretability of models.

2.4 Interpretability and Explainability

Machine and deep learning methods have been prevalent for a while, however, for the most part they have remained black-box models; whereby their internal logic is not exposed to the user and is usually difficult to understand. The increased adoption of machine and deep learning models in high-stakes decision making and critical situations such as healthcare requires their users, such as medical practitioners, to have trust in them. Practical and ethical concerns about the black-box nature of these automatic decision systems have led to development of new research that attempts to increase interpretability and explainability when developing models.

Barredo Arrieta et al. (2020) provides an in-depth review of this literature and points out that there is inconsistent use and constant interchangeability between the terms 'interpretability' and 'explainability', as well as what is required for models to satisfy these definitions. The authors of this paper decided to clarify on a number of definitions, as shown below.

Interpretability: *"...refers to a passive characteristic of a model referring to the level at which a given model makes sense for a human observer."*

Explainability: *"...viewed as an active characteristic of a model, denoting any action or procedure taken by a model with the intent of clarifying or detailing its internal functions."*

Expanding upon these definitions, models satisfying the interpretability definition are described in Barredo Arrieta et al. (2020) as being interpretable by design, whereas models in general can be explained. This means that most models consisting of neural networks are not interpretable, since it is very difficult for a human to understand the steps taken from input to the output of the model. This is why increasing the understandability of Deep Learning methods to users is performed using *post-hoc explainability* whereby explanations for why and how these decisions were made are produced only once the models have been trained and made predictions. A further distinction of explainability method types concerns whether they are global or local explanations for a model or whether the explainability methods are model specific or model agnostic.

2.4.1 Existing Explainability Methods

Popular examples for *post-hoc local explainability* methods include CAM (Zhou et al. 2016) and Grad-CAM (Selvaraju et al. 2017), especially in image recognition. These methods produce attention heat maps to visualise regions of pixels that the model pays most attention to when making its classification decision. The issue with CAM is that it depends on a model containing a global average pooling (GAP) layer as well as a convolutional layer. However, Grad-CAM is a less restrictive version of CAM in that models only require a convolutional layer, and gradients are calculated between the output of the model with respect to the feature maps of the final convolutional layer to produce a heatmap. Overall, these methods are model specific, due to them only being implementable on models that contain convolutional layers, for example in CNNs or in Long Short-Term Memory (LSTM) networks. Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro et al. 2016) on the other hand is applicable to any model, providing local explanations that can be easily understood by non-expert users. A popular method for global explanations is Permutation Feature Importance (PFI), whereby the score achieved by a model is tracked when removing certain features from the input, to investigate how important each feature is to the model's accuracy for example. This method falls short in providing a detailed explanation of the model decision making process, however.

Rudin (2019) presents criticism towards explainability methods for models, particularly in situations where the decisions made are high stakes, for example in the medical domain. Although the paper mainly revolves around machine learning, it points out that *post-hoc explainability* methods such as attention heat maps can sometimes be misleading and not provide useful information about the decision making process. Furthermore, the paper fails to acknowledge that deep learning methods, that consist of neural networks, cannot be interpretable by design. Instead, research is presented that proposed a prototype network that produces non *post hoc explanations* whilst not having a trade-off between accuracy and interpretability.

Further work in Kindermans et al. (2019) and Viering et al. (2019) present supplementary evidence for how unreliable these *post hoc explanation* methods are with examples involving Grad-CAM and saliency maps. Research such as this exposes how certain manipulations of model architecture or inputs are shown to have little or no effect on prediction accuracy. This reinforces the point of view described in Rudin (2019) where understandability should be taken into account from the start of the model design process.

2.4.2 Attributions Methods

An early method for local explainability of non-linear classification models was proposed in Baehrens et al. (2010) and called attributions. In the context of machine learning this was described as showing what features of the input to a model actually attribute or contribute to the final prediction. This was achieved because calculating the gradients between the inputs and the output predictions of the model can be used to help users understand which particular inputs attributed to the final classification. Calculating attributions for a certain *i*th feature of an input to a linear model of the form:

$$f(x) = w_1x_1 + w_2x_2 + ...w_nx_n, \tag{2.1}$$

produces an *attribution score* $\Phi(i)$ where the calculation can be summarised as:

$$\Phi(f(x), i) = w_ix_i. \tag{2.2}$$

For non-linear models however, the gradients between the output predictions and the input are used to get the attribution, by multiplying the gradient, usually a partial derivative, by the input feature to get the *attribution score*:

$$\Phi(f(x), i) = x_i \times \frac{\partial f(x_i)}{\partial x_i}. \tag{2.3}$$

Numerous works demonstrated the use of this explainability method in a number of applications, for example deep learning image classification of different entities (Simonyan et al. 2014) and on the MNIST dataset (Shrikumar et al. 2017).

There were problems found with this method however, saturation being one of them which has been discussed in Glorot and Bengio (2010) and Sundararajan et al. (2016). For example, the gradients may be very small around a sample input or near flattened if the model uses a ReLU function, where $ReLU(x) = max(0, x)$. Small or even zero gradients would result in an incredibly low or even zero attribution score even if the output prediction did rely heavily on these features.

2.4.3 Integrated Gradients

In Sundararajan et al. (2017) the authors point out that there are a number of flaws with previously proposed gradient based attribution methods for explainability. A number of axioms, that these attribution methods should satisfy, are presented and used to show how previous methods fail these axioms. The researchers used this axiomatic framework to develop a novel attribution method called Integrated Gradients. This new method satisfied all of the proposed axioms and was shown to produce useful visualisations of explanations in object recognition networks and neural machine translation, for example.

The integrated gradients method proposed in Sundararajan et al. (2017) avoids saturation due to its mathematical formulation. To calculate integrated gradients first a baseline has to be chosen, this is to represent an input that contains no signal or no information. In the context of image classification, a completely black image is used. Interpolation is then performed from the baseline to an actual input to train the model, with a set number of steps in interpolation. The predictions are then calculated for each of the steps in interpolation, and then the gradient of these predictions with respect to the input are calculated. Finally, these gradients are integrated over a straight line path from the baseline to the input, which avoids problems with local gradients being saturated. Mathematical details of integrated gradients can be found in Section 3.2.1.

2.4.4 Attribution Priors

Ross et al. (2017) built upon these attribution methods and proposed using the explanations created to train non-linear models, which was the first occurrence of attribution priors. This was achieved by regularising the loss function at every training step, which would penalise certain inputs that had low attribution scores in the final output. To penalise certain inputs, a binary mask using a matrix A would have to be specified, which would be used to indicate the relevance of a certain dimension and shrink gradients associated with those that were irrelevant. The authors of this paper made the limitations of this research clear, including how understanding the meaning of inputs and input gradients would be difficult for users, as well as specifying the masking matrix A as being a whole challenge in itself. Other works have demonstrated applications of attribution priors, whereby the attributions are incorporated into model training, and have shown this to be a successful method in areas like natural language processing (NLP) (Liu and Avci 2019).

2.4.5 Expected Gradients

The integrated gradients method proposed in Sundararajan et al. (2017) uses baselines as a reference to calculate the attribution score for each input value. These baselines are hyperparameters that users of the model have to choose appropriately for the problem domain that the model is being used in. In the case of image classification, a baseline of a black image is typically used to represent an input of no signal.

The work done in Erion et al. (2021) demonstrated that this choice of baseline as a hyperparameter could be obfuscated using their proposed method that built upon integrated gradients, called expected gradients. The proposed attribution method obtains baselines and the step in interpolation using sample based approximation. Similarly to integrated gradients, it manages to satisfy the axioms defined in Sundararajan et al. (2017). It was shown in this research that the attributions produced were comparable to those created by the integrated gradients method and are attainable in much shorter execution times. Furthermore, Erion et al. (2021) demonstrated the use of the attributions derived from expected gradients method as an attribution prior to train models and achieve high performances in image classification and biological prediction tasks.

Sturmfels et al. (2020) discusses the many problems that revolve around having the choice between many different baselines as a hyperparameter to tune when using integrated gradients. This work goes on to point out that different baselines will lead to different attributions and therefore different visual explanations of a models decision making process.

2.5 Building upon this previous work

No current research exists on using both integrated gradients and expected gradients in attribution prior methods to train models in the timeseries classification of ECG heartbeats. It is for this reason that this project aimed to develop novel models to achieve this and therefore investigate the ability of the attribution methods in training deep learning models.

2.6 Chapter Summary

In this chapter an overview of the problem domain was presented, as well as previous works that have been shown to produce state-of-the-art results in deep learning classification of heartbeats. Concepts as to how users have better understanding of model decision making was also covered, alongside existing methods used parallel to deep learning to assist this. Finally how these explainability methods can be used to improve the performance of deep learning models was discussed, for example Expected Gradients as an attribution prior.

3 Methods

3.1 Machine Learning Models used in this Project

3.1.1 SVM

The SVM used in this project was implemented using scikit-learn's C-Support Vector Classification (Varoquaux et al. 2015). SVMs are designed to perform binary classification, where the model has to find the decision boundary or hyperplane between two classes in a dataset. However, this project requires a multi-class classification SVM for all of the eight different heartbeat classes (see Figure 2.1). Therefore, 'ovo' was passed into the *decision_function_shape* parameter of the SVM created using scikit-learn, so that a 'One-vs-One' classification strategy was adopted. This works by splitting the multi-class classification of the eight classes into a number of binary classifications, specifically 28 binary classifications (Brownlee 2020). No other hyper-parameters were tuned for this SVM model, the default parameters provided from scikit-learn which included an RBF kernel, were used.

3.1.2 Convolutional Neural Network

The CNN used in this implementation was developed using Keras. The architecture of this model is summarised in Figure 3.1, the image of this architecture was produced using Netron (Roeder 2020). The input layer takes as input any number of beats and passes this input onto the first convolutional layer. The first two convolutional layers contain Rectifier Linear Unit (ReLU) activation functions and are followed by batch normalisation layers. The output of the second batch normalisation passes into the final convolutional layer which also contains a ReLU activation function and is then passed onto a max pooling layer. Then, this max pooled output is flattened and then passed into three fully connected dense layers that contain ReLU activations and finally to a last fully connected dense layer. The output of this last fully connected dense layer then has a softmax applied to it to produce the final classification output.

3.2 Gradient Based Explainability Methods

3.2.1 Integrated Gradients

As previously mentioned, integrated gradients in image classification typically uses a black image as a baseline to represent a zero-signal input. The baseline for this project, a baseline that referenced no signal or a 'flatline' was used, equivalent to an all zero amplitude ECG beat. Below is the formula presented in Sundararajan et al. (2017) for integrated gradients:

$$IG(x, x') = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial f((1-\alpha)x' + \alpha x)}{\partial x_i} \, d\alpha. \tag{3.1}$$

This equation is shown to be intractable in the work of Sundararajan et al. (2017) and therefore a summation can used to approximate the integral when it comes to implementing Integrated

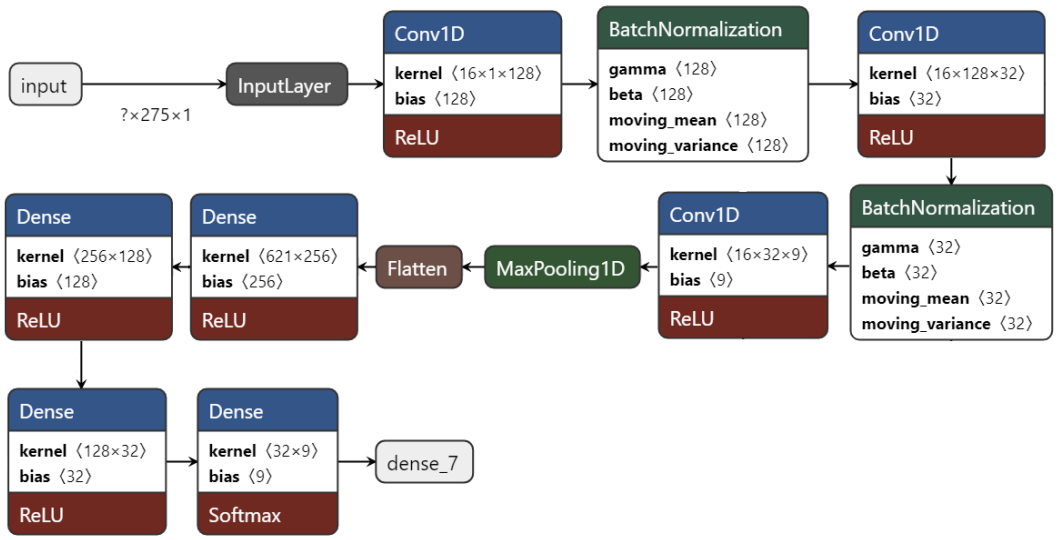


Figure 3.1: *Convolutional Neural Network Architecture, created using Netron.*

Gradients. The Riemann approximation is suggested to be used by the authors of this work, and the approximation equation can be shown below:

$$IG_{riemann}(x, x') = (x_i - x_i') \times \sum_{k=0}^{m} \frac{\partial f((1 - \frac{k}{m})x' + \frac{k}{m}x)}{\partial x_i} \times \frac{1}{m}, \tag{3.2}$$

where m is equal to the number of interpolation steps, therefore the number of gradients calculated, and $k = (1, 2, ..., m)$. However, in the implementation of Integrated Gradients used in this project, a simple Trapezoidal approximation was used instead, based on this equation:

$$Area = \frac{h}{2}\Big(a + b\Big). \tag{3.3}$$

In this use case, h, the height, would be equal to one, since the timeseries data increments by one at each consecutive point. The values a and b in this equation represent adjacent gradients, calculated at each interpolation step:

$$IG_{trap}(x, x') = (x_i - x_i') \times \sum_{k=0}^{m-1} \frac{1}{2}\Big(\frac{\partial f((1 - \frac{k}{m})x' + \frac{k}{m}x)}{\partial x_i} + \frac{\partial f((1 - \frac{k+1}{m})x' + \frac{k+1}{m}x)}{\partial x_i}\Big) \times \frac{1}{m}. \tag{3.4}$$

The original integrated gradients paper (Sundararajan et al. 2017) recommended using $m \in \{20, \ldots, 300\}$ steps in the approximation of the integral. It was pointed out in Liu and Avci (2019) that training a model using integrated gradients with this many steps can take up to 30 times longer. Considering that training times were already quite high for the baseline CNN model, it would take even longer to train with a large number of interpolation steps in the approximation. Due to time constraints, the decision was therefore made to choose $m = 4$, ensuring not only that training overhead remained reasonably low, but also that the number of interpolated gradients could still be averaged over.

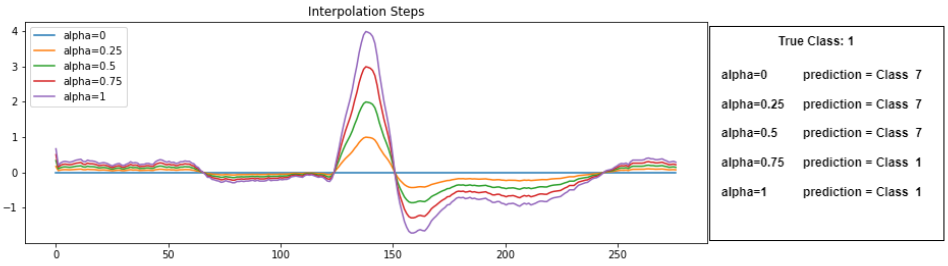


Figure 3.2: *Demonstration of how a model's classification prediction changes as interpolation occurs from the baseline to normal input*

Implementation of integrated gradients in this project was achieved by altering the code provided in the Keras tutorial for integrated gradients to only perform five interpolation steps and to also specifically handle one-dimensional ECG timeseries data.

3.2.2 Expected Gradients

A method to avoid the choice of baselines to produce attributions in a similar way to integrated gradients is proposed in Erion et al. (2021) called expected gradients. Avoiding specifying a baseline can be done by calculating integrated gradients over multiple baselines belonging to the same distribution, for example the uniform or gaussian distributions. The paper demonstrates that this is equivalent to simply integrating over a baseline distribution D:

$$Gradients(x) = \int_{x'} \left((x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial f((1-\alpha)x' + \alpha x)}{\partial x_i} \, d\alpha \right) p_D(x') dx'. \tag{3.5}$$

As integrating over a distribution D is usually intractable, the researchers in the paper changed the calculation to be reformulated as an expectation:

$$ExpectedGradients(x) = \underset{x' \sim D, \alpha \sim U(0,1)}{\mathbb{E}} \left[(x_i - x_i') \times \frac{\partial f((1-\alpha)x' + \alpha x)}{\partial x_i} \right]. \tag{3.6}$$

The work done by Erion et al. (2021) demonstrated that using a sampling based approximation method to sample baselines from the training dataset and to sample the alpha value used in interpolation, provides a method for calculating the expectation for each sample, and through that an average over all samples.

When it came to implementing expected gradients in this project, the modules provided in the repository associated with the original expected gradients paper (Erion 2022) were modified so that the appropriate functions could process the timeseries ECG data instead of 2D images.

3.3 Attribution Priors

The models to be used in evaluation will be based on the CNN architecture mentioned above. However this model will be modified to incorporate attribution priors into its training, to evaluate how the performance is affected. To obtain the attributions, we will be using both the Integrated and Expected gradient functions (see equations 3.1 and 3.6). The next step involves applying these to the training of the models as attribution priors.

Optimal parameters are found in deep learning using this standard equation:

$$\theta = \text{argmin}_\theta \mathscr{L}(\theta; x, y) + \lambda' \Omega'(\theta). \tag{3.7}$$

Where $\Omega'(\theta)$ is a penalty function regularised by a scalar λ' term. To incorporate attributions in minimising the training loss the penalty function has to take these as parameters, so that our new optimal parameters can be found with this new equation, using attribution priors:

$$\theta = \text{argmin}_\theta \mathscr{L}(\theta; x, y) + \lambda \Omega(\Phi(x)). \tag{3.8}$$

The penalty function Ω can take many forms and is the one of the independent variables within the experiments performed in this project. In Erion et al. (2021) the attribution prior function used for image classification, called Ω_{pixel}, was based on looking at the differences between adjacent pixels. Neighbouring pixels in an input image should intuitively have similar attributions towards the final prediction score, and this is also true for neighbouring or adjacent timeseries data points in an ECG recording. It is for this reason that total variation Rudin et al. (1992) was used as a basis for these penalty functions. Both equations 3.9 and 3.11 are direct implementations of total variation between attribution scores for adjacent timeseries datapoints in a heartbeat. Since both of these equations resembles that of an L1 vector norm calculation of the differences between adjacent attributions, the two models IG_L1 and EG_L1 using these priors had L1 appended to their name.

It is possible that this one type of attribution prior function, resembling an L1 norm, may not be particularly well suited to the problem domain of ECG beat classification. This resulted in the use of another type of attribution prior function to investigate, which resembled an L2 norm of the differences between adjacent attributions (see equations 3.10 and 3.12).

The baseline CNN used in this project was implemented using Keras. However, to incorporate attribution prior penalty functions into training the function typically used in training the Keras model had to be overridden. The overridden train step implementation was based on the ones used in the respository linked to the expected gradients paper (Erion 2022) and changed accordingly to accommodate the different attribution prior penalty functions proposed above. To ensure that the baseline CNN was as close as possible to the novel models, it too was implemented using the overridden train step. However, no attributions were calculated and used to regularise its total loss in training.

$$\Omega_{IG\_L1}(\Phi(x, x')) = \sum_b \sum_i \left| \Phi^b(x_{i+1}, x'_{i+1}) - \Phi^b(x_i, x'_i) \right| \tag{3.9}$$

$$\Omega_{IG\_L2}(\Phi(x, x')) = \sum_b \left(\sum_i \left| \Phi^b(x_{i+1}, x'_{i+1}) - \Phi^b(x_i, x'_i) \right|^2 \right)^{\frac{1}{2}} \tag{3.10}$$

In these above integrated gradient based attribution prior penalty functions, both an input and a baseline are parameters, the baseline being zero amplitude, flatline, heartbeat. In the case of expected gradient based attribution prior functions shown below, a baseline is omitted.

$$\Omega_{EG\_L1}(\Phi(x)) = \sum_b \sum_i \left| \Phi^b(x_{i+1}) - \Phi^b(x_i) \right| \tag{3.11}$$

$$\Omega_{EG\_L2}(\Phi(x)) = \sum_b \left(\sum_i \left| \Phi^b(x_{i+1}) - \Phi^b(x_i) \right|^2 \right)^{\frac{1}{2}} \tag{3.12}$$

In all of the above attribution prior penalty functions, the $\Phi^b(x_i)$ refers to the attributions calculated for the ith timeseries data point in the bth ECG heartbeat of the input. In the implementation of these attribution prior penalty functions, the loss function is updated per batch, so every input into the functions shown above will be a batch of beats.

3.4 Technologies Used

The majority of this project was developed using Python 3.8.3, Tensorflow 2.7.0, numpy and sci-kit learn (Varoquaux et al. 2015) on an HP Laptop with an Intel i5 processor, Nvidia GTX 1050 GPU and 8GB of RAM. Since training times are significantly long when using deep neural networks, GPU acceleration was required to be able to achieve results in a reasonable time frame. Therefore, Nvidia's CUDA Toolkit was used alongside the CuDNN package to speed up training in tensorflow. When it came to evaluating the models, the HP laptop was used for the experiment involving beat holdout, however the experiment for patient holdout had significantly more data. The train-test split of the beat holdout experiment was 31912 and 37863 beats, whilst the train-test split for the patient holdout experiment was 206312 and 14380 beats. Due to the large amount of data and memory requirements during training for the patient holdout experiment, an Ubuntu server with four Nvidia RTX A5000 GPUs with GPU acceleration was used.

3.5 Data Pre-Processing

The WDFB package (a20 2022) was used to read records from the MIT BIH dataset. The way this package works is that it can extract both the ECG signals and more importantly the ECG annotations associated with them. This made all the different classes of heartbeats significantly more accessible.

As the models used in this project were to be trained on individual heartbeats, the standard method to extract individual heartbeats was used, finding the R peaks (see Fig 2.1). The package biosspy (bio) was used to find the R peaks and match them with their corresponding annotation. Additionally, its corresponding numerical class value that would be used throughout implementation was appended to the end of each. Generally the annotation would be near or at the index of the data point that equated to the R peak in the heartbeat, making it necessary to search for the annotation in a small range around the R peak. It was important to keep the amount of data points contained in each beat the same, it is for this reason that the beats were centered around the R peak and an equal amount of data points were taken on either side of this peak. Alongside appending the numerical value associated with the heartbeat's class (see Fig 2.1) it was also necessary to append the patient ID number, so that two experiments using beat holdout or patient holdout, could be conducted. Upon visualising the heartbeats, it became apparent that there were inconsistencies and irregularities in the amplitudes, therefore it was necessary to standardise each beat by subtracting the mean of each beat's data and then dividing that by the standard deviation of each beat's data.

3.6 Class Imbalances

The visualisation of the distribution of classes can be seen in Fig 3.3. From this figure alone it is clear that re-sampling had to be performed, to even out the class distribution with the class for normal beats overwhelmingly being the majority. Discussion about the effect of imbalanced or balanced data in classification problems is presented in Wei and Dunbrack (2013). This work concludes that undersampling the majority class can be a simple solution if large amounts of the minority class can be obtained. In the case of using the MIT BIH dataset this is not the case, consider the F class in 3.3 which has far too few beats. However, two other works discussed

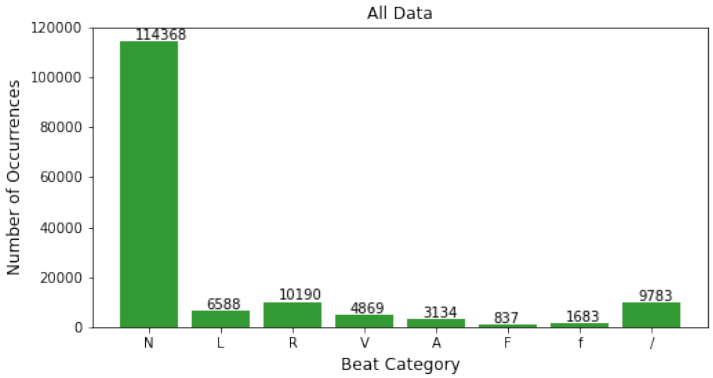


Figure 3.3: *Distribution of the eight heartbeat classes in the whole dataset*

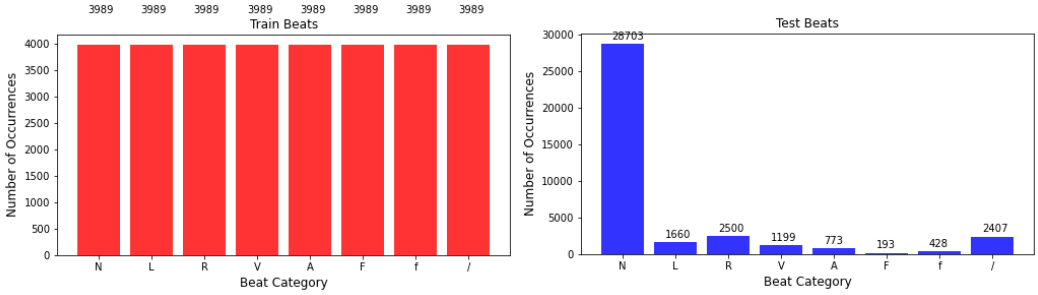


Figure 3.4: *Distribution of heartbeat classes in beat holdout dataset after re-sampling*

in this paper demonstrate that a mix of both upsampling and downsampling can produce high accuracy scores in a classification problem (Chawla et al. 2002), demonstrating this method to be appropriate for the context of this project.

Before re-sampling could be performed, it was crucial to split the data into train and test sets. This is required as it is highly likely that if re-sampling is performed before the split, that data could leak between both the train and test split. In turn this would lead to unreliable results upon evaluation of models on this data. To separate the data into a 0.75/0.25 train-test split for use in the beat holdout experiment, the scikit-learn train_test_split function was used (Varoquaux et al. 2015). On the other hand, separating the data into a 0.9/0.1 train-test split for the patient holdout experiment was performed manually. Patients 104, 208, 113, 210 and 119 were left out from the train set and used in the test set because of the interesting distribution of beat classes their records

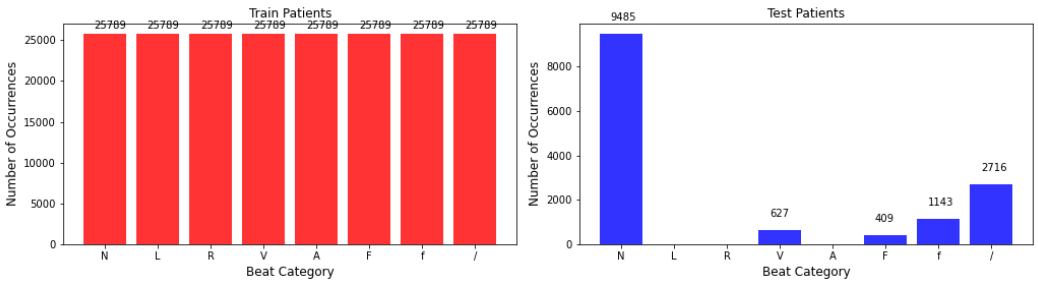


Figure 3.5: *Distribution of heartbeat classes in patient holdout dataset after re-sampling*

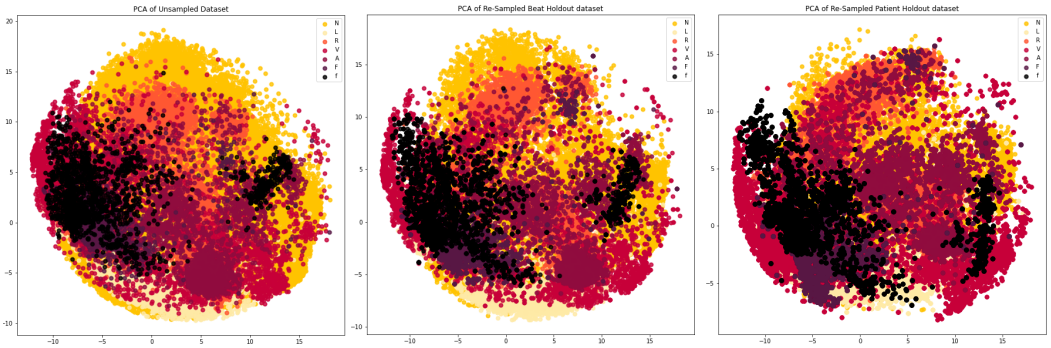


Figure 3.6: *Distribution of heartbeat classes in patient holdout dataset after re-sampling using Principal Componenent Analysis.*

contained.

The *resample* function provided by the scikit-learn python package was used to produce both the beat holdout and patient holdout datasets. This method utilises a part of the bootstrap procedure commonly used in statistics for resampling, proposed in Erion et al. (2021). The part of the bootstrap procedure used by the scikit-learn resample function iteratively resamples from a given dataset whilst using replacement. The mean number of beats in the abnormal classes was calculated and used as the amount of samples to be used across the classes. For each class set, if the number of beats was above the population mean then downsampling was applied. If the number of beats was below the population mean then upsampling was applied using replacement provided in the scikit-learn *resample* function. The results of the re-sampling for both the beat holdout and patient holdout datasets can be found in bar charts presented in Figures 3.4 and 3.5. Another method to visualise the distribution of classes is through dimensionality reduction using Principal Component Analysis (PCA) (Jolliffe 2011). The results of PCA are presented in Figure 3.6 and the decrease in the N class can be seen visually in the re-sampled beat and patient holdout datasets.

3.7 Hyper-parameter Tuning

Before evaluation was performed on the different models, the optimal hyper-parameters for the baseline CNN needed to be found so as to ensure that high performance metrics would be attained across models. Since the number of epochs that a model is trained over, is a hyper-parameter that when changed to high values results in incredibly long training times, the optimal number of epochs was estimated from plotting how the training and validation accuracy changed with epoch number. The baseline CNN model used investigate the epoch number was compiled with an Adam optimizer and a categorical cross-entropy loss function, whilst being fit to the beat holdout training dataset in batches of 64. After ten epochs, the training and validation accuracy seemed to converge, therefore this was used as the number of epochs in a grid search to find the rest of the optimal hyper parameters (see Figure 3.7).

Grid search was performed over the beat holdout training dataset using scikit-learn's GridSearchCV function. The hyper parameters investigated were the activation function and kernel initialisation used in the first dense fully connected layer in the network, the dropout rate used by a dropout layer after this first fully connected layer and the optimizer that the model was compiled with. Each of the 81 hyper parameter combinations was run for three folds, resulting in 241 fits for the grid search which took approximately 24 hours to run. The optimal hyper-parameters for the baseline CNN model was a ReLU activation and uniformly initialised mode for the kernel

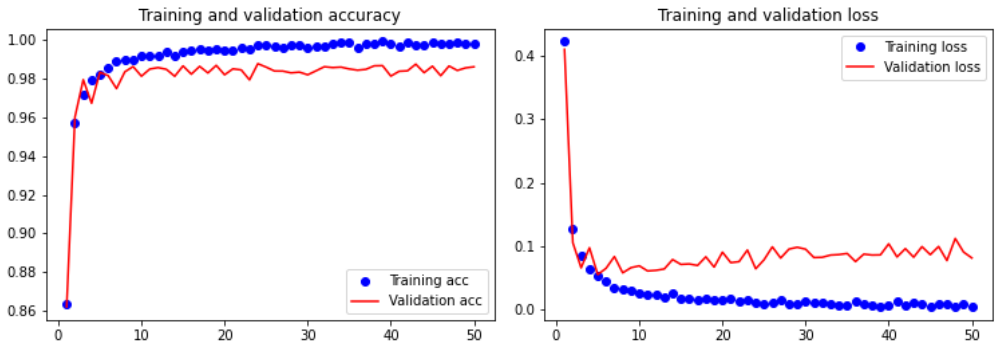


Figure 3.7: *Effect of epoch number on accuracy and loss of baseline CNN model*

in the first dense layer, a Nadam optimizer and a zero dropout rate. A final grid search was run to investigate the combination of different dropout rates and kernel weight constraints on the first two convolutional layers and the first dense fully connected layer. The results showed that a weight constraint of five and dropout rate of zero were optimal, this indicated that a dropout layer was not needed in the models used for evaluation.

A final hyper-parameter that was investigated to be optimised was the lambda value used to regularise the attribution prior penalty function in model training (see Equation 3.8). Using the standard GridSearchCV function mentioned above was not possible in this scenario. This is due to the fact that finding the optimal lambda values for these novel models would require training with the overridden *@train_step* adapted from the repository associated with the work in Erion et al. (2021). Therefore, a basic manual grid search was performed by performing 3 Fold Cross Validation over a number of candidate lambda values for each of the four novel models and tracking the mean accuracy and loss over each fold. Four candidate values were investigated for the IG L1 and EG L1 models and four other candidates were investigated for the IG L2 and EG L2 models. It was found from early experimentation that the L2 models generally performed better with larger lambda values than the L1 models. The 'optimal values' found from the manual investigation was a lambda value of 7.0×10^{-6} for the L1 models and a lambda value of 1.1×10^{-4} for the L2 models.

3.8 Chapter Summary

In this chapter, the models used in the evaluation and their implementations were presented, including the explanation of the mathematics they were based on. The steps used to pre-process the data were also discussed, and how due to an imbalance in the heartbeat classes in the MIT-BIH dataset, a resampling method had to be applied so that a balanced set was used for training of the models. Finally, the grid search procedure and its use in finding optimal hyper parameters for the model was shown.

4 Evaluation

Two experiments were performed in this study, the beat holdout experiment and the patient holdout experiment. The former involves evaluating models abilities to generalise to an unseen set of shuffled beats whereas the latter evaluates how well the models performed on an unseen set of patients. These experiments were used to compare the performance of the proposed novel models to a baseline CNN and SVM. Four performance metrics were used in the quantitative evaluation of the classifier models in this study. These were accuracy, precision, recall and F1 macro which were all implemented using scikit-learn. Details about each and their equivalent equations are presented in Section A.1.

4.1 Beat Holdout Experiment

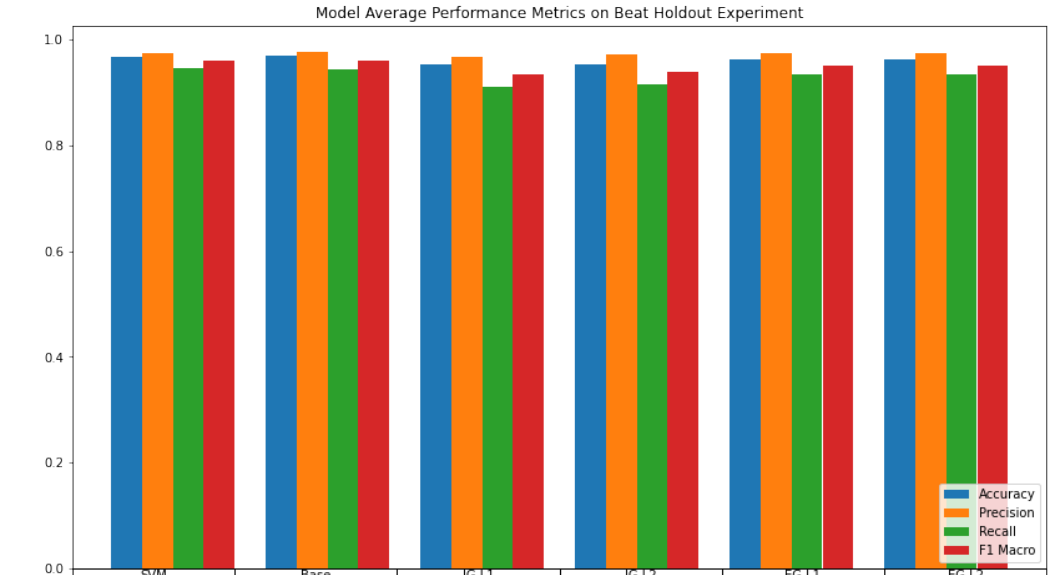
4.1.1 Repeated Stratified K-Fold Cross Validation

The re-sampled beat holdout data set mentioned in Section 3.6 was evaluated upon by the six models in Stratified K-Fold Cross Validation. Cross Validation is commonly used in statistical model validation and hence why it is a popular validation method for machine learning models (Refaeilzadeh et al. 2009). It works by splitting the entire dataset into a different training and testing set over ten folds to evaluate how well each of the models generalises to differing test sets of unseen data repeatedly. Stratified K-Fold Cross Validation is a specific type of Cross Validation, whereby for each fold the number of classes in each is evenly distributed, this has been shown to be more appropriate in classification problems when models can become more biased towards a highly represented class in a given fold (Diamantidis et al. 2000; Kohavi 1995). After each fold, metrics such as accuracy, precision, recall, and F1-macro were calculated, then the model is reset to be re-trained on a new train-test split in the next fold. To increase the statistical power of this experiment, this Stratified 10-Fold Cross Validation was repeated five times for each model, the average execution time for each model was around 15 hours. The mean of every metric was calculated over each of the ten folds, resulting in five sets of each metric for each repeat.

4.1.2 Performance Results for Each Model

In Fig 4.1 the mean score of every metric achieved by each model is displayed. On average it appears from this experiment that all of the models seemed to achieve similar scores for each metric, as all of the mean scores lie between 0.91 and 0.98. The bar graphs for both of the integrated gradient based models are noticeably smaller than those belonging to all of the other models, this can be confirmed quantitatively by comparing the values in the table below the barchat. Overall, this graphic mostly indicates to use that all of the models achieved very high scores for all performance metrics in this experiment.

Fig 4.2 provides a closer look at the differences between model performances in the form of boxplots for the average scores of accuracy, precision, recall and F1 macro. Immediately, both of the models based on Integrated Gradients, IG L1 and IG L2, perform significantly worse in all metrics than the models. The gap in scores between the IG models appears to be slightly bigger for recall and F1 macro than for accuracy and precision. Both of these models are trained on



	SVM	Base	IG L1	IG L2	EG L1	EG L2
Accuracy	0.9677	0.9687	0.9525	0.9532	0.9631	0.9636
Precision	0.9733	0.9776	0.9684	0.9711	0.9751	0.9754
Recall	0.9468	0.9445	0.911	0.9145	0.9334	0.9337
F1 Macro	0.9591	0.9592	0.9346	0.9381	0.9514	0.9516

Figure 4.1: *Barchart of each Models' Mean Performance Metrics for the Beat Holdout Experiment.*

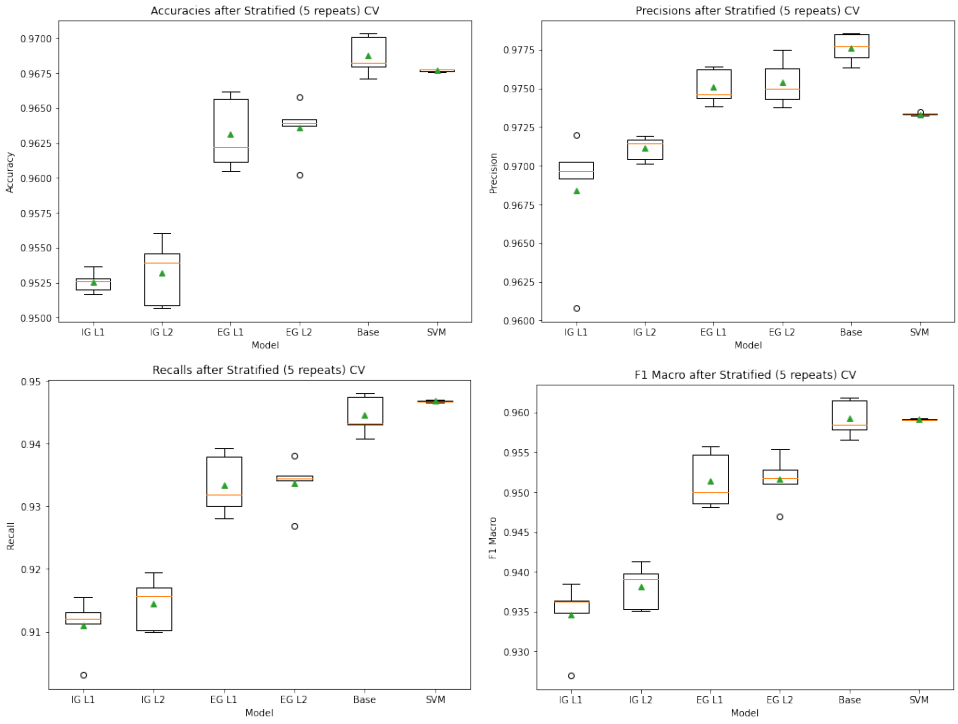


Figure 4.2: *Box Plots of Model Metrics after 5 Repeat 10 Fold Stratified Cross Validation.*

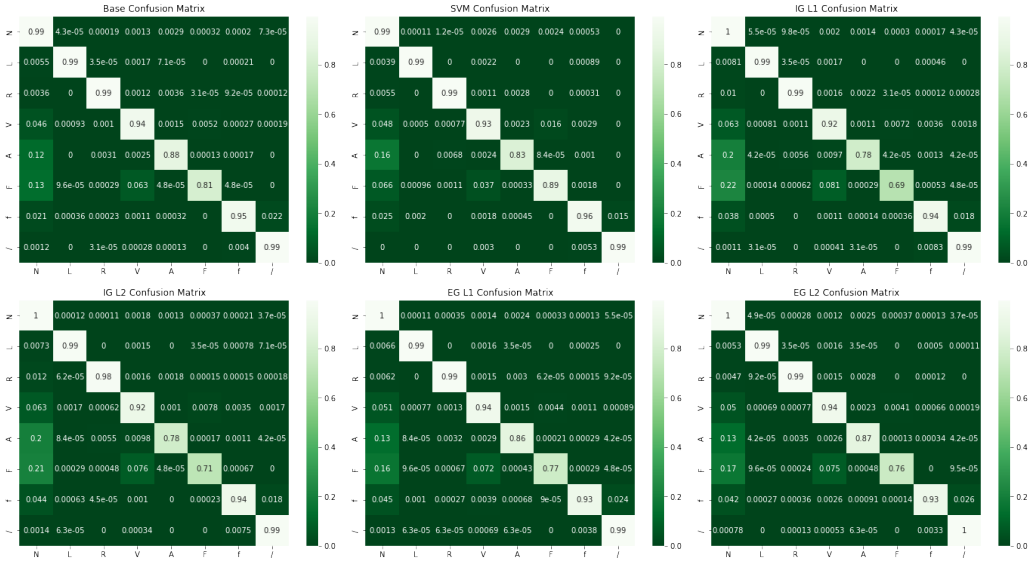


Figure 4.3: *Means of Confusion Matrices after 5 Repeat Stratified 10 Fold Cross Validation for each model for each class. The x-axis represents the predicted labels and the y-axis represents the true labels.*

the same types of attributions and this is reflected in how close their scores are for each metric. It appears that the IG L2 overall performs better than IG L1 model, with the mean of the IG L2 consistently being higher than that of the IG L1. The models based on expected gradients, EG L1 and EG L2 both performed better than the integrated gradient based models and also performed very similarly across metrics. It appears that EG L2 was generally better as a classifier in the beat holdout experiment, as its mean scores for each metric was consistently higher than those of the EG L1 model. None of the attribution prior based models performed as well as the baseline CNN model, where across all metrics apart from recall and F1 macro, it performed the best. Variation in the results of the SVM were incredibly low, which can be seen in the very small and narrow box shown for this model over each metric. Despite the SVM performing better than both the EG L1 and EG L2 models in all other metrics, it failed to achieve higher average precision scores than these two models.

To examine how many of the beat classes were correctly classified into their respective classes, confusion matrices were also calculated at each fold, and the average over every fold and repeat for each model is presented in 4.3. The x-axis represents the predicted labels and the y-axis represents the true labels. Across all models, the classification performances were exceptionally high for the N, L, R and / classes. The diagonals of each confusion matrix are very clearly defined which means overall there was a minuscule amount of mis-classfication by all of the models. As expected, the IG L1 and IG L2 models performed worse than the rest, on classes with much less distinctive characteristics such as the A and F classes. More detail about the metric scores of each model achieved on each class can be seen in Fig A.1. Overall, each model had very low precision, recall and F1 macro scores for each of the classes, but significantly high accuracy.

4.1.3 Quantitative Evaluation - Statistical Analysis

The Wilcoxon Signed Rank Test (Demšar 2006) is used to measure whether the distributions of two samples are equal or not and Fig 4.4 shows the results of this statistical test. The Wilcoxon Signed Rank Test was calculated for every combination of models and displayed in the confusion matrices. Intuitively, the diagonal value of the confusion matrices are all one, since exact copies of a distribution are compared. For all of the metrics, the null hypothesis is rejected because

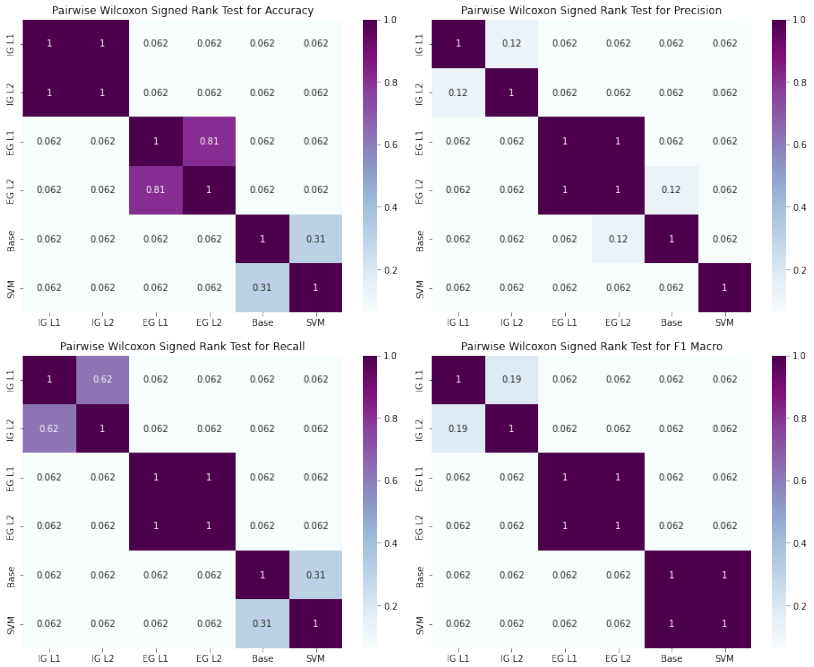


Figure 4.4: *Pairwise Wilcoxon Signed Rank Tests per Model.*

all of the p values are greater than 0.05. This result means that the results of each model for each of these metrics per fold all come from the same distribution. Rejecting the null hypothesis enforces that the results obtained for this experiment are statistically significant. The Friedman Test (Pereira et al. 2014) was also used to test the null hypothesis that all of the means of each population were equal. After running the Friedman Test on all mean metric results for every model, the p-value returned was 0.0029 for the beat holdout experiment. This meant that the null hypothesis is rejected and there is a statistically significant difference in metric results for the models. To further investigate, the Nemenyi test was used to identify which of the models had different means. This was implemented using scikit's posthocs package and the results of this test can be summarised in Figure 4.5. From this table we can conclude that the only statistically significant differences in means is that of the baseline CNN and the IG L1.

	IG L1	IG L2	EG L1	EG L2	Base	SVM
IG L1	1.000000	0.900000	0.526622	0.136905	0.004452	0.052161
IG L2	0.900000	1.000000	0.900000	0.526622	0.052161	0.298439
EG L1	0.526622	0.900000	1.000000	0.900000	0.410222	0.854075
EG L2	0.136905	0.526622	0.900000	1.000000	0.854075	0.900000
Base	0.004452	0.052161	0.410222	0.854075	1.000000	0.900000
SVM	0.052161	0.298439	0.854075	0.900000	0.900000	1.000000

Figure 4.5: *Nemenyi Test results from mean model performance metrics on the beat holdout experiment.*

4.2 Patient Holdout Experiment

This experiment utilised the re-sampled patient holdout dataset discussed in 3.6. The rationale behind holding out all heartbeats from a a set number of patients is presented in Jones et al. (2020). In this work, it is argued that in the normally used beat holdout method, results from experiments are affected by data bleeding between the training and testing sets which in turn leads to performance scores for models being higher than they should be. For example, when using the normal beat holdout method, the train-test split will cause half of Patient Y's beats to be in the train set and the other half to be in the test set. It is highly likely that the morphologies of beats of the same patient are very similar, and will result in a model being evaluated on a test set for which it has effectively already 'seen' some of the data. Therefore, the researchers in Jones et al. (2020) proposed that the test set should contain the heartbeats belonging to a select number of patients, to avoid the effect of data bleeding.

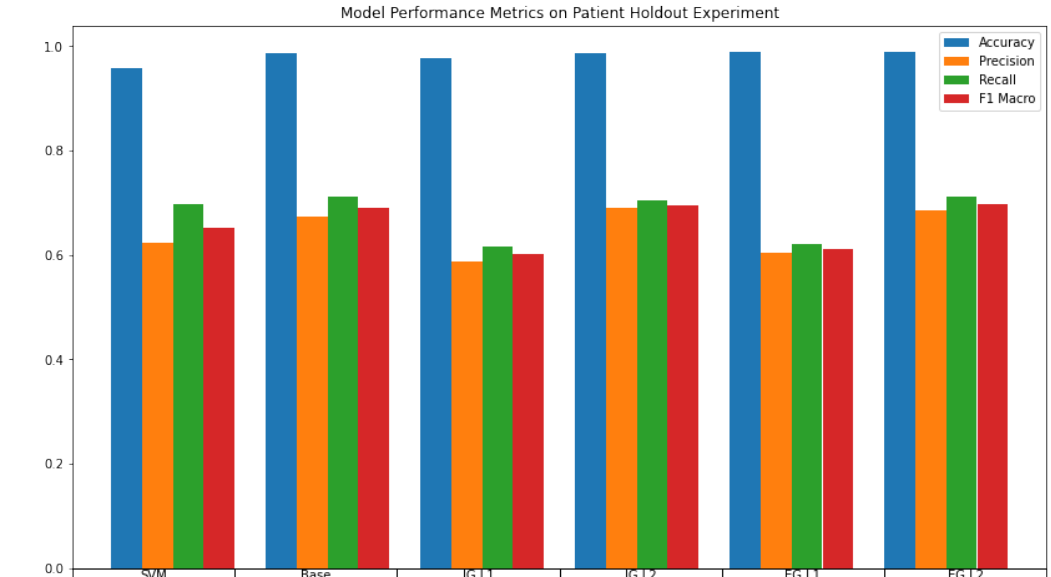
The testing set contains the heartbeats belonging to patients 104, 113, 119, 208 and 210 and the training set contains the heartbeats from the rest of the 43 patients. In Jones et al. (2020) these specific patients were also held out from the training dataset, due to the distribution of beat classes in these patients being varied enough to test models on whilst ensuring that overall the distribution of beat classes in the training dataset was not severely affected. Finally, this experiment also differs from the previous beat holdout experiment as repeated stratified 10-fold cross validation wasn't performed. This is due to the fact that the stratified 10-fold cross validation method used in the previous experiment evaluates the model on a new train-test split of the beat holdout data for each fold. However, this would not work alongside the patient holdout method, as the test set cannot change from containing the heartbeats from the patients listed above.

4.2.1 Performance Results for Each Model

The performances of each model for this experiment are displayed in Fig 4.6. The accuracy scores achieved by each model are considerably higher than precision, recall and F1 macro. It appears that the shallow machine learning model, the SVM, has the worst accuracy score of around 0.958, compared to the rest of the deep learning models which have accuracies ranging from around 0.978 to 0.99. With respect to the precision, recall and F1 macro scores the SVM performs beter than the IG L1 and EG L1 models, which both have the lowest scores in all of these metrics. The baseline CNN, the IG L2 and the EG L2 all have very comparable scores for every metric in the patient holdout experiment.

The confusion matrices in Fig 4.7 show the how well each model classified each class correctly. Similarly to the beat holdout experiment, the x-axis of each confusion matrix shows the predicted class and the y-axis shows the true class. Note that there are no results for the L, R, or A classes, as these are not present in the patients that were heldout in the test set. There is very little differences in these results, which indicates that there was very little mis-classification done by all of the models. The only noticeable differences are how the SVM mis-classified the V class (Premature Ventricular Contraction) and the IG L2 also having a slightly lower classification rate for the f class (Fusion of Paced and Normal Beat). A further detailed insight into how each model performed for each metric per class is displayed in Fig A.2. The nan values and 0.0 values mean that no metrics were calculated for the classes that weren't present in the test set. Overall, it appears that the models in the patient holdout experiment performed better per class than in the beat holdout experiment in the precision, recall and F1 macro scores. Notably, all of the models had incredibly high scores, ranging from around 0.7 to 1, with regards to the F class (Fusion of Ventricular and Normal). Across models, there are no outstanding differences in the scores achieved per class apart from the expected gradient based models having perfect scores for every metric.

Statistical analysis like the pairwise wilcoxon signed rank test was not possible on the results



	SVM	Base	IG L1	IG L2	EG L1	EG L2
Accuracy	0.9579	0.9871	0.9778	0.986	0.9896	0.9894
Precision	0.6241	0.6728	0.5881	0.6889	0.6036	0.6843
Recall	0.6979	0.7104	0.615	0.7036	0.6213	0.7106
F1 Macro	0.653	0.6897	0.6007	0.6958	0.6122	0.6967

Figure 4.6: *Barchart of each Models' Performance Metrics for the Patient Holdout Experiment.*

from this experiment as it was not repeated. However it is possible to perform the Friedman and Nemenyi test on the performance metrics for each model. The Friedman Test performed was similar to that used on the beat holdout experiment results, where the null hypothesis was that the means of each performance metric for all models was equal. The test on these results yielded a p-value of 0.0186 which means the null hypothesis is rejected and there is statistically significant differences between the models' mean metric results. The results of the Nemenyi test can be found in Figure 4.8 and this test allows us to conclude that the only statistically significant differences in model performance metrics is that of the EG L2 and the IG L1 models.

4.3 Explainability Results

All of the explainability methods investigated in this project mainly use heatmaps when it comes to their visualisation, traditionally to visualise explainability on images, two-dimensional heatmaps are used. To compare the integrated gradients, expected gradients and GradCAM methods, one-dimensional heatmaps were produced for the attributions of correctly and incorrectly classified heartbeat classes. The heatmaps produced using the attributions from integrated gradients can be found in Figure A.3. In the correctly classified heatmaps, there is generally very little noise overall and the 'hotter' parts of the map all correspond to the area around the R peak, the QRS complex. For the incorrectly classified heartbeats, there is generally more noise, particularly in the classes 2, 3 and 4 which correspond to L, R and V respectively (see Table 2.1). In the heatmaps produced by expected gradients and GradCAM (see Figures A.5 and A.4) the average attributions are incredibly noisy. There is also a lot of similarity between the heatmaps for the correctly and incorrectly classified average attributions.

4.3.1 Beat Holdout

For a more global look at what parts of the heartbeat morphology each of the explainability methods lent more attribution towards, the correct and incorrect attributions across each class

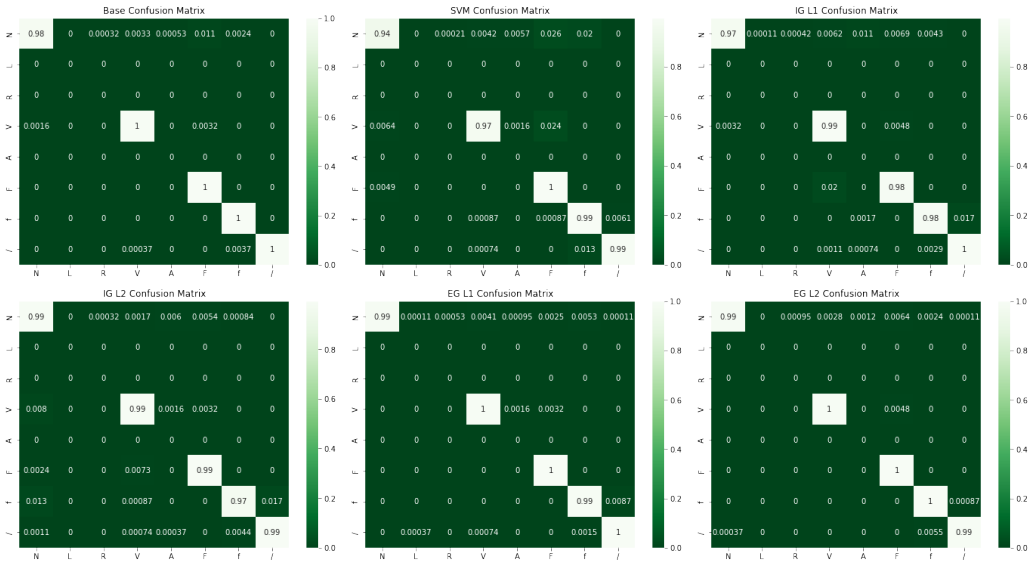


Figure 4.7: *Confusion Matrices for each model for each class. The x-axis represents the predicted labels and the y-axis represents the true labels.*

	IG L1	IG L2	EG L1	EG L2	Base	SVM
IG L1	1.000000	0.136905	0.744925	0.016584	0.207292	0.900000
IG L2	0.136905	1.000000	0.854075	0.900000	0.900000	0.635776
EG L1	0.744925	0.854075	1.000000	0.410222	0.900000	0.900000
EG L2	0.016584	0.900000	0.410222	1.000000	0.900000	0.207292
Base	0.207292	0.900000	0.900000	0.900000	1.000000	0.744925
SVM	0.900000	0.635776	0.900000	0.207292	0.744925	1.000000

Figure 4.8: *Nemenyi Test results from mean model performance metrics on the patient holdout experiment.*

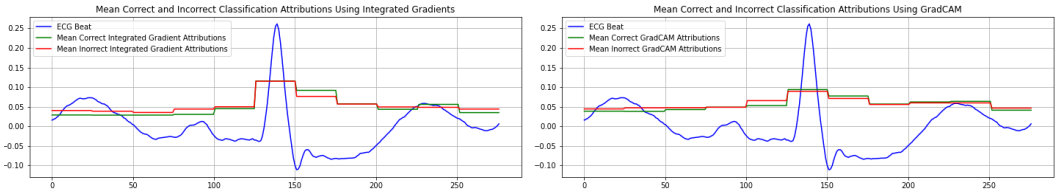


Figure 4.9: *Comparison of average attributions over 11 segments for correct and incorrect classifications produced by both Integrated Gradients and GradCAM using the IG L2 model. This was achieved with results from the Beat Holdout Experiment.*

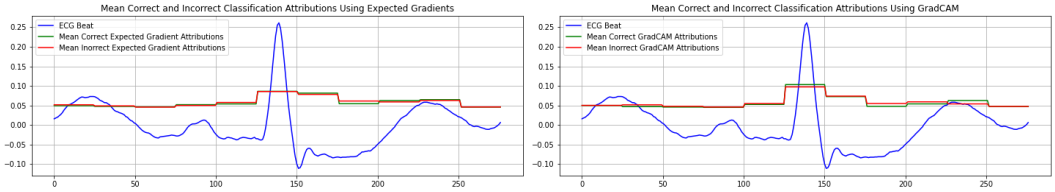


Figure 4.10: *Comparison of average attributions over 11 segments for correct and incorrect classifications produced by both Expected Gradients and GradCAM using the EG L2 model. This was achieved with results from the Beat Holdout Experiment.*

were averaged. These average correct and incorrect attributions were then split into 11 segments and normalised to be displayed over a sample ECG heartbeat. This was performed using the attribution results from the beat holdout experiment, Figure 4.9 displays the attributions achieved with the IG L2 model and Figure 4.10 shows the attributions obtained with the EG L2 model. The sample ECG heartbeat is used as a reference to know generally whereabouts attributions are placed on the beat morphology. In Figure 4.9 it is clear that both the integrated gradients and GradCAM methods place the most importance on the area around the R peak, the QRS complex. In the graph produced using GradCAM there is a lot of similarity between the average correct and incorrect attributions. In Figure 4.10 the average correct and incorrect attributions produced by expected gradients and GradCAM are incredibly similar, in that there is a lot of overlap between correct and incorrect attributions.

4.3.2 Patient Holdout

The segmented average correct and incorrect attributions produced by the three explainability methods for the patient holdout experiment was implemented the same way as for the beat holdout experiment. In Figure 4.11, similarly to the average attributions produced by the integrated gradients method in the beat holdout experiment, there is a clear distinction between the correct

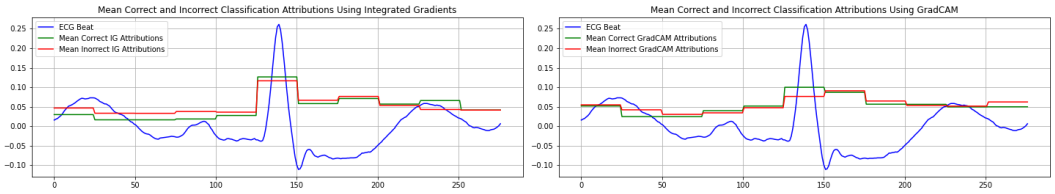


Figure 4.11: *Comparison of average attributions over 11 segments for correct and incorrect classifications produced by both Integrated Gradients and GradCAM using the IG L2 model. This was achieved with results from the Patient Holdout Experiment.*

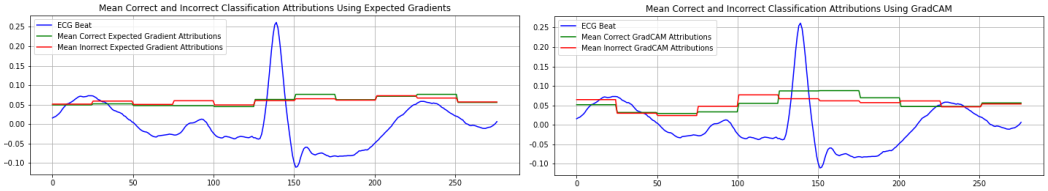


Figure 4.12: *Comparison of average attributions over 11 segments for correct and incorrect classifications produced by both Expected Gradients and GradCAM using the EG L2 model. This was achieved with results from the Patient Holdout Experiment.*

and incorrect attributions. On the other hand, there appears to yet again be more overlap and closeness between the average correct and incorrect attributions produced by the GradCAM method. The incorrect attributions from GradCAM place less importance on the segment that usually corresponds to the QRS complex as opposed to the segment right after it. Figure 4.12 shows that again the average correct and incorrect attributions for the expected gradients are incredibly similar. On the other hand, the average correct attributions from the GradCAM method show much more attention to the QRS complex area of the beat morphology whilst the average incorrect attributions are much more spread out.

5 Discussion and Conclusion

5.1 Summary

This study involved evaluating both the performance and explainability of machine learning models. This was performed via the classification of eight heartbeat classes from time-series ECG data extracted from the MIT-BIH dataset. Six classifiers were evaluated in a 5 repeat stratified cross fold validation using a beat holdout dataset and also evaluated in the patient holdout experiment. Two explainability methods, integrated and expected gradients, were incorporated into the novel models compared to baseline CNN and SVM. These two explainability methods were also compared with the GradCAM method. Discussion about the results from the quantitative and qualitative evaluation will be presented below, followed by the conclusion, limitations and possibilities for future work.

5.2 Beat Holdout Experiment

The low performance of the integrated gradient models is likely due to the requirement of these models to be trained over only three epochs as opposed to ten epochs that the baseline CNN and expected gradient models, EG L1 and EG L2, were trained over. The reason both the integrated gradient models had to be trained over fewer epochs, is because for each epoch there are five gradient calculations at each interpolation step. As a result, training the integrated gradient models for three epochs for each fold in the repeated stratified 10-cross fold validation had an execution time of 16 hours. Increasing the number of epochs to 10 would of resulted in the execution times to around 53 hours for each model, which would not of been feasible considering the time constraints of the project. Another possible reason for the integrated gradient models performing a lot worse than the rest of the models is to do with the number of interpolation steps used in the approximation. In the original paper about integrated gradients (Sundararajan et al. 2017), it was recommended to use anywhere between 20 and 300 steps to approximate the integral. However, even if the lower bound of 20 recommended steps was used this would therefore lead to a complete execution time of around 80 hours if run for three epochs and 260 hours if run for 10 epochs.

It appears as though all of the novel models proposed don't perform as well as the SVM or the baseline CNN, despite the expected gradient models performing only slightly worse across all metrics. One possible explanation could be that the attribution prior penalty functions that these novel models were based on, were not properly suited to the problem domain of timeseries ECG classification. It was argued in Erion et al. (2021) that using total variation as a basis for the penalty function incorporating attributions was decided since attributions should be smooth over neighbouring pixels. This is intuitive as it is likely that adjacent pixels in the x and y axis of an image will only have small differences between them. When developing the novel models in this project it was believed that this too would be the case for timeseries datapoints, as changes between consecutive points in an ECG would also be small. However, the attribution prior models in this paper performing worse than the baseline CNN are an indication that a new penalty function should be investigated further.

Although in this experiment it was shown that the attribution prior based models did not perform as well as the shallow SVM and baseline CNN, this experiment was useful to compare them amongst themselves. The results in Fig 4.2 clearly outline that the both the L2 models generally performed better than their L1 equivalents. The values returned from the penalty functions of the L2 based models are considerably smaller than the values returned from the L1 penalty functions. The magnitudes of the regularising lambda values for these penalty functions found in 3.7 provide supporting evidence for this, as the lambda value for both the L1 penalty functions is 7.0×10^{-6} whilst for the L2 penalty function it is considerably larger as 1.1×10^{-4}. It is possible that higher values returned from attribution prior penalty functions result in worse regularising to the main loss function of a model that hinders model performance. In the code repository connected to the work in Erion et al. (2021), a bigger lambda value of 1.0×10^{-3} was used alongside their expected gradient prior penalty function, which could suggest that the values returned from their attribution prior penalty function weren't incredibly large. If a new penalty function were to be investigated, then trying to keep the values returned from the penalty function small should be an objective.

5.3 Patient Holdout Experiment

One of the most important results from this experiment is how the SVM appears to perform significantly worse in this experiment than for the beat holdout one, when comparing Fig 4.1 and Fig 4.6. The reason that the SVM was so successful in the previous experiment is likely due to the data bleeding effect that is discussed in Jones et al. where beats of the same patient are shuffled and therefore present in both the training and testing datasets. This means that although it is all technically 'unseen' data, the general morphologies will be incredibly similar and have already been trained on. Therefore the decision boundary or the hyperplane created by the SVM in classifying a certain beat will be very effective when encountering a beat morphology in the test set that it has already been trained on. This is not the case in the patient holdout experiment, which could lend to why this shallow machine learning model appears to struggle more than the rest of the models particularly in terms of classification accuracy.

The barchart in Fig 4.6 helps to support the argument that the models based on an attribution prior penalty function that resembles an L1 norm generally perform worse than the IG L2 and EG L2 models. While the IG L2 and EG L2 performance results are very comparable to the baseline CNN, it is clear that there is a considerable gap in the precision, recall and F1 macro scores of the L1 models. This suggests that the IG L1 and EG L1 struggle when encountering completely unseen patients' data in the testing set.

Although the results of the EG L2 model is comparable, and slightly higher in all metrics, to that of the baseline CNN, and could suggest that this novel model's performance is significant, there is not sufficient evidence to prove this. This is due to the fact that this experiment could not be performed the same way that the beat holdout experiment was by using repeated stratified 10-fold cross validation so that statistical analysis could be performed. The patient holdout experiment does provide enough evidence to make the claim that the EG L2 model is the best of the proposed novel models. This is because the results of the repeated patient holdout experiment showed it to perform better overall than the IG L2 model. It is possible that a repeatable experiment based on the patient holdout method could be devised, whereby the models at each fold have a new set of unseen patients to be tested upon. This is not likely to be effective when using the MIT-BIH dataset, considering that there are only 47 patients, however, could be possible on a much larger dataset in future work, such as the PTB Diagnostic ECG Database (ptb).

5.4 Explainability Methods

In Figure A.3, the sets of correct and incorrect integrated gradient heatmaps provides useful information about coming to conclusions about why the IG L2 model misclassified a certain heartbeat class. For example, the incorrect heatmap for class 8, which is a paced beat, indicates that the model didn't attribute the output prediction to the region from the R peak down to the peak of the S wave (see Figure 2.1), which the model appears to usually do when correctly classifying this heartbeat class. This is not the case for correct and incorrect heatmaps produced using expected gradients and GradCAM (see Figures A.5 and A.4), as the correct and incorrect heatmaps are too similar. This issue, alongside the large amount of noise in both these sets of heatmaps, means that is difficult to draw conclusions about why a model made a mis-classification, which was made possible with integrated gradients.

When looking at the more general segmented average correct and incorrect attributions shown in Figures 4.9, 4.10, 4.11 and 4.12 a few more observations can be made about the quality of these explainability methods. A clear difference between the average correct and incorrect attributions can facilitate in concluding why the a model failed in classification due to its incorrect average focus on a certain section of the beat morphology. This clear difference is visible for the integrated gradients explainability method in Figures 4.9 and 4.11. For the most part, the expected gradients and GradCAM methods did not show this clear distinction between correct and incorrect attributions and this further demonstrates that the integrated gradients appears to be the best explainability method out of the three.

5.5 Conclusion

The evaluation results clearly show that the baseline CNN was the best performing classifier model in both the beat holdout experiment with the highest achieved accuracy, precision and F1 macro scores. The EG L2 displayed the highest performance in the patient holdout experiment with an accuracy of 0.989, precision of 0.684, recall of 0.711 and F1 macro of 0.6967. The EG L2 model performing slightly better than the baseline CNN across all metrics in this patient holdout experiment. However, because this experiment was not repeatable no statistical significance can be concluded from this result. Generally, the results have shown that the L2 novel models perform better than the L1 models. The integrated gradients was the best explainability method evaluated in this study. The attributions it gave to sections of each heartbeat morphology were much clearer and less noisy than those produced by the expected gradients and GradCAM methods. Overall, I can conclude that incorporating explanations from expected gradients as attribution priors into model training is viable and appropriate in the problem domain of time-series ECG heartbeat classification.

5.6 Limitations and Future Work

Due to the amount of resources and time required to execute the quantitative experiment to compare model performances in the beat and patient holdout experiments, much less emphasis was placed on the qualitative comparisons of the explainability aspects provided from integrated and expected gradients. The heatmaps produced in Figures A.3, A.5 and A.4 were not the most informative when it came to assessing the abilities of each explainability method. In addition, looking at more general results provided in the segmented average correct and incorrect attributions did little in uncovering further reasons as to why expected gradients and GradCAM were not as useful in explaining model decisions for classification. In future work, more time could be spent on the comparison between these three explainability methods and furthermore comparisons of each method on a more micro-scopic level, for example examining heatmaps

from the attributions produced on specific beats. Future work into investigating the explainability could involve comparison against other methods, such as LIME (Ribeiro et al. 2016).

One of the discussed reasons for why the models based on integrated gradients did not perform as well as their equivalents based on expected gradients was that they could only be trained over two epochs due to long training times, as opposed to ten epochs used by all the other deep learning models. The integrated gradient based models were as only trained whilst using five interpolation steps due to the same reasons for long training times, as opposed to a recommended number of interpolation steps being between 20 and 300. It is clear that expected gradients are much more suitable as an attribution prior due to the significantly shorter training times. However, further work could show that despite a possible increase in performance achieved by training the IG L1 and IG L2 models for 10 epochs and over an optimized number of interpolation steps, this does not provide a noteworthy enough advantage over expected gradients.

In Section 3.7 it was discussed that the lambda values used for regularising the attribution prior function in model training as being found manually. There are obvious drawbacks to manually iteratively tuning a hyperparameter, including that the optimal lambda was definitely not found. This had to be done manually because the models were trained in an overridden *@train_step* function and so traditional grid search was not possible for this hyperparameter. If instead the overridden *@train_step* was implemented in a completely overridden Keras model class, then using the automated grid search provided by scikit-learn may of been possible.

Finally, the patient holdout experiment that was adapted from the work done in (Jones et al. 2020) was not repeatable, since there was only one set of patients heldout in the testing dataset. This meant that testing the statistical significance of the result of this experiment were not possible due to it not being repeatable. As discussed in Section 5.3, a patient holdout experiment that was performed with repeated stratified 10-fold cross validation would be possible with a much larger dataset and more patients such as the PTB database, as opposed to the limited 47 patients provided in the MIT-BIH dataset. This would mean that at each fold the models would be tested on a new set of unseen patients and statistical analysis could be performed on the results. Following this method would be very useful in evaluating how well the novel models proposed in this work generalise to unseen patients' data.

A Appendices

A.1 Performance Metrics

Where T_p is the number of true positives, T_n is the number of true negatives, F_p is the number of false positives and F_n is the number of false negatives. The below definitions are all from the scikit-learn documentation pages (Varoquaux et al. 2015).

A.1.1 Accuracy

From the scikit-learn documentation, it is *"...the set of labels predicted for a sample must exactly match the corresponding set of labels..."*.

$$Accuracy = \frac{T_n + T_p}{T_n + T_p + F_n + F_p} \tag{A.1}$$

A.1.2 Precision

From the scikit-learn documentation, it is *"...the ability of the classifier not to label as positive a sample that is negative."*.

$$Precision = \frac{T_p}{T_p + F_p} \tag{A.2}$$

A.1.3 Recall

From the scikit-learn documentation, it is *"...the ability of the classifier to find all the positive samples."*.

$$Recall = \frac{T_p}{T_n + T_p} \tag{A.3}$$

A.1.4 F1 Macro

From the scikit-learn documentation, it is *"...the harmonic mean of the precision and recall..."*.

$$F1Macro = \frac{Precision \times Recall}{Precision + Recall} \tag{A.4}$$

	SVM (BEAT HOLDOUT)							
	N	L	R	V	A	F	f	/
Accuracy	0.977	0.879	0.827	0.907	0.715	0.684	0.824	0.92
Precision	0.23	0.351	0.274	0.254	0.237	0.294	0.247	0.368
Recall	0.225	0.318	0.235	0.231	0.175	0.203	0.211	0.351
F1 Macro	0.228	0.326	0.243	0.24	0.193	0.23	0.222	0.356
F1 Micro	0.977	0.879	0.827	0.907	0.715	0.684	0.824	0.92
	BASELINE CNN (BEAT HOLDOUT)							
	N	L	R	V	A	F	f	/
Accuracy	0.979	0.84	0.767	0.882	0.647	0.599	0.776	0.893
Precision	0.245	0.355	0.277	0.253	0.242	0.299	0.25	0.356
Recall	0.24	0.312	0.225	0.225	0.165	0.183	0.203	0.335
F1 Macro	0.243	0.322	0.235	0.236	0.185	0.215	0.217	0.341
F1 Micro	0.979	0.84	0.767	0.882	0.647	0.599	0.776	0.893
	IG L2 CNN (BEAT HOLDOUT)							
	N	L	R	V	A	F	f	/
Accuracy	0.976	0.871	0.812	0.903	0.689	0.635	0.808	0.917
Precision	0.238	0.357	0.273	0.25	0.238	0.303	0.249	0.364
Recall	0.233	0.322	0.235	0.227	0.172	0.195	0.209	0.347
F1 Macro	0.235	0.331	0.243	0.237	0.192	0.228	0.221	0.352
F1 Micro	0.976	0.871	0.812	0.903	0.689	0.635	0.808	0.917
	IG L1 CNN (BEAT HOLDOUT)							
	N	L	R	V	A	F	f	/
Accuracy	0.979	0.965	0.951	0.945	0.814	0.733	0.893	0.984
Precision	0.217	0.378	0.287	0.256	0.253	0.308	0.251	0.394
Recall	0.214	0.368	0.276	0.245	0.209	0.228	0.228	0.391
F1 Macro	0.215	0.372	0.281	0.25	0.229	0.261	0.238	0.392
F1 Micro	0.979	0.965	0.951	0.945	0.814	0.733	0.893	0.984
	EG L2 CNN (BEAT HOLDOUT)							
	N	L	R	V	A	F	f	/
Accuracy	0.996	0.992	0.991	0.939	0.862	0.763	0.924	0.995
Precision	0.216	0.386	0.326	0.232	0.286	0.306	0.268	0.448
Recall	0.215	0.383	0.323	0.218	0.247	0.234	0.248	0.446
F1 Macro	0.215	0.385	0.325	0.225	0.265	0.264	0.258	0.447
F1 Micro	0.996	0.992	0.991	0.939	0.862	0.763	0.924	0.995
	EG L1 CNN (BEAT HOLDOUT)							
	N	L	R	V	A	F	f	/
Accuracy	0.995	0.992	0.989	0.939	0.863	0.77	0.925	0.994
Precision	0.22	0.412	0.29	0.243	0.291	0.3	0.257	0.407
Recall	0.219	0.409	0.287	0.229	0.251	0.232	0.238	0.405
F1 Macro	0.219	0.41	0.289	0.236	0.269	0.261	0.247	0.406
F1 Micro	0.995	0.992	0.989	0.939	0.863	0.77	0.925	0.994

Figure A.1: *Full breakdown of all model results per class for beat holdout experiment.*

	SVM (PATIENT HOLDOUT)							
	N	L	R	V	A	F	f	/
Accuracy	0.976	nan	nan	0.989	nan	0.994	0.989	0.994
Precision	0.152	nan	nan	0.333	nan	0.722	0.472	0.283
Recall	0.148	nan	nan	0.33	nan	0.72	0.469	0.281
F1 Macro	0.15	nan	nan	0.332	nan	0.721	0.47	0.282
F1 Micro	0.976	0.0	0.0	0.989	0.0	0.994	0.989	0.994
	BASELINE CNN (PATIENT HOLDOUT)							
	N	L	R	V	A	F	f	/
Accuracy	0.982	nan	nan	0.993	nan	0.994	0.988	0.995
Precision	0.149	nan	nan	0.35	nan	0.767	0.517	0.273
Recall	0.146	nan	nan	0.348	nan	0.764	0.513	0.272
F1 Macro	0.147	nan	nan	0.349	nan	0.765	0.515	0.273
F1 Micro	0.982	0.0	0.0	0.993	0.0	0.994	0.988	0.995
	IG L2 CNN (PATIENT HOLDOUT)							
	N	L	R	V	A	F	f	/
Accuracy	0.982	nan	nan	0.992	nan	0.993	0.985	0.995
Precision	0.144	nan	nan	0.354	nan	0.708	0.396	0.258
Recall	0.142	nan	nan	0.352	nan	0.705	0.391	0.257
F1 Macro	0.143	nan	nan	0.353	nan	0.707	0.393	0.258
F1 Micro	0.982	0.0	0.0	0.992	0.0	0.993	0.985	0.995
	IG L1 CNN (PATIENT HOLDOUT)							
	N	L	R	V	A	F	f	/
Accuracy	0.981	nan	nan	0.994	nan	0.993	0.99	0.996
Precision	0.137	nan	nan	0.389	nan	0.833	0.444	0.278
Recall	0.134	nan	nan	0.387	nan	0.83	0.441	0.277
F1 Macro	0.136	nan	nan	0.388	nan	0.832	0.443	0.277
F1 Micro	0.981	0.0	0.0	0.994	0.0	0.993	0.99	0.996
	EG L2 CNN (PATIENT HOLDOUT)							
	N	L	R	V	A	F	f	/
Accuracy	0.986	nan	nan	0.995	nan	1.0	0.995	0.996
Precision	0.134	nan	nan	0.417	nan	1.0	0.5	0.292
Recall	0.132	nan	nan	0.415	nan	1.0	0.498	0.29
F1 Macro	0.133	nan	nan	0.416	nan	1.0	0.499	0.291
F1 Micro	0.986	0.0	0.0	0.995	0.0	1.0	0.995	0.996
	EG L1 CNN (PATIENT HOLDOUT)							
	N	L	R	V	A	F	f	/
Accuracy	0.986	nan	nan	0.995	nan	1.0	0.991	0.997
Precision	0.125	nan	nan	0.333	nan	1.0	0.5	0.25
Recall	0.123	nan	nan	0.332	nan	1.0	0.496	0.249
F1 Macro	0.124	nan	nan	0.333	nan	1.0	0.498	0.25
F1 Micro	0.986	0.0	0.0	0.995	0.0	1.0	0.991	0.997

Figure A.2: *Full breakdown of all model results per class for patient holdout experiment.*

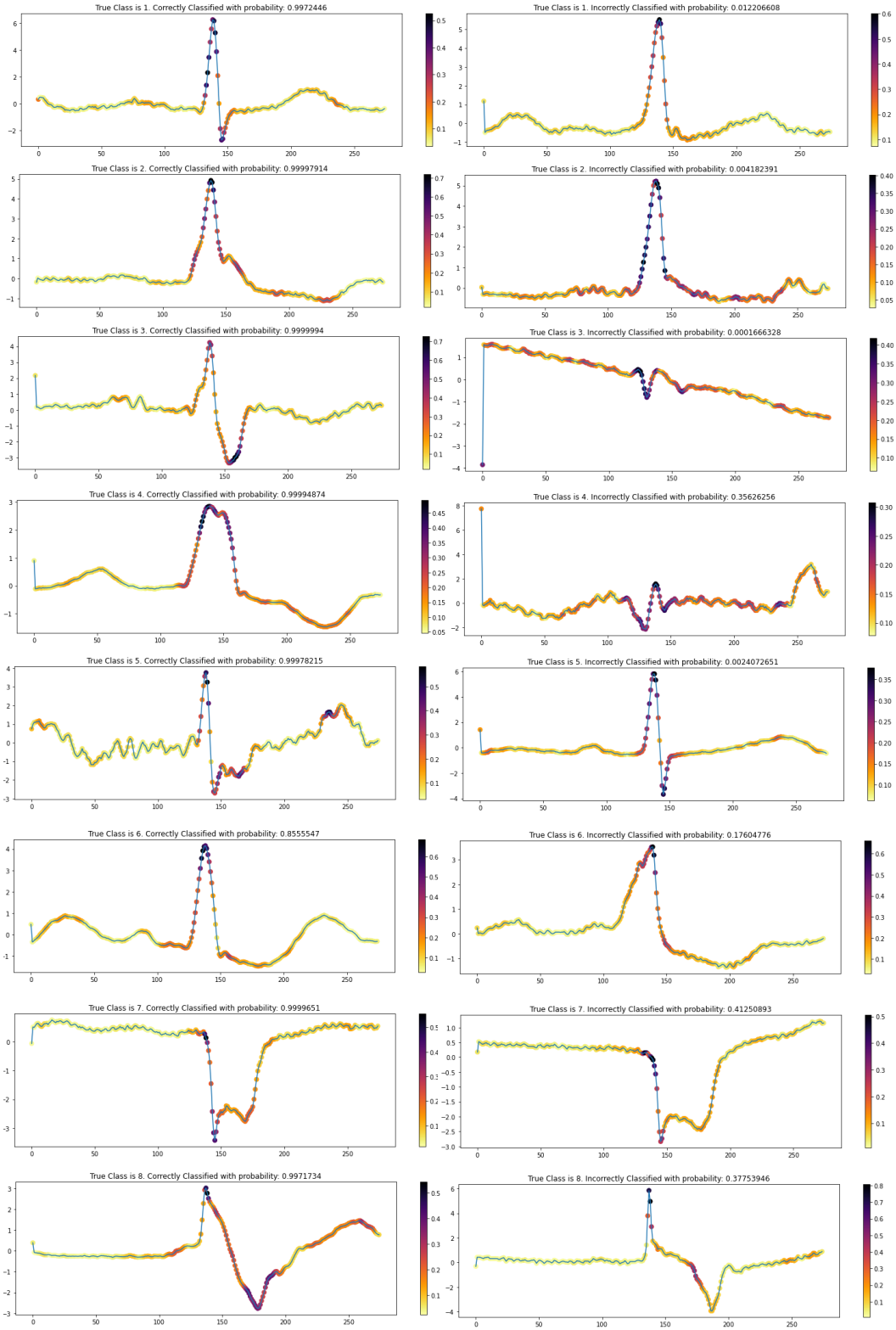


Figure A.3: *Heatmaps produced from integrated gradient attributions on correctly and incorrectly classified heartbeat classes. This was created using the IG L2 model trained on the beat holdout dataset.*

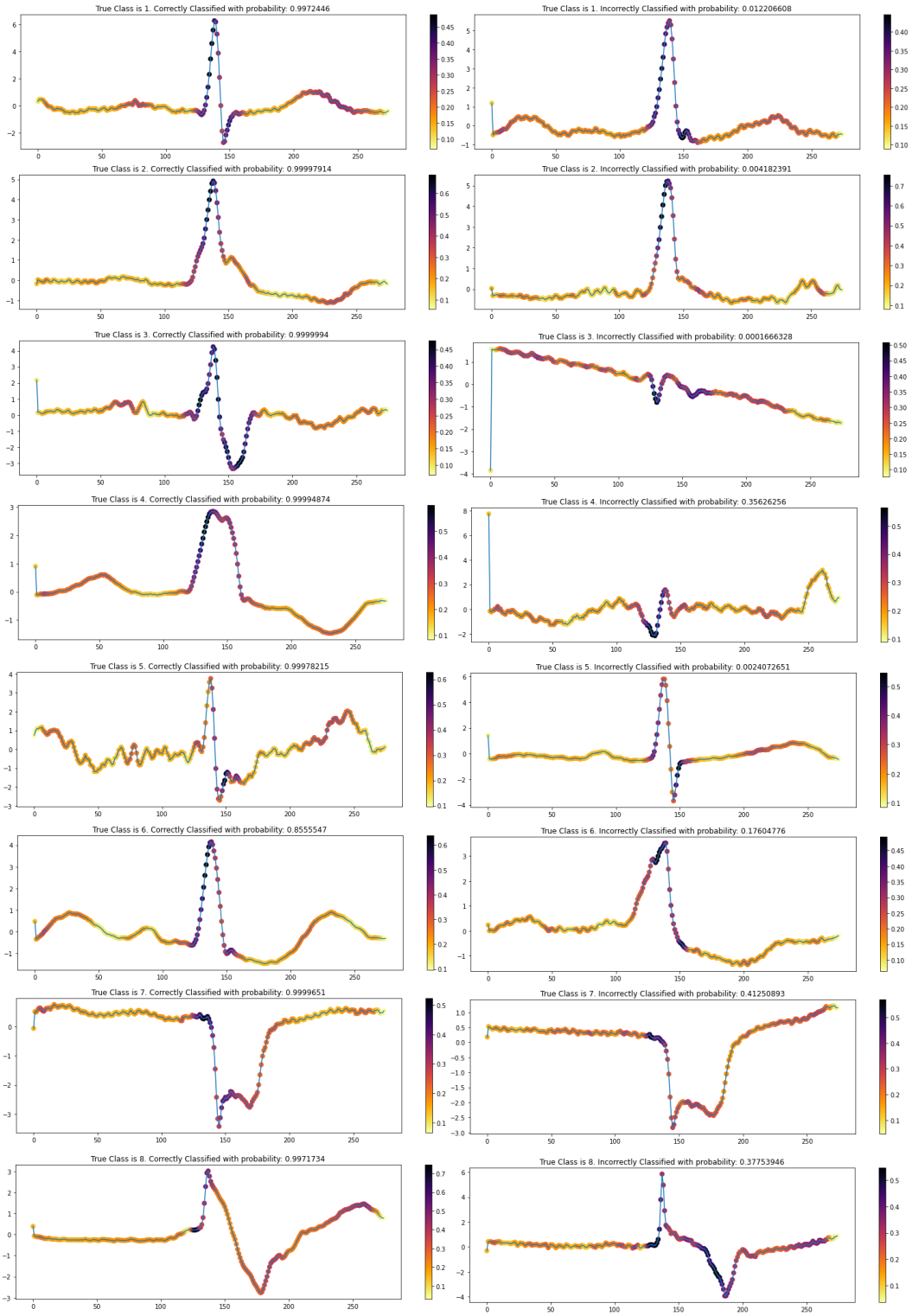


Figure A.4: *Heatmaps produced from GradCAM on correctly and incorrectly classified heartbeat classes. This was created using the IG L2 model trained on the beat holdout dataset.*

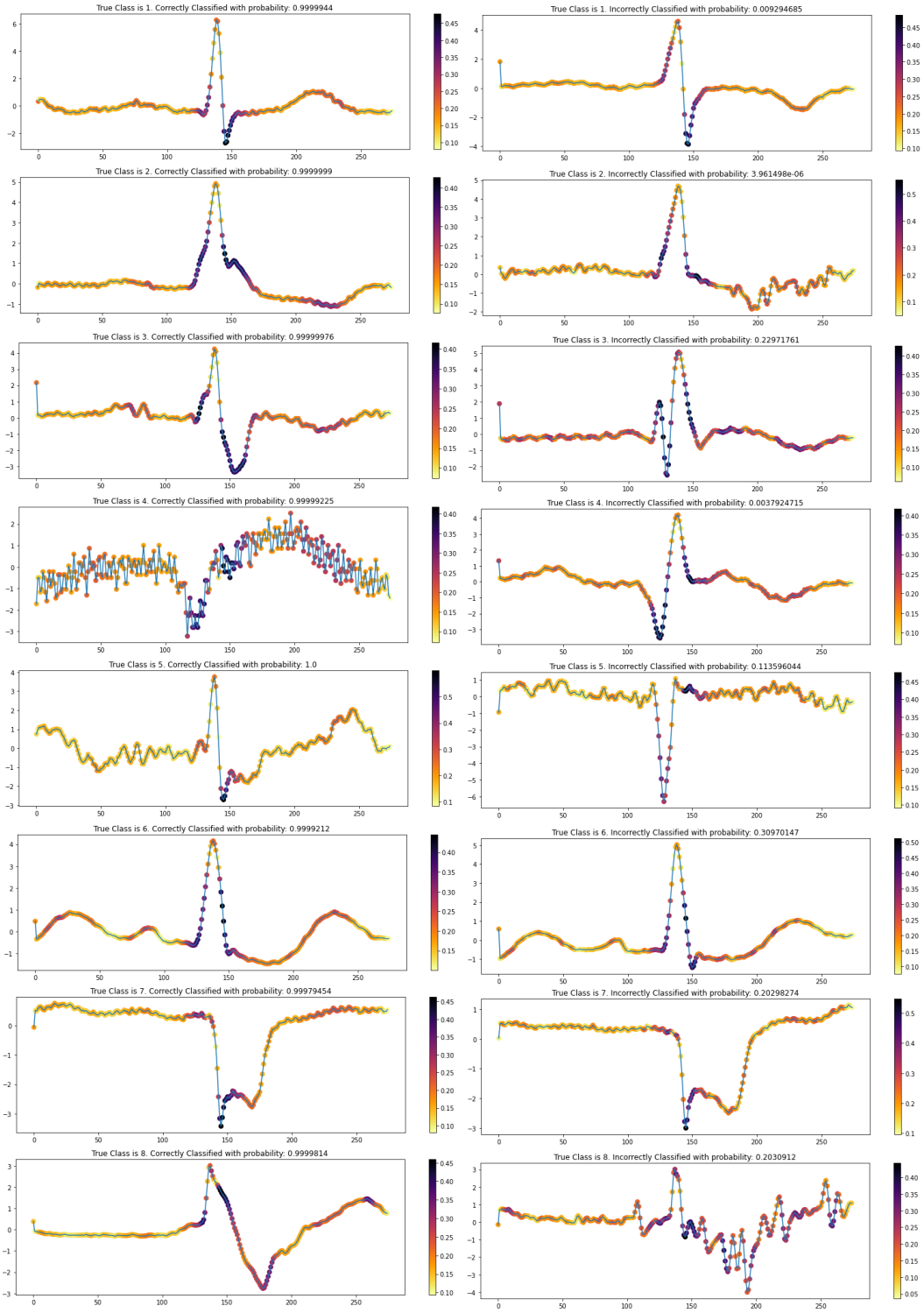


Figure A.5: *Heatmaps produced from integrated gradient attributions on correctly and incorrectly classified heartbeat classes. This was created using the EG L2 model trained on the beat holdout dataset.*

5 Bibliography

Welcome to biosppy — biosppy 0.6.1 documentation. URL https://biosppy.readthedocs.io/en/stable/.

Ptb diagnostic ecg database v1.0.0. URL https://www.physionet.org/content/ptbdb/1.0.0/.

wfdb-python, 04 2022. URL https://github.com/MIT-LCP/wfdb-python.

U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, A. Gertych, and R. S. Tan. A deep convolutional neural network model to classify heartbeats. *Computers in Biology and Medicine*, 89:389–396, 10 2017. doi: 10.1016/j.compbiomed.2017.08.022.

A. Anaya-Isaza, L. Mera-Jiménez, and M. Zequera-Diaz. An overview of deep learning in medical imaging. *Informatics in Medicine Unlocked*, 26:100723, 2021. doi: 10.1016/j.imu.2021.100723.

L. E. Atlas, T. Homma, and R. J. Marks. An artificial neural network for spatio-temporal bipolar patterns: Application to phoneme classification. In *NIPS*, 1987.

D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Mazller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11:1803–1831, 06 2010.

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*, 58:82–115, 06 2020. doi: 10.1016/j.inffus.2019.12.012.

G. I. Boser, B.E. and V. Vapnik. A training algorithm for optimal margin classifiers. *In Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.

J. Brownlee. How to use one-vs-rest and one-vs-one for multi-class classification, 04 2020. URL https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 06 2002. doi: 10.1613/jair.953.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, dec 2006. ISSN 1532-4435.

N. Diamantidis, D. Karlis, and E. Giakoumakis. Unsupervised stratification of cross-validation for accuracy estimation. *Artificial Intelligence*, 116:1–16, 01 2000. doi: 10.1016/s0004-3702(99)00094-6.

L. Edenbeandt, B. Devine, and P. W. Macfarlane. Classification of electrocardiographic st-t segments – human expert vs artificial neural network. *European Heart Journal*, 14:464–468, 04 1993. doi: 10.1093/eurheartj/14.4.464.

G. Erion. Attribution priors, 03 2022. URL https://github.com/suinleelab/attributionpriors/blob/master/example_usage_tf2.ipynb.

G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, and S.-I. Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*, 3:620–631, 05 2021. doi: 10.1038/s42256-021-00343-w.

S. Farrugia, H. Yee, and P. Nickolls. Implantable cardioverter defibrillator electrogram recognition with a multilayer perceptron. *Pacing and Clinical Electrophysiology*, 16:228–234, 01 1993. doi: 10.1111/j.1540-8159.1993.tb01567.x.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine*, 25:65–69, 01 2019. doi: 10.1038/s41591-018-0268-3.

B. Hedén, L. Edenbrandt, W. K. Haisty, and O. Pahlm. Artificial neural networks for the electrocardiographic diagnosis of healed myocardial infarction. *The American Journal of Cardiology*, 74:5–8, 07 1994. doi: 10.1016/0002-9149(94)90482-0.

N. Jannah, S. Hadjiloucas, and J. Al-Malki. Arrhythmia detection using multi-lead ecg spectra and complex support vector machine classifiers. *Procedia Computer Science*, 194:69–79, 2021. doi: 10.1016/j.procs.2021.10.060.

I. Jolliffe. *Principal Component Analysis*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

Y. Jones, F. Deligianni, and J. Dalton. Improving ecg classification interpretability using saliency maps. *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*, 10 2020. doi: 10.1109/bibe50027.2020.00114.

Y. Kikawa and K. Oguri. A study for excluding incorrect detections of holter ecg data using svm. *Neural Information Processing*, pages 1223–1228, 2004. doi: 10.1007/978-3-540-30499-9_190.

P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim. The (un)reliability of saliency methods. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280, 2019. doi: 10.1007/978-3-030-28954-6_14.

S. Kiranyaz, T. Ince, and M. Gabbouj. Real-time patient-specific ecg classification by 1-d convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, 63:664–675, 03 2016. doi: 10.1109/tbme.2015.2468589.

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, page 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603638.

F. Liu and B. Avci. Incorporating priors with feature attribution on text classification. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. doi: 10.18653/v1/p19-1631.

G. Moody and R. Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20:45–50, 2001. doi: 10.1109/51.932724.

NHS. Arrhythmia, 2019. URL https://www.nhs.uk/conditions/arrhythmia/.

S. Osowski, L. Hoai, and T. Markiewicz. Support vector machine-based expert system for reliable heartbeat recognition. *IEEE Transactions on Biomedical Engineering*, 51:582–589, 04 2004. doi: 10.1109/tbme.2004.824138.

K. O'Shea and R. Nash. An introduction to convolutional neural networks. 2015.

D. G. Pereira, A. Afonso, and F. M. Medeiros. Overview of friedman's test and post-hoc analysis. *Communications in Statistics - Simulation and Computation*, 44:2636–2653, 08 2014. doi: 10.1080/03610918.2014.931971.

A. Rabee and I. Barhumi. Ecg signal classification using support vector machine based on wavelet multiresolution analysis, 07 2012. URL https://ieeexplore.ieee.org/document/6310497.

P. M. Rautaharju, M. Ariet, T. A. Pryor, R. C. Arzbaecher, J. J. Bailey, R. Bonner, C. R. Goetowski, J. K. Hooper, V. Klein, C. K. Millar, J. A. Milliken, D. W. Mortara, H. V. Pipberger, L. Pordy, R. L. Sandberg, R. L. Simmons, and H. K. Wolf. The quest for optimal electrocardiography. task force iii: Computers in diagnostic electrocardiography. *The American Journal of Cardiology*, 41:158–170, 01 1978. doi: 10.1016/0002-9149(78)90150-9.

M. Reddy, L. Edenbrandt, J. Svensson, W. Haisty, and O. Pahlm. Neural network versus electrocardiographer and conventional computer criteria in diagnosing anterior infarct from the ecg. *Proceedings Computers in Cardiology*, 1992. doi: 10.1109/cic.1992.269345.

P. Refaeilzadeh, L. Tang, and H. Liu. Cross-validation. *Encyclopedia of Database Systems*, pages 532–538, 2009. doi: 10.1007/978-0-387-39940-9_565. URL https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_565.

M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?". *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 2016. doi: 10.1145/2939672.2939778.

L. Roeder. lutzroeder/netron, 05 2020. URL https://github.com/lutzroeder/netron.

A. S. Ross, M. C. Hughes, and F. Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 08 2017. doi: 10.24963/ijcai.2017/371.

G. A. Roth, G. A. Mensah, C. O. Johnson, G. Addolorato, E. Ammirati, L. M. Baddour, N. C. Barengo, A. Z. Beaton, E. J. Benjamin, C. P. Benziger, A. Bonny, M. Brauer, M. Brodmann, T. J. Cahill, J. Carapetis, A. L. Catapano, S. S. Chugh, L. T. Cooper, J. Coresh, M. Criqui, N. DeCleene, K. A. Eagle, S. Emmons-Bell, V. L. Feigin, J. Fernández-Solà, G. Fowkes, E. Gakidou, S. M. Grundy, F. J. He, G. Howard, F. Hu, L. Inker, G. Karthikeyan, N. Kassebaum, W. Koroshetz, C. Lavie, D. Lloyd-Jones, H. S. Lu, A. Mirijello, A. M. Temesgen, A. Mokdad, A. E. Moran, P. Muntner, J. Narula, B. Neal, M. Ntsekhe, G. Moraes de Oliveira, C. Otto, M. Owolabi, M. Pratt, S. Rajagopalan, M. Reitsma, A. L. P. Ribeiro, N. Rigotti, A. Rodgers, C. Sable, S. Shakil, K. Sliwa-Hahnle, B. Stark, J. Sundström, P. Timpel, I. M. Tleyjeh, M. Valgimigli, T. Vos, P. K. Whelton, M. Yacoub, L. Zuhlke, C. Murray, and V. Fuster. Global burden of cardiovascular diseases and risk factors, 1990-2019: Update from the gbd 2019 study. *Journal of the American College of Cardiology*, 76:2982–3021, 12 2020. doi: 10.1016/j.jacc.2020.11.010.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 05 2019. doi: 10.1038/s42256-019-0048-x.

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, 11 1992. doi: 10.1016/0167-2789(92)90242-f.

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. *2017 IEEE International Conference on Computer Vision (ICCV)*, 10 2017. doi: 10.1109/iccv.2017.74.

M. A. Serhani, H. T. El Kassabi, H. Ismail, and A. Nujum Navaz. Ecg monitoring systems: Review, architecture, processes, and key challenges. *Sensors*, 20:1796, 03 2020. doi: 10.3390/s20061796.

D. Shen, G. Wu, and H.-I. Suk. Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19:221–248, 06 2017. doi: 10.1146/annurev-bioeng-071516-044442.

A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, 2014.

P. Sturmfels, S. Lundberg, and S.-I. Lee. Visualizing the impact of feature attribution baselines. *Distill*, 5, 01 2020. doi: 10.23915/distill.00022.

M. Sundararajan, A. Taly, and Q. Yan. Gradients of counterfactuals. 2016.

M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org, 2017.

G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller. Scikit-learn. *GetMobile: Mobile Computing and Communications*, 19:29–33, 06 2015. doi: 10.1145/2786984.2786995.

T. J. Viering, Z. Wang, M. Loog, and E. Eisemann. How to manipulate cnns to make them lie: the gradcam case. 2019.

Q. Wei and R. L. Dunbrack. The role of balanced training and testing data sets for binary classifiers in bioinformatics. *PLoS ONE*, 8:e67863, 07 2013. doi: 10.1371/journal.pone.0067863.

WHO. Cardiovascular diseases, 2019. URL `https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1`.

S. S. Yadav and S. M. Jadhav. Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big Data*, 6, 12 2019. doi: 10.1186/s40537-019-0276-2.

Q. Zhao and L. Zhang. Ecg feature extraction and classification using wavelet transform and support vector machines. *2005 International Conference on Neural Networks and Brain*, 2005. doi: 10.1109/icnnb.2005.1614807.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2016. doi: 10.1109/cvpr.2016.319.