

PREDICTING HEART DISEASE: ID CNN VS. LINEAR AND NONLINEAR SVM TECHNIQUES FOR ECG ANALYSIS

by

PATRICK COURTS

A thesis submitted in partial fulfilment
of the requirements for the degree of

MSC PREDICTIVE MODELLING AND
SCIENTIFIC COMPUTING

UNIVERSITY OF WARWICK
DEPARTMENT OF ENGINEERING

May 2024

CONTENTS

Contents	ii
List of figures	iv
List of tables	v
Declaration	vi
Abstract	vii
List of abbreviations	viii
1 Introduction	1
2 Literature Review	3
2.1 Background	3
2.2 Preprocessing	4
2.3 Feature Extraction	6
2.4 Feature Selection	6
2.5 Machine Learning	8
2.6 Model Validation	9
3 Methods	11
3.1 Database Filtering	11
3.2 Data Preprocessing	12
3.2.1 Butterworth	13
3.2.2 Discrete Wavelet Transform	13
3.3 Feature Extraction	15
3.3.1 Time and Amplitude Features	15
3.3.1.1 Moments	15
3.3.1.2 R parameters	16
3.3.2 Frequency Domain Features	16
3.3.2.1 discrete wavelet transform (DWT)	16
3.3.2.2 Power Spectrum	17
3.3.3 Nonlinear Domain Features	17

3.3.3.1	Signal Embedding	17
3.3.3.2	Shannon Entropy	17
3.3.3.3	Sample Entropy	18
3.3.3.4	Poincaré Plot	18
3.3.3.5	Fractal Dimensions	18
3.3.4	Covariates	21
3.4	Feature Selection	21
3.5	Machine Learning	22
3.5.1	Support Vector Machine (SVM)	22
3.5.2	Convolutional Neural Network (CNN)	23
4	Results	28
4.1	Binary SVM	29
4.1.1	Calculated Features	29
4.1.2	Selected Features	29
4.1.3	Channel and Optimised Performance	30
4.2	Binary 1D CNN	32
4.2.1	Model Performance	32
4.3	Filter Type Evaluation	34
5	Discussion	35
6	Conclusions and Further Work	38
Bibliography		40
A	Appendix A	46
B	Appendix B	47
C	Appendix C	48

LIST OF FIGURES

2.1	Location of 12-lead electrocardiogram (ECG) electrodes	3
2.2	Ideal electrocardiogram (ECG) signal	4
2.3	Prediction vs Truth table	9
3.1	Outline of method structure	11
3.2	Different standards of raw data	13
3.3	DWT schematic	14
3.4	Compares Butterworth and DWT filters	15
3.5	Power spectral density of ECG	16
3.6	Example Poincaré Plot	19
3.7	Example embedded attractor	20
3.8	CNN max pooling	24
3.9	Outline of structure of CNN	25
3.10	Activation functions	26
4.1	Diagnosis distribution of filtered database	28
4.2	Violin plot for RR mean calculations	29
4.3	Selected features distribution per channel	30
4.4	Selected channels in optimised accuracy	31
4.5	Confusion Matrix and ROC Curve for SVM	32
4.6	Accuracy and Loss whilst training CNN	32
4.7	CNN ROC curve and confusion matrix	33
A.1	Example metadata file	46
B.1	Example CNN code	47
C.1	Example Violin plots	49

LIST OF TABLES

2.1	Commonly used heart rate variability parameters	7
3.1	Dataset diagnostic distribution	12
3.2	Gridsearch hyperparameters for SVM	23
3.3	Hyperparameters for creation and training of convolutional neural network (CNN)	26
4.1	Scores with standard deviations for each channel, averaged over 10 iterations.	31
4.2	Performance Metrics for convolutional neural network (CNN)	33
4.3	Comparison of filter methods for both models	34
5.1	Comparison of different studies with respect to model, database, number of classifications, and accuracy.	35

DECLARATION

This thesis is submitted to the University of Warwick in support of my application for the degree of MSc Predictive Modelling and Scientific Computing. The work presented (including data generated and data analysis) was carried out by the author. The length of the main body of this report does not exceed 12,000 words.

ABSTRACT

Heart disease is one of the leading causes of mortality in the UK. The electrocardiogram (ECG) is a common tool used to graphically represent the electrical activity of the heart and provides the means to obtain diagnoses of heart conditions. It is standard practise for this diagnosis to be done visually by a medical professional however, this is a time consuming task, prone to human error. As such, automatic diagnosis methods, making use of machine learning methods, may provide a more desirable alternative. This paper investigates two such methods: the support vector machine (SVM) and the 1D convolutional neural network (CNN) for the binary classification of ECGs into healthy and unhealthy classes. The ECG signals are denoised using the discrete wavelet transform (DWT) method. The denoised signals are discretised into time, frequency and non-linear domain parameters before the optimal features are used within the SVM model. The efficiency of each subset of parameters is investigated before it is concluded that parameters from every domain are required for an accurate classification model. The SVM model resulted in an optimised classification accuracy of $86 \pm 1\%$. In contrast the 1D CNN takes the complete denoised signals as its input and resulted in an average accuracy of $96 \pm 1\%$. This implies that the 1D CNN has a greater potential for accurate real-time ECG signal classification than the SVM method.

LIST OF ABBREVIATIONS

AUC	area-under-the-curve
CNN	convolutional neural network
CVD	cardiovascular disease
DB4	Daubechies4
DWT	discrete wavelet transform
ECG	electrocardiogram
FFT	fast Fourier transform
FN	false negative
FP	false positive
HRV	heart rate variability
KNN	K-nearest neighbour
PSD	power spectral density
PTB	Physikalisch-Technische Bundesanstalt
RBF	radial basis functions
ReLU	rectified linear unit
RMSSD	root mean sum of the squared differences
ROC	receiver operating characteristic
SNR	signal-to-noise ratio
SVD	singular value decomposition
SVM	support vector machine
TN	true negative
TP	true positive

INTRODUCTION

The British Heart Foundation states that, as of April 2023, 25% of all deaths in the UK are attributed to various forms of heart and cardiovascular disease (CVD) [1]. The large range of CVDs, with each having their own method of treatment, means that the early detection of a potential cardiac defect is vital to ensure the appropriate response is given in order to reduce this mortality rate [2]. Through attempts to achieve this goal, the human heart and its underlying dynamics have been extensively examined for potential indicators of cardiac disease [3], [4] and [5]. These studies have shown that the heart exhibits a range of behaviours, stemming from underlying non-linear dynamics in the form of complex oscillations, chaotic patterns, and bifurcations [6]. Whilst the complexity of a healthy heart has been shown to differ to that of an unhealthy heart, the multitude of parameters leading to this complexity means that the development of a cardiovascular model is a non-trivial task [7], [8]. Subsequently, investigation of the cardiovascular system is often performed through a signal analysis approach.

The electrocardiogram (ECG) has become the most commonly used biosignal in the prompt detection of CVDs [9]. This non-invasive approach requires the placement of several sensors across the body to provide a graphical representation of the electrical activity of the heart. The resulting ECG can be visually inspected by medical professionals to interpret the patient's diagnosis. Although this is the current standard, visual inspection is a time-consuming process, requiring considerable expertise whilst still being prone to bias and human error [10]. Therefore, many studies are investigating different methods that do not require manual interpretation, instead using automatic machine learning (ML) classification [10], [11], [12], [13], [14] and [15].

These methods can be broadly categorised into two groups: those that require manual feature extraction from ECG signals, such as K-nearest neighbour (KNN), decision trees and support vector machine (SVM), and those where features and patterns are automatically learned from the data, like neural networks. This report will focus on an approach from each category. The first approach involves parameterising the ECG signal into respective time, frequency and non-linear domain parameters, followed by binary classification using the SVM. The SVM has recently gained popularity in ECG classification due to its potential for rapid diagnosis, provided that suitably discriminatory parameters can be extracted from the signals. This approach will be compared with the use of the complete signals within a 1D convolutional neural network (CNN). The 1D CNN is a promising method for ECG classification due to attributes such as its temporal and spatial invariance. These enable

efficient automatic feature extraction through learning patterns that may appear throughout the length of the ECG signals. Whilst traditional CNNs are typically applied to 2D data, such as images [16], the 1D variant is an emerging technique designed specifically for 1D signals, showing success in biomedical signal processing [17]. The performance of both methods will be evaluated using standard ML accuracy metrics as well as confusion matrices and receiver operating characteristic (ROC) curves. These scores will also be compared to those from similar studies to evaluate the proposed models' performance.

While the clinical applications of this investigation are evident, the growing prevalence of wearable technology, such as smart watches, is expanding access to continuous heart rate data at the individual level. If these technologies can be equipped with methods to accurately predict various cardiac defects, they could potentially lead to a significant reduction in cardiac fatalities through early detection allowing timely intervention [5]. Subsequently, the classification method implemented should not only provide acute accuracy but should also have the potential to give health diagnoses in an appropriate time frame to be used in a mass setting. This search for a model to provide accurate and real-time ECG classification is the key issue in this field and will be the subsequent areas of evaluation for the methods proposed in this project.

The rest of this report is organised in the following manner. A literature review outlining and comparing relevant techniques for the classification of ECG signals can be seen in chapter 2. Chapter 3 presents descriptions of the methods and techniques used within this report. The outcomes of these methodologies are then displayed in chapter 4 before they are evaluated and explained within chapter 5. Finally, chapter 6, concludes the work and offers potential areas for further investigation.

LITERATURE REVIEW

2

The field of research considered in this project is rapidly evolving, therefore, this literature review aims to analyze the techniques within more recently published materials while highlighting fundamental methodologies common to multiple periods and their origins. The review is performed with the goal to provide a foundational base for the project which will look to use and combine the robust aspects of those existing materials. These will be used as benchmarks for analysis such that comparisons between initial results to those reported in the relevant literature would provide adequate validation. Combination, evaluation and modification of the techniques seen within these materials will ensure this project builds upon previous research and presents a new piece of scientific literature.

2.1 BACKGROUND

The cardiac cycle is governed by the heart's electrical conduction system. The electrical signals dispersed throughout the heart cause different muscles within the heart to contract, leading to blood being pumped around the circulatory system [18]. The 12-lead ECG makes use of 12 electrodes, in contact with the body, to measure 12 different perspectives on the electrical activity of the heart. The different electrodes are placed in different positions on the body as described in Figure 2.1 with the transverse leads (V1-V6) placed directly on the chest, whilst the frontal leads (I, II, III, aVL, aVF, aVR) are located across the limbs [19].

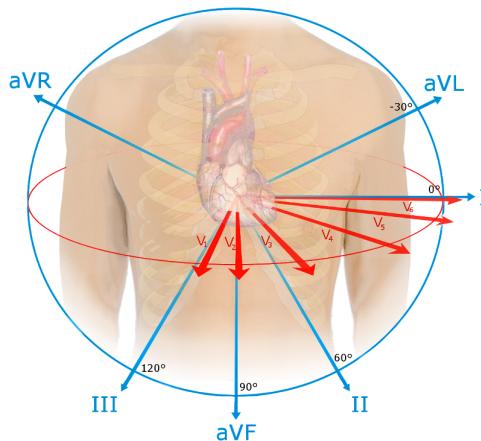


FIGURE 2.1: Graphic of the location of transverse (RED) and frontal (BLUE) electrodes on the body when measuring the 12-lead ECG [20].

A single ECG channel can be visualised graphically, as illustrated in Figure 2.2, the ideal ECG shows both P and T waves as well as the QRS complex. The duration, magnitude and shape of these features allow medical professionals to give diagnoses from ECG signals [21]. Whilst this is the current standard, as mentioned in chapter 1, the visual analysis of these ECGs can be a time demanding process requiring considerable expertise. In addition to this, noise in the signal can often obscure these key features meaning diagnoses may be prone to error. This has led to the development of softwares to aid medical professional through the computational analysis of ECG signals.

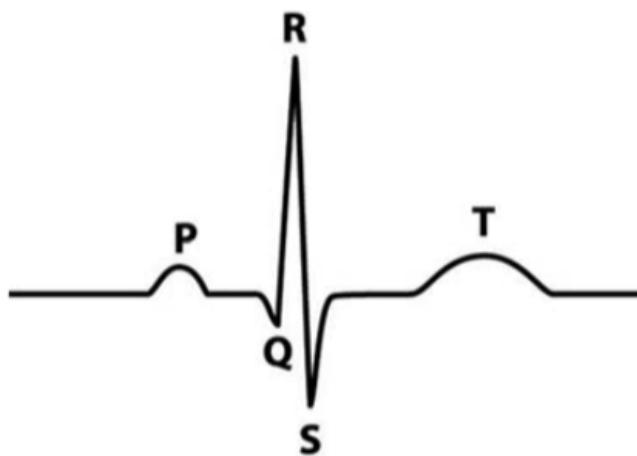


FIGURE 2.2: Ideal ECG signal with the location of key features indicated. The QRS complex is given by the time between the onset of the Q wave and the end of the S [22].

Though the specific techniques used to perform such analysis are constantly adapting, the underlying idea that measuring differences in complexity and parameters governing heart rhythms can lead to heart defect diagnoses has remained consistent. These ideas have conglomerated from numerous papers with the most notable of these including L. Glass' study of the dynamics of cardiac arrhythmias where the non-linear complexity of the heart's dynamics was investigated [7] as well as A. L. Goldberger who analysed the successes of different dynamical models of the heart before suggesting that an older or diseased heart may experience losses in complex variability [6]. The origins of complexity within the heart have been thoroughly investigated since then, and the many processes contributing to this overall behaviour are comprehensively examined within [8].

2.2 PREPROCESSING

The raw ECG signal is prone to noise from a number of sources. The patient moving, breathing activity, electrode contact noise and the current flowing within the cables can all contribute to a poor signal-to-noise ratio (SNR). This results in features such as baseline wander and ECG distortion within the signal [23]. These features are evident when comparing Figure 2.2 with the raw ECGs from the dataset shown in Figure 3.2. It is imperative to reduce these effects in order to improve the signal quality and allow accurate feature

extraction. This is a non-trivial task as often the frequency distributions of some noise components overlap with those of the desired ECG signal, for example low-frequency noise such as baseline wander may coincide with the low-frequency aspects of the ECG signal [24]. Therefore the problem develops into a trade-off between the removal of noise whilst maintaining the key fundamental structure of the ECG signal.

The most basic methods seen in the literature only solve the problem of baseline drift. For example, through fitting a polynomial to the data, where there appear to be no peaks, an estimate of the shape of the baseline skew can be made. Through subtracting this from the data it is possible to remove this baseline. This can be seen in [25] where a polynomial of order 20 is used to remove the baseline from the signal. Although achieving accuracies >90% in their machine learning model, this method does nothing to reduce the high-frequency noise in the signal. In addition to this, the use of such a high order polynomial may lead to over-fitting of the signal meaning that features extracted later may not be fully representative of the underlying ECG.

Another method prevalent in the literature is the Butterworth filter [26], [27] and [28]. The Butterworth filter is capable of performing both low and high band pass filtering to remove high and low frequency noise respectively. The success of this filter is governed by its hyper-parameters, namely the cut-off frequency and the order of the filter used. Whilst the purpose of the cut-off frequency is evident, the order of the filter determines the rate at which the frequency falls off outside the cut-off frequency [29]. It is not apparent from the literature what values of hyper-parameters give the best results for an ECG signal. For example, in [26] an 8th-order low-pass filter with a cut-off frequency of 35Hz is adopted. This removes the high frequency noise however it retains the low-frequency baseline wander. In contrast to this, within [27] a second order Butterworth band filter using cut-off frequencies of 1 and 40 Hz is implemented, removing both high and low frequency components. Both papers resulted in model accuracies of >90%.

An additional frequently used method is the discrete wavelet transform (DWT). The DWT sequentially uses high and low band pass filters to separate a signal into multiple discrete frequency bands. The signal can then be reconstructed through an inverse DWT using the desired bands whilst the undesired bands, presumed to contain noisy frequencies, are discarded. Choosing the correct coefficients to keep and discard is important, if not done correctly, the reconstructed signal may not properly convey the key information needed from the inputted signal, such as its R-peak [30], or there may still be large noise amplitudes present in the signal. The components to be selected can be chosen through hard or soft thresholds. Hard thresholds set the undesired coefficients to zero and soft thresholds use a function to determine the amplitude of each coefficient used in the reconstructed signal. Hard thresholds often result in a better approximation but can cause oscillations in the reconstructed signal. Conversely, soft thresholds typically produce a smoother reconstructed signal but tend to have a larger error [24].

Although there are numerous techniques being developed, the Butterworth filter and DWT are still the most commonly used methods to solve these problems ([30], [31], [32],

[33] and [34]).

2.3 FEATURE EXTRACTION

Whilst simultaneously allowing investigation of the effect of individual parameters on classification accuracy, the reduction of the dimensions of the ECG signal into a discrete set of features is vital to the SVM procedure. The extraction of features from ECGs can be split into those from time, frequency and non-linear domains. Although there are instances where a single parameter is effective at the classification task [25], the majority of the literature uses a combination of features, [35], [36], [37] and [38]. Time-domain features are highly reliant on the accuracy of the extraction which is subsequently reliant on the quality of the signal. In contrast, frequency domain features are less prone to error but have a high feature dimension space. This results in increased computational cost due to the increase in quantity of training data, as well as potential issues in classification as excess data may contain irrelevant information. Non-linear features are also hindered by a greater computational cost, however they can contain the most information relating to the underlying complexity of the signal [27]. Subsequently, studies have shown that nonlinear methods can be more accurate than linear methods for the problem of ECG classification [39].

Frequently used and successful features have been collated and are presented in table 2.1. These will be further explained within chapter 3.

2.4 FEATURE SELECTION

The greater number of features that can be extracted can lead to detailed information about the underlying ECG signal. However, an increased number of features can lead to increased computational cost as well as an increased risk of overfitting meaning the model developed will not generalise well to unseen circumstances. Therefore it is vital to ensure the features passed into the model are only the most important, in particular those providing the greatest potential to differentiate between the healthy and unhealthy classes.

Typically, dimensional reduction is done through principal component analysis (PCA), which transforms the feature set onto an artificial latent variable space [43]. Although this includes all of the extracted features it loses the specific information of each feature, wasting this prior information. Other commonly used methods include filters that use statistical methods to rank the features allowing the strongest performers to be selected and those that are highly correlated to be removed. This has the benefit of being a computationally efficient test. However, as the selected features are not based on the model being used, the final accuracy may not be optimised [44]. Embedded methods include the classification model in the feature selection process, these typically involve the use of Lasso (L1) regularisation for classification using a linear kernel [24]. L1 regularisation promotes sparsity meaning that many of the linear coefficients are pushed towards zero. This allows the relative importance

Domain	Parameter
Time Domain	Mean RR intervals
	Std RR intervals
	Mean amplitude of R peak
	RMSSD of RR intervals
	pNN50 (%)
	Mean QRS duration
	Std QRS duration
	Signal moments
Frequency Domain	Absolute power of LF (0.04 - 0.15 Hz) band
	Absolute power of HF (0.15 to 0.4 Hz) band
	LF/HF ratio
	Total power (0-0.4 Hz)
Non-linear	DWT coefficients
	Higuchi fractal dimension
	Multifractal box-counting dimension
	Std of Poincaré - (SD1, SD2)
	Ratio of SD1, SD2
	Shannon entropy
	Sample entropy

TABLE 2.1: List of commonly used parameters within heart rate variability analysis [40] [41][42].

of each feature to be seen, with features with zero coefficients being discarded. This method however is limited by its reliance on the linear kernel which may not be the optimal kernel for the feature space and so again may not lead to maximal accuracy [24]. This is not the case for wrapper methods which can be used with any model paradigm and so these appear frequently in the literature. These include forward-based approaches, that add the best performing feature to the feature set, backwards-based that remove the worst performing feature as well as recursive feature elimination where features are recursively pruned based on their importance until the desired number of features remain. The performance of each feature can be measured using the same metric as used in the ultimate classification scheme allowing features to be chosen that optimise the performance metric. A negative outcome of wrapper methods is that they have a high time complexity and so are not suitable for high dimensional problems. These methods have been successful within papers such as [45] where a forward-based approach reduced the number of features inputted into the ML model from 71 to 9 features, whilst increasing the accuracy from 79% to 90%. Further attempts to retrieve maximal prediction accuracy have used optimisation methods such as

genetic algorithms or particle swarm to select the best subset of features [12], [46].

2.5 MACHINE LEARNING

Traditionally, classification of ECGs has been done through the manual calculation of parameters representing the signal, subsequent classification of these parameters and thus the signal. This has been done through machine learning paradigms such as KNNs, decision trees, random forests and SVMs ([9], [15], [24] and [47]). Each method has its own advantages and disadvantages. KNN provides a relatively simple approach to divide the feature space into multiple labels based on a distance metric. However, KNN struggles with unbalanced datasets often favouring predictions for the majority class [48]. This problem is common also with decision tree and random forest methods. A decision tree approach continuously splits the data into branches based on the most frequent class in the training data or a similarly weighted target function. This can often lead to bias towards the majority class when working with unbalanced data sets. Although the random forest may alleviate some of this bias due to averaging the results of all the decision trees, if the majority of the decision trees show some bias then it is likely that the random forest model will also [47]. The SVM method is more adept at dealing with this issue, the dataset and objective function can be weighted to penalise bias towards the more frequent class allowing more accurate predictions to be made. In addition to this, the ability to use different kernels makes the SVM highly generalisable to different patterns, linear and nonlinear, as well as those with high feature dimension. This has made the SVM method a very popular approach when dealing with typically unbalanced, nonlinear samples from within ECG databases. The drawback of the SVM method is that the importance of each individual feature is somewhat obscure [47].

In recent years the development of neural networks (NN) has allowed many studies to skip the arduous step of manually parameterising time series signals. This not only means that this expert knowledge is not needed but allows the development of end-to-end models that take advantage of the robustness of variants of the Deep Neural Network (DNN), (a neural network with more than one hidden layer) such as CNN and Recursive Neural Networks (RNNs) to input the high-dimensional complete datasets. Whilst these methods may be more computationally expensive, many studies show better accuracies when comparing these deep learning models with a simpler SVM [49], [50]. The CNN is particularly well suited for time series problems [51], and whilst CNNs are typically used for 2D data such as images, recently one-dimensional variants have shown success within applications involving 1D signals [17]. This success stems from the CNN's inherent abilities, such as its proficiency in local feature detection, which enables the identification of trends and patterns throughout a time series. Additionally, its temporal and translational invariance allows these patterns to be detected and compared across anywhere within the time series [52], [53]. In the context of ECG analysis, both the 2D and 1D variant of the CNN have been employed to high degrees of success [13],[54].

There are some disadvantages to these end-to-end models. The absence of prior knowledge in selecting appropriate parameters means their success relies heavily on the quality of the database. In addition to this, deep networks (those with many layers) are prone to issues of overfitting and interpretability. This means that the model will not generalise well to unseen data and is hard to modify due to its black box nature [24]. Also models such as these generally require a substantial amount of data to infer patterns and trends from them. This means that they may not be applicable to all datasets.

These issues are particularly apparent in cases where the specific disease type is important. In many ECG databases each specific disease may only appear rarely within a very large dataset. An emerging technique to deal with this problem is the prototypical neural network. The prototypical neural network creates a prototype vector of each class and then makes predictions based on a distance metric, often the Euclidean distance, of the inputted data to each prototype [26], [55].

2.6 MODEL VALIDATION

		PREDICTION	
		POSITIVE	NEGATIVE
TRUTH	POSITIVE	TP	FN
	NEGATIVE	FP	TN

FIGURE 2.3: Image of prediction vs truth table used to categorise predictions made by a binary classification model. The table containing the quantity of each category of prediction is known as the "confusion matrix".

A binary classification model's outcomes can be grouped into four distinct groups. These are shown in Figure 2.3 where true positive (TP) and true negative (TN) represent the number of correct positive and negative predictions respectively and false positive (FP) and false negative (FN) represent all the incorrect positive and negative predictions. To test the ability of a model to make accurate predictions there are a number of different metrics that are used. Many of the metrics are built upon these four categories. The most basic, accuracy, is defined as shown in equation (2.1):

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} [56]. \quad (2.1)$$

This is a valid metric for balanced datasets, however skewed sets can often give high accuracy scores even when the model is not performing correctly. For example, if the dataset is heavily skewed towards positive classes, the model may be biased and predict only positive classes.

In this scenario the model will still have a high accuracy despite having incorrectly predicted all of the negative classes. Consequently, there are a variety of metrics that can be used to thoroughly validate and quantify the performance of a model. Many of these metrics revolve around sensitivity, i.e., true positive rate $\frac{TP}{P}$ and specificity, i.e., true negative rate $\frac{TN}{N}$, where P and N are the total number of positive and negative cases respectively. For example, balanced accuracy, often useful for skewed data sets is given by equation (2.2):

$$\text{balanced accuracy} = \frac{\text{sensitivity} + \text{specificity}}{2}. \quad (2.2)$$

In addition to this metrics such as the area under the ROC curve are one of the most widely used for binary classification problems. The metric demonstrates the probability that a classifier will rank a randomly chosen positive sample higher than a randomly chosen negative sample. It is evaluated by plotting the false positive rate $\frac{FP}{N}$ against the sensitivity and then calculating the corresponding area-under-the-curve (AUC). The higher the resulting AUC the better the discrimination power of the model, with perfect classification models scoring an AUC of 1. Other metrics include the F1 score, presenting a trade off between precision, $\frac{TP}{TP+FP}$ and recall, $\frac{TP}{TP+FN}$, as shown in equation 2.3:

$$F1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}, \quad (2.3)$$

as well as the confusion matrix as described in Figure 2.3 [56]. Often studies will use a combination of these metrics to evaluate the model in all areas and ensure consistency. For example, within [25] the accuracy is defined as a linear combination of the specificity with the sensitivity, ensuring both classes are predicted with a reasonable level of accuracy.

3 METHODS

All analysis performed in this report was done on data from the Physionet Physikalisch-Technische Bundesanstalt (PTB) Diagnostic ECG database [57]. The outline of the methods employed is illustrated in Figure 3.1, where it can be seen that the SVM will deal with the parameterised signals whereas the CNN applied will take the complete signals as inputs, before the success of both models is evaluated.

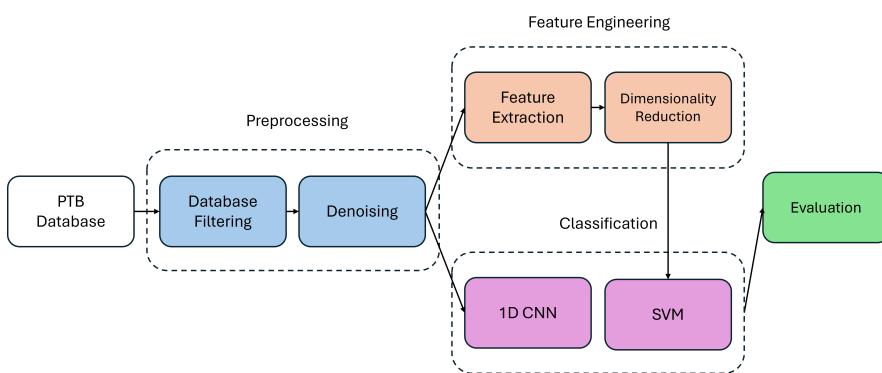


FIGURE 3.1: Flowchart showing the order of events and structure of the methodologies.

All of the methodologies described in this chapter were implemented within Python. The code can be viewed on GitHub as desired, at: <https://github.com/patcourts/ES98C-ECG-Project>.

3.1 DATABASE FILTERING

The Physionet PTB ECG data base contains 290 subject files with 1-5 records for each subject totalling 549 different ECG records. Each record contains the typical 12 lead ECG time series as described in section 2.1. Each subject file also contains a header file describing the available metadata for the patient. An example metadata file for patient 001 is shown in Appendix A for the reader's interest. The contents of each file vary per patient but most contain information on age, gender and their diagnosis. Health information does not exist for 22 of the subjects but of the remaining 268, 52 are labelled as healthy with the diagnoses of the remaining patients distributed as shown in table 3.1.

To prepare the data for analysis it is important that incomplete entries are removed from the dataset. Incomplete data can lead to inaccurate results or biases in the analyses and so should be removed. This includes patients whose health state is unknown, as this is

obviously needed for later classification, as well as those whose age is not stated, as this is a potential covariate to be investigated later. In addition to this, to ensure that each signal has enough samples for later analysis, patients whose ECG's do not meet the minimum time required of 1 minute were discarded. Many of the frequency domain techniques discussed in section 3.3, along with some of the non-linear ones, require a signal length longer than this threshold so that there are enough samples within them for the method to be applicable. Finally, to standardise the dataset and ensure there is no bias, only the first ECG from each patient will be kept.

Diagnostic Class	Number of Subjects
Myocardial infarction	148
Cardiomyopathy/Heart failure	18
Bundle branch block	15
Dysrhythmia	14
Myocardial hypertrophy	7
Valvular heart disease	6
Myocarditis	4
Miscellaneous	4
Healthy controls	52

TABLE 3.1: Distribution of the diagnoses of patients within the PTB diagnostic ECG database.

Although each record contains a 12 lead ECG, in line with the analysis performed in [25], only the v1-v6 leads will be taken forward to analysis. This is done as the v1-v6 leads are closest to the chest and so therefore should have the smallest SNR allowing more accurate analysis. Each of these 6 leads will be tested based on the ECG quality check provided by Neurokit2 [58]. This check uses the method proposed in [59] to qualitatively class the quality of the ECG signal into: 'unacceptable', 'barely acceptable' and 'excellent'. Here the signals scoring 'unacceptable' will be removed from subsequent analysis whereby any patients with no acceptable signals will also be removed. Examples of signals that fall into each category are shown in Figure 3.2.

3.2 DATA PREPROCESSING

In line with section 2 both the Butterworth filter and DWT transform will be implemented to provide options for the data preprocessing methods as well as a potential area for comparison. The outcomes of the applications of the two methods on an example signal can be seen in Figure 3.4.

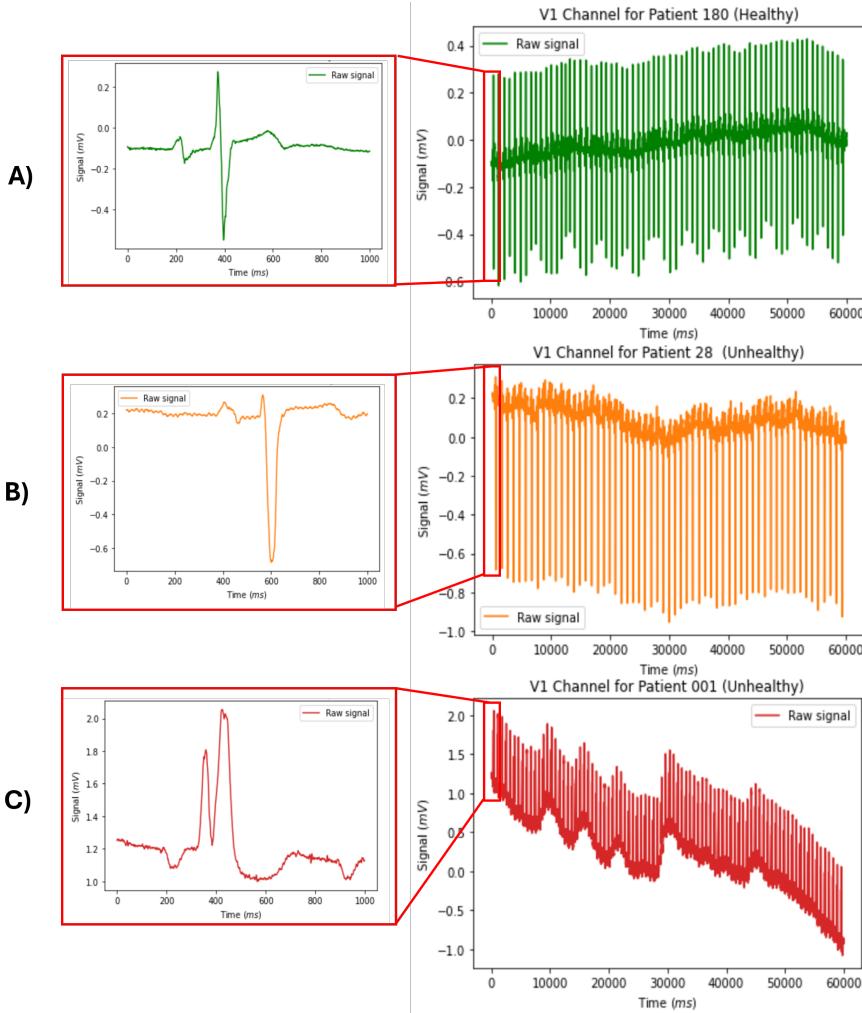


FIGURE 3.2: 'Excellent', 'Barely Acceptable' and 'Unacceptable' qualities of raw data, labelled as A), B) and C) respectively. Quality determined by the method proposed in [59].

3.2.1 BUTTERWORTH

The range of useful frequencies for an ECG is considered to be between 0.5 to 45 Hz with frequencies below 0.5 Hz being attributed to baseline drift and those greater than 45 Hz considered as high frequency noise [60], [61]. Therefore, both a low and high pass Butterworth filter should be used to filter out the respective noise frequencies. In line with the literature, the cut-off frequencies are typically placed just within the desired ranges. Therefore a bandpass filter with cut-off frequencies of 1 - 40 Hz will be applied to the signals. As the optimal value for the filter order is not evident within the literature, through experimentation a filter of order 5 was deemed most appropriate.

3.2.2 DISCRETE WAVELET TRANSFORM

The DWT sequentially uses high and low band pass filters, g and h , to separate a signal into discrete frequency bands. The band pass filters are normally built upon each other such

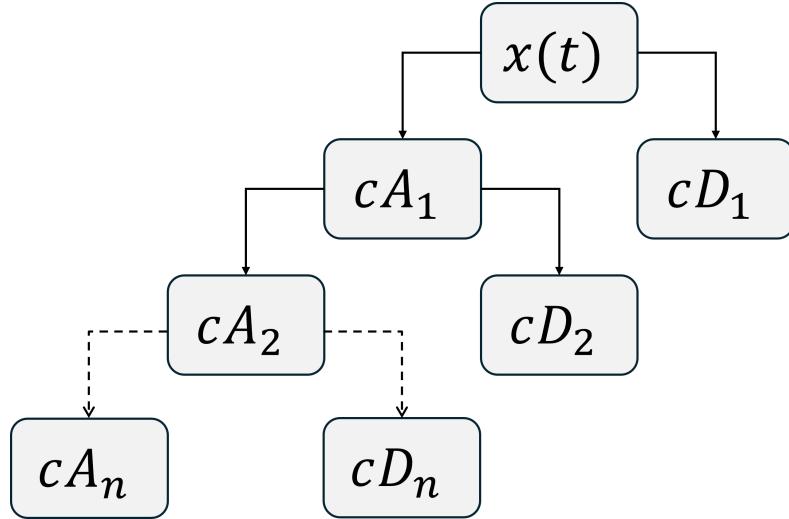


FIGURE 3.3: A schematic showing the process undertaken by the DWT to the n^{th} order. The raw signal, $x(t)$, is split into two frequency bands. The high frequency components are inverse transformed to produce the first level detailed coefficients, cD_1 whereas the low level components, the approximation coefficients cA_1 , are further split till the desired level.

that

$$g_k = (-1)^k h_{n-k-1}, \quad k \in [0, \dots, n-1], \quad (3.1)$$

where n denotes the length of the filter. These band pass filters are applied iteratively to split the signal into high and low frequency components at each stage of a multi-level filter. The low frequency components are labelled as approximation coefficients whereas the high frequency components are wavelet transformed to produce detailed coefficients. This process is described schematically in Figure 3.3.

A signal that has been sampled at f Hz can have a maximum frequency of $f/2$ Hz, the Nyquist frequency. Therefore, application of one level of DWT produces low frequency components below $f/4$ Hz, and high frequency components from $f/4$ to $f/2$ Hz. The low frequency components are the first approximation coefficients, A_1 , whereas the high frequency components are wavelet transformed to produce the first detailed coefficients D_1 . The wavelet used to perform this convolutional transform can determine the success of the process. Often the wavelet most similar to the signal is used with commonly applied wavelets being Haar and Daubechies. For ECG signals, the Daubechies4 (DB4) wavelet is most frequently used with a central frequency factor F_c , (representing the similarity of the wavelet with the ECG signal between 0 and 1) of 0.7 [33], [62]. The process can then be repeated on the lower frequency band until the desired level is reached. The maximal level of decomposition, L is given by,

$$L \leq \log_2(N), \quad (3.2)$$

where N denotes the length of the signal [62]. Once the signal has been split into approximate and detailed coefficients, a selection of these is used to reconstruct the original signal through application of an inverse DWT. The components to be selected can be chosen

through hard or soft thresholds, where hard thresholds set the undesired coefficients to zero and soft thresholds use a function to determine the amplitude of each coefficient. This allows the removal of both high and low frequency components from the signal whilst reconstructing the key descriptive features. The levels at which the hard threshold can be applied are decided through

$$F_a = \frac{F_c F_s}{2^a}, \quad (3.3)$$

where F_a represents the cut-off frequency at each scale, a and F_s is the sample frequency [9]. For the PTB Diagnostic ECG database the sampling frequency, F_s , is 1000 Hz. Using the desired ECG frequency range in tandem with equation (3.3) to remove the baseline drift all coefficients above level 9 should be discarded, whereas to remove high frequency components, all coefficients up to level 4 should be discarded. This results in a reconstructed signal just using the coefficients 5, 6, 7, 8 and 9. The reconstructed signal is compared with the signal denoised through the Butterworth filter in Figure 3.4.

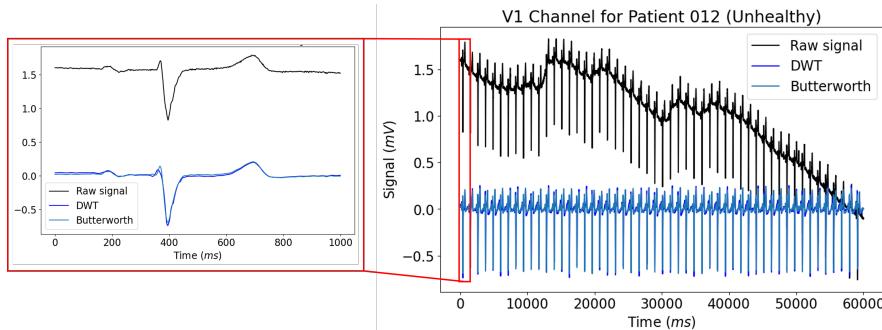


FIGURE 3.4: Comparison of the Butterworth filter and DWT as methods to remove noise from the signal. It can be seen that there is minimal difference between the two methods in the enlarged section.

3.3 FEATURE EXTRACTION

As discussed in section 2, the SVM requires the reduced dimension parameterised signals. Thus features are extracted from the denoised signals pertaining to three subsets: time, frequency and non-linear domain. The specific parameters calculated can be seen within table 2.1.

3.3.1 TIME AND AMPLITUDE FEATURES

3.3.1.1 MOMENTS

The moments of a time series provide statistical information about its shape and characteristics. Moments include the signal's mean, standard deviation, skew and kurtosis. These are straightforward to calculate using built-in functions within the Python package: *Numpy* [63].

3.3.1.2 R PARAMETERS

These parameters rely on the ability to identify the locations of specific instances in the signal, namely the R peak and its duration, given by the QRS interval in an ECG signal. There are a number of packages that help with this problem. The most straightforward, is the *Scipy 'findpeaks'* algorithm which can help identify peaks within a signal within pre-specified ranges. However, the range at which peaks occur is different for each signal and usually not known without previous analysis. This leads to inaccuracies with false peaks being identified and some being missed. Consequently, packages such as *Neurokit2* offer several more advanced algorithms to solve this specific problem. The algorithm used here detects the QRS complex by measuring the steepness of the absolute gradient of the signal before finding the R peak as the local maximum within the given QRS complex [58]. This allows calculation of the RR intervals from the difference between the location of each successive peak and therefore the mean and standard deviation of the RR intervals for each signal. The amplitudes, root mean sum of the squared differences (RMSSD) and pNN50 which represents the percentage of the total RR intervals that are greater than 50ms, are also then trivial to calculate.

3.3.2 FREQUENCY DOMAIN FEATURES

3.3.2.1 DWT

The DWT is further used to calculate parameters for potential classification. The signal is decomposed to the maximal level as given by equation (3.2). The means and standard deviations of each coefficient are then a potential indicator of the class of the signal.

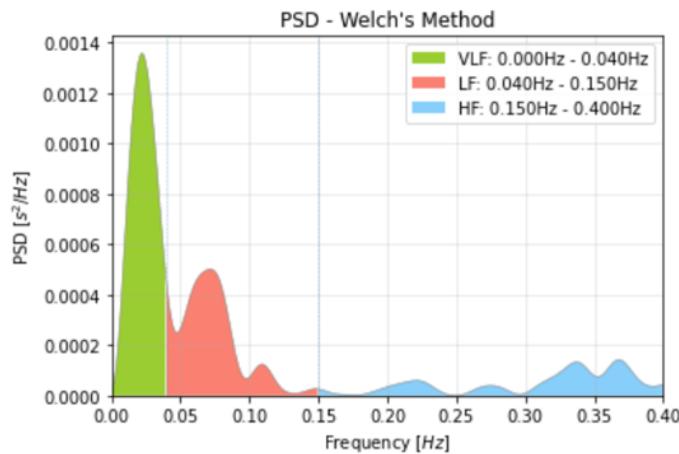


FIGURE 3.5: The power spectral density (PSD) for an unhealthy patient example within the database. Figure generated using the *Biosppy* package from within *Neurokit2* [58].

3.3.2.2 POWER SPECTRUM

The power spectrum of a signal is calculated through Welch's method. Welch's method takes advantage of the fast Fourier transform (FFT) to allow estimation of the power spectral density by splitting the data into overlapping segments called windows. A modified periodogram is taken from each window and these are subsequently averaged [64]. The success of this method is determined by the choice of window and its subsequent overlap. An overlap of zero is equivalent to Bartlett's method [65]. Here the Hann window is used, in line with common practice, as well as a 50% overlap, commonly used as it provides a good trade off between computational complexity and accurate power estimations for the Hann window [66]. The resulting power spectral density can then be integrated over the relevant ranges to find the power within that band. An example power spectral density (PSD) for a signal representing an unhealthy patient is shown in Figure 3.5.

3.3.3 NONLINEAR DOMAIN FEATURES

3.3.3.1 SIGNAL EMBEDDING

Many non-linear domain features require the use of vectors embedded from the signal. Thus it is useful to describe the process of generating these vectors. A time series, X , where X contains a series of consecutive points, i.e., $X = [x(i), i = 1, \dots, N]$, has embedded vectors defined as

$$X_m^\tau(i) = (x(i), x(i + \tau), x(i + 2\tau), \dots, x(i + (m - 1) \cdot \tau)) \quad (3.4)$$

where $i = [1, \dots, N - (m - 1) \cdot \tau]$ [42]. Here m is the embedding dimension and τ represents the time lag. The choice of the embedding dimension is critical for the success of the subsequent methodologies although often varies between methods. The optimal value of τ is given by the point at which the autocorrelation (the correlation of the signal with itself) drops to $\frac{1}{e}$ of its initial value. This $\frac{1}{e}$ threshold is commonly used as a balance between retaining dynamical information whilst avoiding excessive correlation that can lead to poor embedding quality [42].

3.3.3.2 SHANNON ENTROPY

The Shannon entropy was proposed as a method to quantify the average rate at which information in a signal is produced from a stochastic source. It is essentially a measure of randomness or unpredictability [67]. The Shannon entropy, (SE), of a discrete random variable, X is calculated through

$$SE(X) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i) \quad (3.5)$$

where $p(x)$ represents the probability distribution function of x .

3.3.3.3 SAMPLE ENTROPY

The Sample entropy (SampEn) was introduced specifically for physiological time series [68]. It quantifies the level of randomness and complexity within a time series, and so has particular applications to ECG signals. The SampEn method relies on the correlation integral C , defined in equation (3.6):

$$C_m^r(i) = \frac{1}{N-m+1} \sum_{j=1}^{N-m+1} RM_{ij}. \quad (3.6)$$

Here RM_{ij} is a matrix containing values relating to the similarity of two separately embedded vectors, $X_m^r(i), X_m^r(j)$ as given by equation (3.4). The Euclidean distance, given by equation (3.7),

$$d(X_m^r(i), X_m^r(j)) = \sum_{k=1}^m (X_m^r(i, k) - X_m^r(j, k))^2, \quad (3.7)$$

of the two vectors is compared with a tolerance interval, r . If the two embedded vectors are within the interval then $RM_{ij} = 1$, if not then $RM_{ij} = 0$.

The SampEn statistic is then calculated through

$$\text{SampEn} = -\ln \left[\frac{\sum_{i=1}^{N-m} C_{m+1}^r(i)}{\sum_{i=1}^{N-m} C_m^r(i)} \right]. \quad (3.8)$$

This leads to two parameters that determine the success of this method, namely, the embedding dimension m and the tolerance, r . In accordance with previous studies, r is set to $\frac{\text{std}}{5}$ where std represents the standard deviation of the signal [69], [70] and m is set as 2 [71].

3.3.3.4 POINCARÉ PLOT

A Poincaré plot (Pplot) provides a graphical representation of the self similarity within a time series. It can be used to distinguish between chaos and randomness through a simplified phase-space embedding [72]. Within heart rate variability (HRV) analysis, RR intervals are characterised within a Pplot. By plotting each subsequent RR interval against each other, each RR interval is a function of the previous, meaning that each point on the Pplot shows two subsequent heart beats, (RR_i and RR_{i+1}). An example Pplot for an unhealthy subject can be seen in Figure 3.6. The shape of the Pplot is used to draw conclusions about the health state of the patient and this is quantified through the metrics SD_1 and SD_2 representing the standard deviations in each orthogonal direction, as shown in Figure 3.6. Both SD_1 , SD_2 and the ratio between them are commonly used HRV statistics [73].

3.3.3.5 FRACTAL DIMENSIONS

A fractal is a mathematical model used to evaluate self-similar structures. The fractal dimension of such a structure is considered to be the universal indicator of the degree of its irregularity. There are numerous methods and algorithms that can be used to calculate the fractal dimension of a time series such as correlation dimension, box-counting dimension [74] as well as algorithms proposed by Katz [75]. This report incorporates two fractal

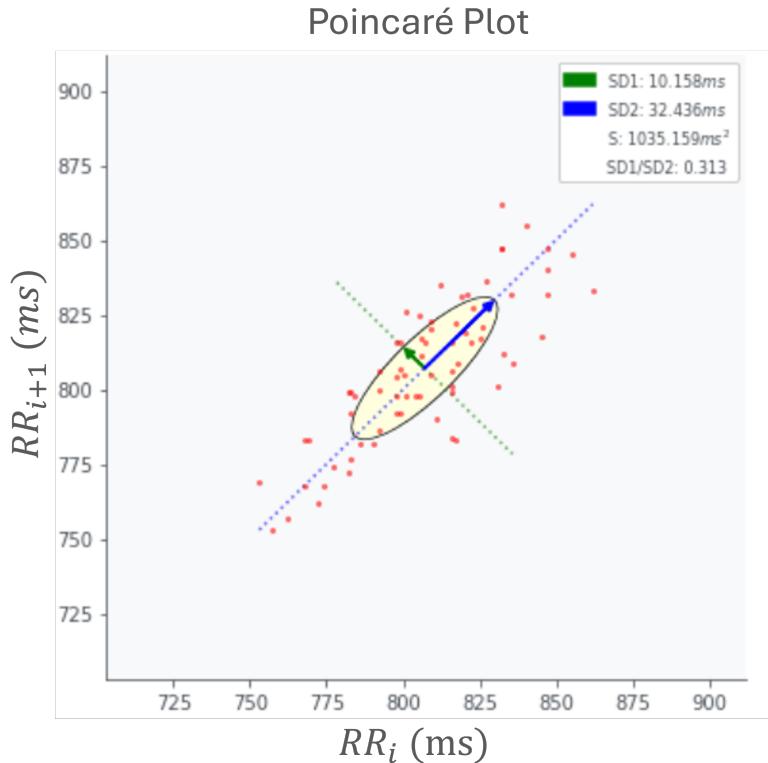


FIGURE 3.6: An example Poincaré plot for a random unhealthy patient in the database. SD1 and SD2 are visualized in green and blue respectively. Figure generated through the *Biosppy* package within *Neurokit2* [58].

dimension measures. The Higuchi fractal dimension is investigated due to its increased use within time series analysis [42] as well as a box-counting multifractal analysis method implemented due to its success with ECG signal classification within [25].

The Higuchi fractal dimension considers the time series as a finite set of observations taken at regular intervals. A new series, X_k^m is constructed from the time series, defined as

$$X_m^{(k)} = \left\{ X(m), X(m+k), X(m+2k), \dots, X\left(m + \left(\frac{N-m}{k}\right)k\right) \right\} \quad (3.9)$$

where $m = (1, 2, \dots, k)$ represents the initial time and k represents the chosen time interval [76]. This results in k new time series. From this new set of time series, Higuchi takes the average length of each time series where the length is defined as

$$L_m(k) = \frac{1}{k} \left(\sum_{i=1}^{\lfloor \frac{N-m}{k} \rfloor} (X(m+ik) - X(m+(i-1)k)) \right) \left(\frac{N-1}{\lfloor \frac{N-m}{k} \rfloor k} \right). \quad (3.10)$$

If the average is proportional to k^{-D} then the time series can be considered a fractal with dimension D [77]. Thus, through a log-log plot of $\ln(L)$ vs $\ln(k)$ the Higuchi fractal dimension can be inferred from the negative gradient of the slope. This method relies on the maximum value of k used, k_{max} , however the optimal value is not apparent. The current literature states that k_{max} should be between 1-10% of the signal length N [78],

however Higuchi used $k_{\max} = 2^{11}$ for a time series of $N = 2^{17}$ [77]. As large values of k_{\max} lead to increasingly longer computational time, a value of $k_{\max} = 50$ was settled on for this investigation.

The second approach implemented within this study aims to quantify the differences in amplitude variations within the dynamical system. This is done through the development of phase-space structures called attractors. The fractal dimension of the embedded attractor is then quantified through the box-counting method.

The embedded attractor is established through first embedding the time series as in equation (3.4) above. In this case an embedding dimension of 4 was used in line with the analysis within [25]. This can be visualised, as shown in Figure 3.7, through the singular value decomposition (SVD) method. SVD involves breaking down a data matrix, A , into three constituents:

$$A = U\Sigma V^T, \quad (3.11)$$

where U and V^T are square matrices that represent the eigenvalues of $A^T A$ and $A A^T$ respectively whilst Σ is a diagonal matrix containing the singular values of $A^T A$. From equation (3.11), Σ allows access to the principle components of the embedded attractor.

Embedded Attractor from SVD

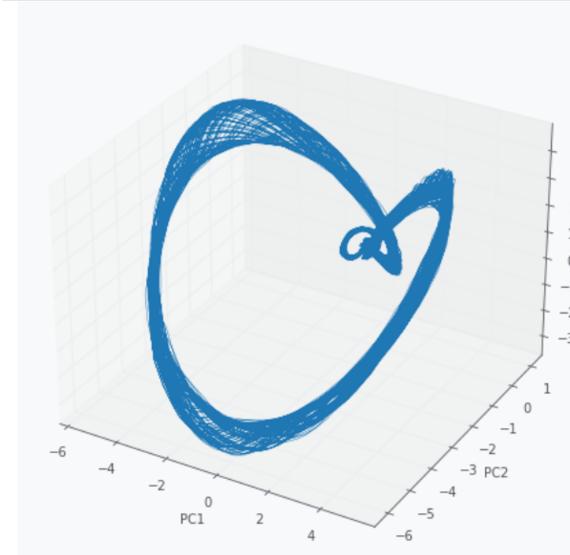


FIGURE 3.7: An example of a 3D embedded attractor for a random patient from the database.

The fractal dimension in 3D space can then be calculated using the box counting method on the first three principle components of the embedded attractor. The box-counting method involves covering the structure with a grid of boxes and counting the number of boxes within which part of the structure resides. A smaller box size leads to a more accurate quantification of the pattern of the structure, however this comes with a trade-off of computational cost. The fractal dimension can then be calculated through:

$$D = \frac{\log(Y)}{\log(z)}, \quad (3.12)$$

where Y is the number of boxes containing the structure and z is the magnification i.e. inverse of the size of the box [74]. To avoid the need to find an appropriate value for z, whilst simultaneously increasing the accuracy of the method, a number of different box sizes can be used. The fractal dimension is then equivalent to the gradient of a plot of $\log(Y)$ against $\log(z)$.

3.3.4 COVARIATES

Due to varying levels of information present within a patient's metadata files, only 3 of the healthy patients in the remaining database have information regarding their smoking status. This meant that this covariate was unable to be used. In contrast, every patient remaining in the database had information about their age when the ECG was taken and so this covariate will be included in the analysis.

3.4 FEATURE SELECTION

In accordance with chapter 2, feature selection is performed through the sequential forward selection (SFS) wrapper method as this allows features to be chosen based on their potential to optimise the desired scoring function. However, as the SFS repeatedly iterates over every feature in the set, an initial filter method will be implemented to remove the features with least potential whilst reducing the size of the feature set for the SFS to chose from. The SFS will then be applied on the already reduced feature set, thereby reducing the computational time of the procedure whilst still selecting only the best performing features.

Although the filter method does not involve use of the specific classification model used later, there is a variety of metrics and models within which the filter method can be employed. In this case the 'mutual information' metric will be used to score each feature where a 'select-K-best' approach will be implemented. The mutual information metric investigates the dependency between the features and the target variable. The test returns a positive value and will only be equal to zero if the two variables are independent [79]. This metric is used due to its ability to handle both linear and non-linear relationships as well as its ability to remove redundant features, reducing the chances of overfitting when training the model. The metric is calculated through the use of KNN distances to provide entropy estimation [80]. To effectively half the size of the feature set, the best 8 features are then passed into the wrapper method. The subsequent wrapper method then evaluates each of these features using the same paradigm used within model training. The features are evaluated based on their performance within the desired scoring metric. To promote positive classifications (overcoming the skewed database), whilst still retaining balance, the F1 score metric will be used to evaluate the features, as seen in equation (2.3). The sequential forward selection is used with the number of desired features set to 4. This has been chosen

as a trade-off between supplying the SVM with enough information to make a classification whilst avoiding over-specification leading to a loss in generalisability of the model. Each iteration of the method selects the feature which promotes the best F1 score, resulting in a subset of 4 features that optimise the metric for each channel. These features are then used to perform the final classification using the SVM.

3.5 MACHINE LEARNING

3.5.1 SUPPORT VECTOR MACHINE (SVM)

The SVM is the ideal candidate for dealing with the skewed dataset considered here, as discussed in chapter 2. The SVM will use the features selected in section 3.4 to classify each patient into 'healthy' and 'unhealthy' class labels. The SVM produces this classification through an attempt to find an ($n-1$) dimensional hyperplane through the n -dimensional data, where the margin is maximised. This means that the distance between the hyperplane and the closest data point of each class is maximal. Given a set of training labels, \mathbf{X}_i, y_i , where \mathbf{X}_i are the feature vectors and y_i are the corresponding labels, the hyperplane can be defined as

$$P_{ij}(\mathbf{x}) = \mathbf{w}_{ij}^T \cdot \mathbf{x} + b_{ij} = 0. \quad (3.13)$$

Here w is a weight vector in the feature space, and b is a scalar representing the bias [11]. Maximising the margin of the hyperplane is then equivalent to minimising the weights, i.e.,

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2. \quad (3.14)$$

In situations where the data may not be perfectly separable, meaning an optimal hyperplane directly splitting the two groups of data is not possible, a fuzzy boundary can be introduced. This is done through the form of slack variables, ζ_i as shown in equation (3.15):

$$\min_{\mathbf{w}, b, \zeta} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \zeta_i, \quad (3.15)$$

where the regularization hyperparameter, C , has been introduced. C controls the trade-off between maximising the margin and minimising the classification error.

The SVM method is particularly suited for non-linearly separable data due to its ability to transform the data into higher dimensional feature spaces through the use of the kernel, coined the 'kernel-trick'. This transformation makes it easier to find linear separations within the data. Commonly used kernel types include linear, polynomial and radial basis functions (RBF). The choice of kernel type is done based on the trade off between accuracy and complexity. However, due to the relatively low-dimensional nature of this particular problem, complexity is not too much of an issue. Therefore, the kernel is chosen based solely on accuracy through the use of a grid search. The grid search is performed prior to the final training of the model in an attempt to obtain some idea of the best hyperparameters, (C and the kernel) that should be used in the final model. The grid search iterates over

each combination the hyperparameters shown in table 3.2 and gives the subset that leads to maximisation of a desired metric, the F1 score.

Hyperparameter	Range of Values
Kernel	[RBF, Linear, Polynomial]
C	[0.01, 0.1, 1, 10]

TABLE 3.2: The range of hyperparameters covered by the grid search to find the optimal combination for use within the SVM.

As there are 6 channels, each with potentially different descriptive features, the grid search is done for each channel so that the model can use the appropriate hyperparameters to maximise the performance for each channel.

To reduce the chance of overfitting, once the optimal hyperparameters have been found they are used to train a model on a different, random subset of the data. A 3-way fold is used to split the data into 3 random subsets. Each subset is then further split into train and test sections with a 70:30 split ratio, as is most commonly used within ML problems with relatively small datasets. A model is trained on each subset's training data and then subsequently tested on the test data. The labels predicted by the model from the test data are then compared with the true labels to obtain a measure of accuracy of the model for each subset. These are then cross-validated with the accuracies achieved in the remaining two subsets to obtain an overall average accuracy. The model accuracy will be determined using a variety of the methods discussed in section 2.6.

In addition to measuring the accuracies of each channel, it is possible to see which combination of channels would provide the best performance. To do this the probabilities assigned to each patient by the model for each channel can be combined and averaged before making predictions. Due to the relatively small number of channels, it is computationally viable to achieve this via an exhaustive search, where every possible combination of channels will be tested to find the best possible set.

All of these calculations will be repeated several times, limited by the time of computation, with averages and standard deviations taken in order to gain an insight into the reliability of and the error on the results.

3.5.2 CONVOLUTIONAL NEURAL NETWORK (CNN)

As discussed in chapter 2, the 2D CNN requires an extensive dataset to achieve generalisable results. Given its lower computational complexity relative to the 2D model, along with its demonstrated effectiveness in handling biomedical signals, the 1D CNN presents a more viable approach for this project and so will be implemented here [51].

The 1D CNN takes the complete ECG signals as its inputs and attempts to classify them through automatically extracting features from the data. The key difference of a CNN with a typical NN is the inclusion of a convolutional layer and the pooling layer. The convolutional layer replaces the matrix multiplication, seen in regular neural networks,

by convolving the input with a convolution kernel [14]. The convolution kernel or filter, is typically a lot smaller than the size of the inputted data and contains a set of learnable weights that are adjusted during training. The filter passes across the whole input space and multiplies the values of the filter with those within the kernel sized section of the inputted values. The results are then summed to produce a singular output at each kernel location across the input space [51]. The pooling layer is then used to help reduce the number of computations whilst keeping the most important learned features. Often characterised as maximum- or average- pooling, the pooling layer uses the maximum or average values of data within a certain section to represent the data in that section. An example can be seen in Figure 3.8 where it can be seen that a pooling layer with a pool size of 2 can effectively halve the number of data points and so therefore reduce the number of calculations needed.

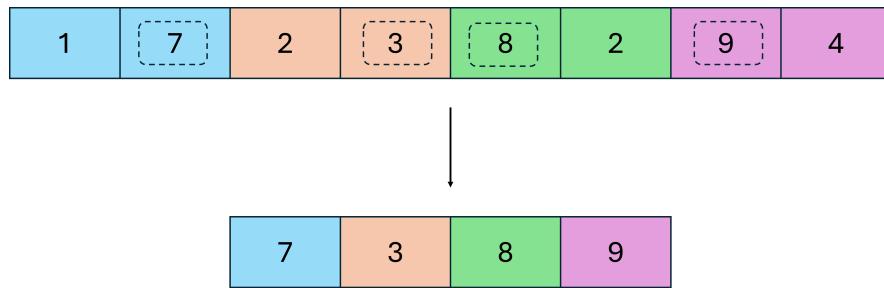


FIGURE 3.8: Outlines how the max pooling layer with a pool size of 2 halves the number of data points and so reduces the computational complexity.

Often the CNN is built up of many blocks with the main layers in each block comprising of a convolutional layer as well as a pooling layer before a final fully-connected layer. The fully-connected layer is implemented outside of the block to combine the information from the preceding layers, and transform this into an appropriate sized, flattened, vector allowing predictions to take place. Given the relatively small size of the database, using multiple blocks could lead to overfitting. Therefore, a relatively shallow model with only one block is proposed. An outline of all the layers within this block is shown in Figure 3.9 with the corresponding Python code included within Appendix B for the reader's interest.

For the proposed model, a convolutional layer containing 16 filters each with a kernel size of 3 is proposed. These values have been chosen mainly through trial and error. Batch normalisation is then applied to the output of the convolutional layer and before the activation function. This is incorporated to normalise the output of the convolutional layer which can stabilise training and lead to quicker convergence [81]. The activation layer is then used to introduce non-linearity in the model, this incorporates a rectified linear unit (ReLU) activation function, as used in the first successful CNN, AlexNet [82]. The ReLU activation function avoids issues of vanishing gradient that can be seen in other activation functions such as a sigmoid or tanh function whilst also promoting sparsity by setting any negative inputs to zero, as can be seen in Figure 3.10. Subsequently, a max pooling layer of pool size 2 is applied, as visualised in Figure 3.8. The minimum pool size is used here due to the already small nature of the dataset whilst the maximum pooling technique is

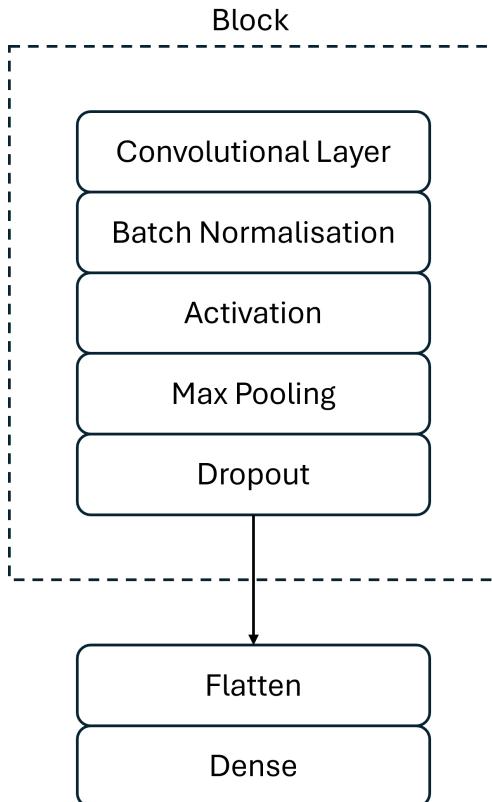


FIGURE 3.9: Outlines how the layers of the CNN are structured.

implemented as it promotes the most prominent features allowing for easier classification [51]. The block finishes with a dropout layer. Here a 0.5 dropout is used to randomly set 50% of the neurons to zero during training. This regularisation technique forces the network to learn more robust features rather than relying on any one neuron and potentially overfitting to the training data [83]. As mentioned, the block can be repeated depending on the desired complexity of the model. Once this is finished, the data are flattened to a single 1D array appropriate for the fully-connected layer or ‘Dense’ layer to combine information and make predictions. Predictions are made using the sigmoid activation function as shown in Figure 3.10. The sigmoid activation function is used here as it returns a value between 0 and 1 which can be interpreted as a probability and pairs well with the binary cross-entropy loss function, as seen in equation (3.16), used within training.

The described model is then trained with the complete ECG signals as inputs. As with the SVM model an initial training split ratio of 70:30 is incorporated, leaving 30% of the data to be used for model validation. In addition to this, class weighting is applied to the data in an attempt to combat the skew in the database. The model training evolves through updating the weights and nodes within the network so as to minimise the loss. The binary cross entropy loss function is used here, as shown in equation (3.16):

$$\text{Binary Cross-Entropy Loss} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]. \quad (3.16)$$

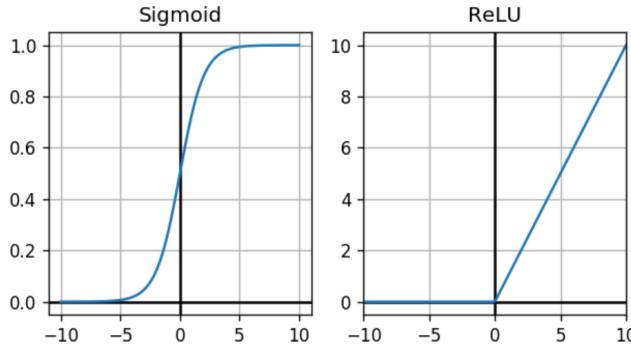


FIGURE 3.10: Figure showing the difference between the ReLU and sigmoid activation functions.

Here, N is the number of samples, y_i is the true label of the i -th sample (0 or 1) and \hat{y}_i is the predicted probability for the same sample, as given by the sigmoid function. This is the appropriate loss function as it can effectively measure the distance between true and predicted binary labels, applying greater penalties to less accurate predictions. The model performance can then be improved by minimising this loss function when optimising the weights within the CNN. This is done through the Adam optimisation algorithm due to its popularity within NN optimisation problems [84].

In addition to dropout, the early stopping regularisation technique is included within the training process to further reduce the chance of overfitting. Early stopping monitors the progress of the model, including the training and validation accuracy. If the validation accuracy is stagnating or declining the model will return to the weights that previously gave the best performance. Here a patience of 5 epochs is used to allow suitable time for the model to learn. If the validation accuracy has not increased over 5 epochs, the model is considered to have overfitted and so will be stopped early.

The specific hyperparameters used to create and train the proposed model are summarised in table 3.3. Hyperparameters such as the number of epochs and batch size, not previously mentioned, were chosen through trial and error, to allow the model suitable time and data with which to train.

Hyperparameter	Values
Epochs	20
Batch size	16
Filters	16
Kernel size	3
Loss function	Binary cross-entropy
Dropout	50%
Max pooling	Pool size 2

TABLE 3.3: The hyperparameters used to make and train the CNN used within this study.

The success of the model is determined through the minimisation of the loss function as well as similar accuracy metrics as described in section 3.5.1.

RESULTS

The database filtering described in section 3.1 resulted in a database containing 221 patients, with each patient having from one to six leads attributed to them. The diagnosis distribution of the filtered database is shown in Figure 4.1.

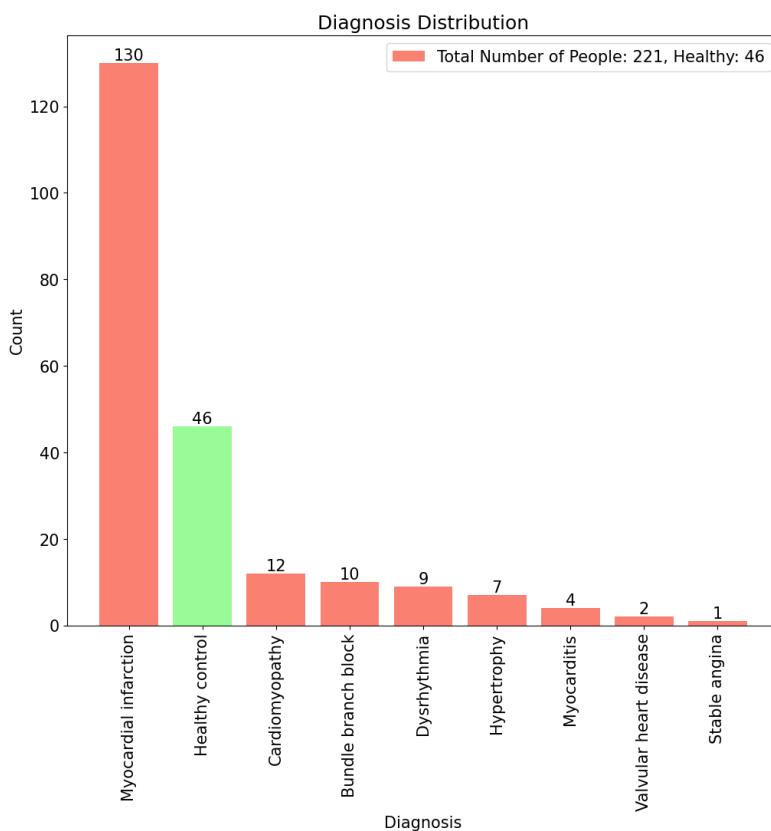


FIGURE 4.1: Bar chart of the number of patients of each diagnosis within the filtered database. There are 46 healthy controls remaining and 175 with various heart arrhythmias.

This chart shows that there is indeed a skewed database as only 46 out of 221 patients are classed as healthy meaning 175 have some form of heart condition attributed to them. In addition to this it can be seen that the overwhelming majority have a Myocardial Infarction, which would lead to further difficulties if attempting to classify specific disease types.

4.1 BINARY SVM

4.1.1 CALCULATED FEATURES

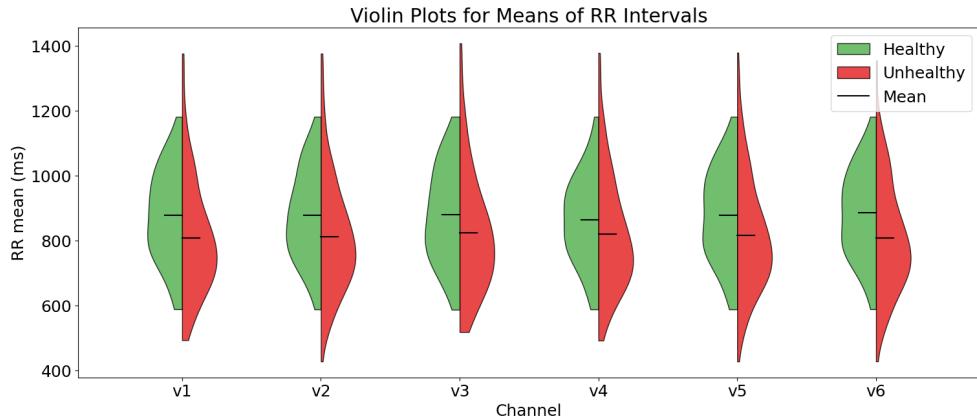


FIGURE 4.2: Violin plot showing the distribution of RR means calculated for healthy and unhealthy patients across each channel.

An example of the resulting values of the mean of the RR interval calculations can be seen in Figure 4.2. This plot highlights the challenges encountered when classifying data based on parameterised values. While it is evident that the mean RR interval for the unhealthy patient is consistently lower than that for the healthy patient across all channels (as is logical), the overlap in data distribution between the two classes is substantial. This overlap makes distinguishing between the two classes a non trivial task.

As the exact values of each parameter calculated is not the primary focus within this report, the difficulty of classification is what this area of analysis aims to illustrate, violin plots depicting the distributions of additional parameters are provided in Appendix C for further insight.

4.1.2 SELECTED FEATURES

By performing the feature selection routine 100 times, the features most important to each channel can be seen. This is expressed in the bar chart in Figure 4.3. From the chart it can be seen that the most commonly selected feature is the age covariate which has been chosen almost every time for each channel. Other commonly chosen parameters include the low frequency contribution from the power spectral density, the RR amplitudes and standard deviation, the second standard deviation of the Poincaré plot and the Shannon entropy. It can thus be seen that features from each domain investigated here are key to making classifications. Another notable aspect of the plot is that the standard deviation of the signals has been consistently selected for channel 4, but not even once for any other channel.

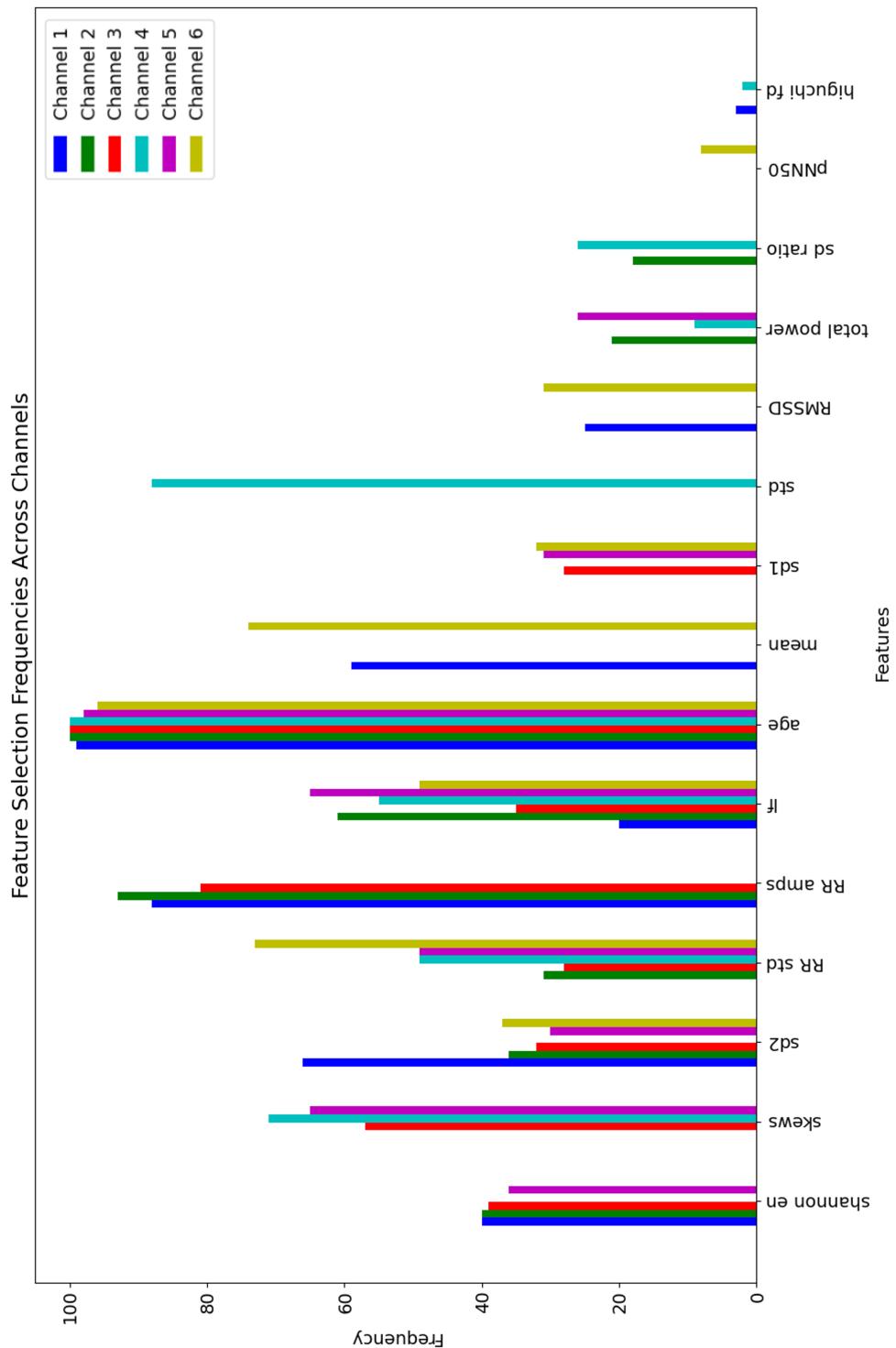


FIGURE 4.3: Bar chart of the features that have been selected for each patient for each channel over 100 iterations of the features selection method.

4.1.3 CHANNEL AND OPTIMISED PERFORMANCE

The accuracy metrics for each channel, with errors averaged over ten iterations, are presented in table 4.1.

Success Metric	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6
F1 score	0.86 ± 0.02	0.86 ± 0.02	0.88 ± 0.01	0.81 ± 0.03	0.81 ± 0.02	0.84 ± 0.01
Balanced Accuracy	0.72 ± 0.02	0.71 ± 0.01	0.71 ± 0.01	0.67 ± 0.02	0.74 ± 0.02	0.73 ± 0.02
Accuracy	0.79 ± 0.03	0.78 ± 0.03	0.81 ± 0.01	0.73 ± 0.04	0.74 ± 0.00	0.77 ± 0.01
Precision	0.88 ± 0.01	0.88 ± 0.01	0.88 ± 0.01	0.86 ± 0.02	0.91 ± 0.01	0.90 ± 0.01
Recall	0.85 ± 0.04	0.84 ± 0.05	0.89 ± 0.01	0.78 ± 0.05	0.74 ± 0.02	0.80 ± 0.02

TABLE 4.1: Scores with standard deviations for each channel, averaged over 10 iterations.

From the table it can be seen that channel 3 has the highest accuracy, at 81%, F1 score, at 88%, and recall, at 89%, whereas channel 5 has the highest precision at 91% and balanced accuracy at 74%. This suggests that channels 3 and 5 are the optimal channels for ECG classification accuracy. Through optimising the combination of channels to achieve the best performance, as described in section 3.5.1, an accuracy of 0.86 ± 0.01 is achieved. This is a strong accuracy that will be further evaluated within chapter 5. The channels used to achieve this optimised accuracy over the ten iterations are shown in Figure 4.4. The figure shows that channels 5 and 6 were used most often whereas channel 3 was only used half of the time. This enforces the suggestion that channel 5 is the optimal channel for ECG accuracy as seen in table 4.1.

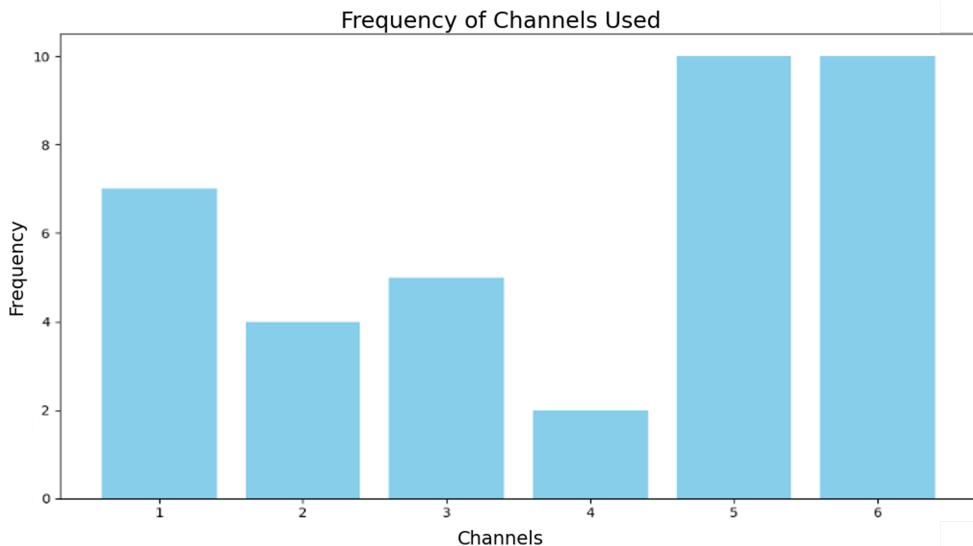


FIGURE 4.4: Bar chart of the channels that have been selected to optimise the accuracy in each iteration.

The ROC curve and confusion matrix for the optimised combinations of channels are shown in Figure 4.5. Figure 4.5 (A) suggests good performance with an area-under-the-curve of 0.83 ± 0.01 . The confusion matrix in Figure 4.5 (B) shows the quantity of each type of prediction. It can be seen that whilst the majority of predictions align with their true labels, almost half of the healthy predictions were misclassified unhealthy signals.

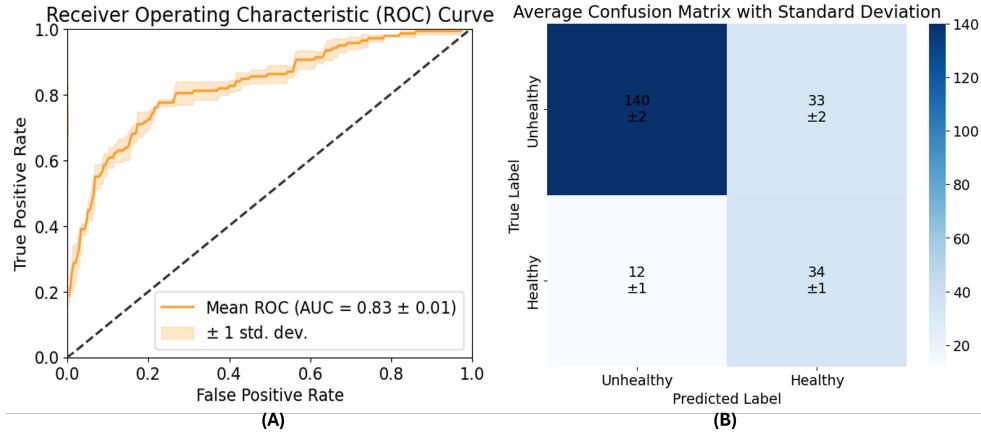


FIGURE 4.5: Averaged confusion matrix (B) and ROC curve (A) for the optimised channels over three iterations of the SVM method.

4.2 BINARY 1D CNN

Initially, training was to be performed on each channel separately allowing comparison of the performance of each channel as performed in the previous section. However, the quantity of data contained in a single channel was insufficient for the CNN to learn effectively and produce meaningful results. Therefore, the training was performed on data from each channel at once with the results as shown below.

4.2.1 MODEL PERFORMANCE

The evolution of the model's performance during training is shown in Figure 4.6. Here it can be seen that the accuracy of the training set leads the validation accuracy which both rise as more epochs are trained. In contrast the loss is reduced as more epochs are trained with both the training and validation tending towards zero.

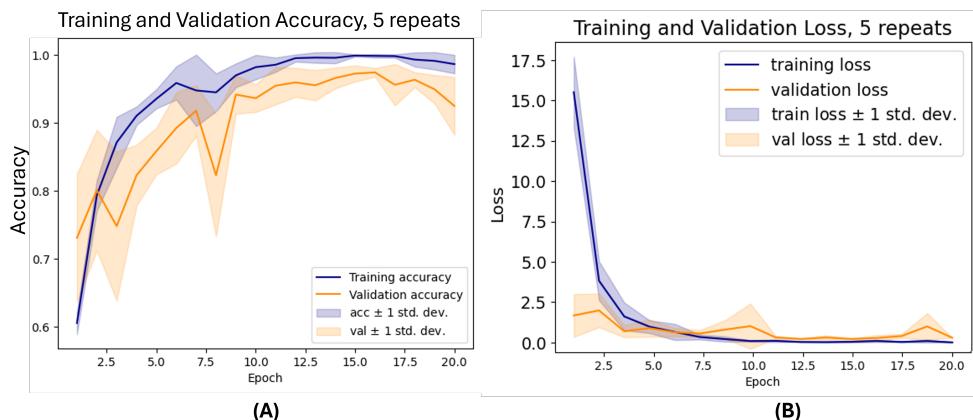


FIGURE 4.6: Graphs to show the evolution of the accuracy (A) and loss (B) with each training epoch.

Within Figure 4.6 (A), the training accuracy appears to plateau at values in the range of 0.98 whereas the validation accuracy peaks at around 16 epochs before trailing off. This behaviour is due to overfitting, where the model has become too specialised to the training data. Early stopping is key in this context, as it allows the final model to revert to the weights and nodes from the peak model at 16 epochs. This ensures that the validated model has optimal applicability and generalisation.

The ROC curve and the confusion matrix for this model can be seen in Figure 4.7.

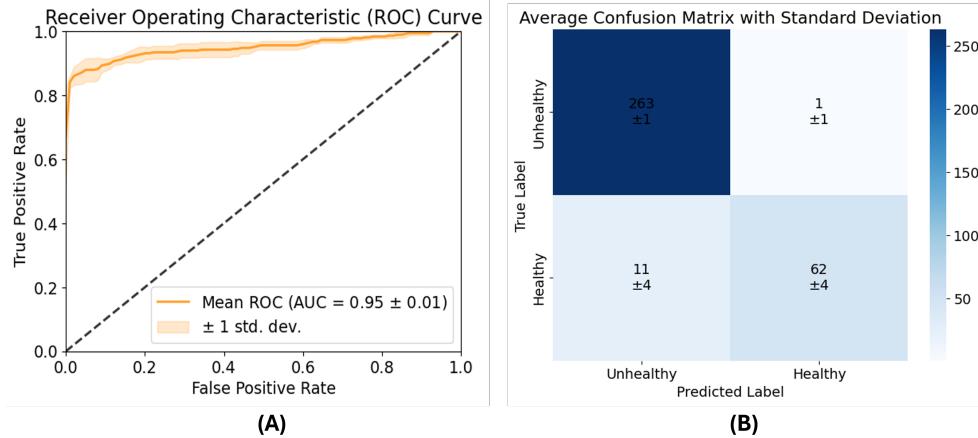


FIGURE 4.7: Average ROC curve (A) and confusion matrix (B) over 5 iterations of the CNN model.

The ROC curve suggests the model has been very successful in terms of the classification with an AUC of 0.95. The small standard deviation shown over the number of repeats shows that this success is repeatable over each iteration. From the confusion matrix it can be seen that there are very few unhealthy labels that have been predicted as healthy (false positives). This means that most of the inaccuracy in the model is from falsely predicting healthy labels as unhealthy (false negatives). However, the model remains effective, with a total average of only 12 misclassifications.

The specific accuracy metrics describing the performance of the CNN model are displayed in table 4.2. This table shows the metrics on the predictions of the positive, (healthy) class.

	Bal Acc	Accuracy	F1 Score	Recall	Precision
Value	0.92 ± 0.03	0.96 ± 0.01	0.91 ± 0.03	0.85 ± 0.06	0.98 ± 0.03

TABLE 4.2: Performance Metrics

From table 4.2 it can be seen that the model boasts strong precision and accuracy with values of 98% and 96% respectively. However, the recall is the lowest performing statistic with a value of 85%. This suggests the model has been too conservative with its positive (healthy) predictions. This is likely due to the skewed database, despite the class weight allowances made.

4.3 FILTER TYPE EVALUATION

Through training both models using data that have been filtered using both the DWT and Butterworth filter denoising methods implemented within this project, it is possible to compare the methods. This can be seen in table 4.3 where the accuracy of the models using no filter has been included as a baseline standard.

	Butterworth	DWT	No Filter
CNN	0.93 ± 0.02	0.96 ± 0.01	0.91 ± 0.03
SVM	0.79 ± 0.01	0.78 ± 0.01	0.65 ± 0.04

TABLE 4.3: Comparison of the accuracy of the CNN and SVM method when using data that have been filtered using the DWT, a 5th order Butterworth filter, or no filter at all.

Table 4.3 presents the accuracies achieved using the different preprocessing techniques for both the CNN and SVM. To ensure a fair comparison with the CNN, the accuracies for the SVM are averaged across all channels. It can be seen that the CNN performs best when the DWT is used as the preprocessing method. In contrast, the SVM achieves its highest accuracy with the Butterworth filter, although the difference between the Butterworth filter and the DWT is less pronounced for the SVM. Notably, the CNN without any preprocessing still achieves higher accuracies than any seen for the SVM method, with scores comparable to those achieved with noise filtering. This differs from the SVM, where the performance drops significantly when no noise filter is applied.

DISCUSSION

Most papers investigating ECG classification focus on the multi-classification problem on specific heart beats. In addition to this, the investigations span a variety of different methods, scoring metrics and databases. This means that a direct comparison of the results achieved here is not possible. Despite this, a more holistic comparison of similar and leading papers is a valuable exercise allowing validation and evaluation of the methods undertaken within this project. The performance of a few relevant papers is summarised within table 5.1.

Study	Model	Database	Number of Classes	Accuracy (%)
This project	SVM	PTB Diagnostic	2 (Binary)	86 ± 1
This project	1D CNN	PTB Diagnostic	2 (Binary)	96 ± 1
Farid et al [12] (2008)	SVM	MIT-BIH	5	85.98
Farid et al [12] (2008)	PSO-SVM	MIT-BIH	5	89.72
Singh et al [85] (2018)	RNN	MIT-BIH	2 (Binary)	85.4
Smigiel et al [86] (2021)	2D CNN	PTB-XL	2 (binary)	88.2
Khan et al [87] (2023)	1D CNN ResNet	MIT-BIH	5	98.63
Ahmed et al [52] (2023)	1D CNN	MIT-BIH	4	99

TABLE 5.1: Comparison of different studies with respect to model, database, number of classifications, and accuracy.

From the table it can be seen that the accuracies achieved here are broadly in line with others achieved in this field. While it can be argued that the accuracies achieved in experiments with more than binary classification are more impressive, it can be countered that these studies utilized much larger databases, providing significantly more data with which to train their models. In addition to this, the MIT-BIH database contains separate heart beats. Classifying a single beat as opposed to the health state of an ECG signal containing many beats may be a more trivial task. Table 5.1 also shows that, within this report, the 1D CNN has resulted in a better accuracy than the SVM. As mentioned in chapter 2, this is in line with conclusions drawn in other papers, however it is still worth considering why this is the case.

The SVM method relies on the accurate calculations of parameters from the ECG signals, this is often a demanding task. The quality of the ECG signals will have a substantial

impact on the accuracy of the parameters calculated from the signals. Poor quality signals with large amounts of noise may lead to poor quality parameters being calculated, and so, as the difference between the healthy and unhealthy signal is small, the parameters may not be able to distinguish between the two. This issue is exacerbated by the computational complexity of determining some of the parameters, particularly those in the non-linear domain. The time required to compute features such as the Higuchi fractal dimension, multi-fractal dimension and sample entropy meant that the desired precision could not be achieved within a reasonable time frame. This meant that these parameters would not have been as effective at distinguishing between classes. For example, when calculating the Higuchi fractal dimension a typical value of k_{\max} was stated to be roughly 5-10% of the total length of the signal, the actual value used within this analysis was a lot shorter than this so as to make the method achievable within an appropriate computational time. This meant that the feature was not calculated to a suitable precision, reducing its potential to differentiate between the two classes. This can be seen in Figure 4.3 where the Higuchi fractal dimension, along with the others named, scarcely appear. In contrast, more simpler methods, both linear and non-linear, appear more frequently. Although preliminary research indicated that non-linear methods could provide deeper insights into heart dynamics and thus have greater potential to contribute to accurate classifications, the limitations in computational power prevented these methods from being calculated with the desired precision. As a result, the assessment of which parameter domain has the most potential in ECG classification is unclear. However, it is apparent that when lacking computational power, the simple methods are the most effective.

In contrast to the SVM, the CNN does not have the problem of manual feature extraction as it automatically extracts features within the model. The time and space invariance of the CNN mean that any patterns can be identified throughout the time series without having to specifically state or pre-calculate them. Table 4.3 suggests that the CNN's ability for feature detection is minimally affected by noise, as evidenced by the smaller difference in accuracy with and without preprocessing compared to the SVM. This is likely to be one of the main contributors to the difference in accuracy, as the table shows that the CNN can perform well even with poor quality, noisy data. This means that if the data are poor the CNN may still maintain performance whereas the SVM may be greatly hindered.

In addition to the CNN boasting a better classification accuracy than the SVM, it is also the better option for potential use in real-time signal processing. The shallow depth of the CNN used here means that the model training process is a relatively fast one. In combination with the short preprocessing stage when compared to the time consuming manual parameterisation needed for the SVM method, the process in terms of the analysis from raw ECG signal to resulting diagnosis is more efficient within the CNN method. Furthermore, as the CNN is less reliant on the data pre-processing stage, as shown in table 4.3, models that achieve a desired accuracy without any pre-processing may be obtained. Therefore the noise reduction stage may be considered unnecessary allowing it to be ignored, further increasing the overall methods computational speed advantage over the SVM approach.

Whilst the dataset used within this study is a relatively small one, another benefit of the CNN over the SVM is its scalability. As every signal used within the dataset has to be parameterised within the SVM method, and it is this parameterisation that is the limiting factor in terms of the computational time of the method, the computational time of the SVM will increase linearly with the number of patients. In contrast to this, the CNN is highly flexible with regards to the size of the database. The number of convolutional layers, depth and filters used in the network can all be adapted in order to deal with larger datasets. In addition to the potential for parallelisation of convolution operations, the CNN is the more efficient method when training larger datasets.

A limitation of the CNN is its need for large quantities of data. This limitation was directly encountered within this report when it was seen that the data within a single ECG channel were not sufficient for the network to learn the patterns within the database. Instead requiring the data from every channel to be used at once. Although the change was only from orders of 10^2 to 10^3 data points, it was enough to surpass the threshold and yield meaningful results from the CNN. In instances where these smaller datasets are the only option, the SVM may be the better option. This may be particularly apt for the problem of specific disease classification. As some diseases appear very infrequently within large databases the problem of identifying and then accurately classifying them is non-trivial.

Both models are limited by the accuracy of the labels placed upon the data within the database. Due to the context of this study the only option is to take the labels as fact however this may not be the case. Diagnosing using ECGs is a time consuming task requiring a professional skill set and so is prone to mistakes through human error. According to a systematic review across 6 studies, trained physicians correctly labelled ECG signals with a median accuracy of 81% whereas trained cardiologists performed better at 88% [88]. This shows there is a possibility for mislabelled data which would have a knock-on effect to the performance of the CNN as well as the discrimination power of the SVM. However, with access to correctly labelled data it can be seen that the CNN has the potential to surpass the accuracy of the clinicians, scoring accuracies of 96% consistently within this study as shown in table 4.2. In addition to this, once a model is trained, a diagnosis can be obtained at much greater speeds than those which a clinician would be able to achieve through visual examination. At the mass level, this efficiency would be compounded. This demonstrates the potential for the 1D CNN to be used in the clinical setting for ECG classification.

CONCLUSIONS AND FURTHER WORK



In conclusion, this work has shown the benefits and limitations of both the SVM and 1D CNN methods of ECG classification. Both models were implemented on V1 - V6 ECG channels and whilst both demonstrated a certain level of proficiency, this study implies the 1D CNN to be the superior method in terms of both classification accuracy and its potential for use in real-time applications. The proposed SVM model was trained on specific discretisations of the ECG signals. These parameters were chosen due to their use in other heart rate classification studies or as biomedical markers. Both the features used and the hyperparameters governing the SVM were optimised for each channel. An automatic feature selection through a combination of a filter method and a sequential forward selector chose the optimal features for each channel whilst a grid search over the typical parameter sets decided the optimal hyperparameters. The performance of the SVM was limited by the quality of the features calculated from the signals, which is further limited by the quality of the signals themselves. During implementation of the SVM method, time, frequency and non-linear domain parameters have been described, calculated and their performance as discriminatory features evaluated. It has been seen that whilst the non-linear features were the most computationally expensive, features from all domains were needed to obtain accurate diagnoses.

The proposed 1D CNN model took the full ECG signals from every channel as its inputted data. The model incorporated a single convolutional block due to the small size of the database. Although achieving excellent performance on the PTB dataset, the model was limited by the size and skew of the dataset. Although taking account of this through class weighting and the use of a shallow model, the lack of balanced data will still have an impact on the model's generalisability. This will be particularly apparent in any further work investigating the use of the 1D CNN for the multiclassification problem, categorising the ECG signals into specific disease types.

Furthermore, the adequacy of both the Butterworth filter and the DWT as methods of denoising the ECG signal has been investigated. These results suggested that whilst the difference in the raw and denoised signals was small, the DWT is optimal for the CNN method. In contrast, the application of the Butterworth filter led to the most success for the SVM method, which had a much larger discrepancy when compared to the accuracies achieved with the raw signal. This led to the conclusion that the performance of the CNN method was a lot less reliant on the quality of the underlying data.

The 96% average accuracy of the CNN method achieved within this study compared to

the 88% average accuracy of trained cardiologists highlights the potential of this approach. When coupled with the speed at which a diagnosis can be reached from a pre-trained model, the method shows promise to be used within a clinical setting. Therefore, future work should focus on more stringent tests of the model's robustness and applicability. For example, the model should be applied to a range of ECG datasets, both large and small. Through comparisons of the accuracies achieved across datasets the generalisability of the model can be seen. Furthermore, the multiclassification problem should be addressed. This would illustrate the potential of the model to diagnose specific disease types as would be required within a clinical setting. Due to the rarity of some CVDs within ECG databases, other methods that make use of the 1D CNN could be investigated, such as the prototypical network discussed in chapter 2.

Whilst showing less promise in this project, future work may chose to address the limitations of the SVM seen here. A more intensive investigation into each of the parameters used to discretise the signals may be fruitful. Particularly in terms of the non-linear parameters, where an investigation into one single parameter may allow more computational time to be spent calculating the parameter to a higher precision, allowing easier classification, or even the development of a more efficient technique with which to calculate that parameter. Any improvements that can be made to the performance of the SVM method will be beneficial due to its dominance over the CNN in terms of performance when dealing with smaller datasets. This, as discussed, may be needed to classify rare CVDs from ECG databases.

BIBLIOGRAPHY

1. British Heart Foundation. *Cardiovascular Disease Statistics - UK Factsheet* <https://www.bhf.org.uk/-/media/files/for-professionals/research/heart-statistics/bhf-cvd-statistics-uk-factsheet.pdf>. Accessed: 2024-01-04. 2024.
2. Martis, R. J., Acharya, U. R. & Adeli, H. Current methods in electrocardiogram characterization. *Computers in biology and medicine* **48**, 133–149 (2014).
3. Tsai, C.-H. *et al.* Usefulness of heart rhythm complexity in heart failure detection and diagnosis. *Scientific Reports* **10**, 14916 (2020).
4. Perc, M. Nonlinear time series analysis of the human electrocardiogram. *European Journal of Physics* **26**, 757 (2005).
5. Chen, C. *et al.* Complexity change in cardiovascular disease. *International journal of biological sciences* **13**, 1320 (2017).
6. Goldberger, A. L. Is the normal heartbeat chaotic or homeostatic? *Physiology* **6**, 87–91 (1991).
7. Glass, L. Synchronization and rhythmic processes in physiology. *Nature* **410**, 277–284 (2001).
8. Qu, Z., Hu, G., Garfinkel, A. & Weiss, J. N. Nonlinear and stochastic dynamics in the heart. *Physics reports* **543**, 61–162 (2014).
9. Aziz, S., Ahmed, S. & Alouini, M.-S. ECG-based machine-learning algorithms for heartbeat classification. *Scientific reports* **11**, 18738 (2021).
10. Acharya, U. R. *et al.* A deep convolutional neural network model to classify heartbeats. *Computers in biology and medicine* **89**, 389–396 (2017).
11. Kohli, N. & Verma, N. K. Arrhythmia classification using SVM with selected features. *International Journal of Engineering, Science and Technology* **3**, 122–131 (2011).
12. Melgani, F. & Bazi, Y. Classification of electrocardiogram signals with support vector machines and particle swarm optimization. *IEEE transactions on information technology in biomedicine* **12**, 667–677 (2008).
13. Jun, T. J. *et al.* ECG arrhythmia classification using a 2-D convolutional neural network. *arXiv preprint arXiv:1804.06812* (2018).

14. Xu, X. & Liu, H. ECG heartbeat classification using convolutional neural networks. *IEEE access* **8**, 8614–8619 (2020).
15. Hassaballah, M., Wazery, Y. M., Ibrahim, I. E. & Farag, A. Ecg heartbeat classification using machine learning and metaheuristic optimization for smart healthcare systems. *Bioengineering* **10**, 429 (2023).
16. Al-Hiyali, M. I., Yahya, N., Faye, I. & Hussein, A. F. Identification of autism subtypes based on wavelet coherence of BOLD FMRI signals using convolutional neural network. *Sensors* **21**, 5256 (2021).
17. Liu, X., Wang, H., Li, Z. & Qin, L. Deep learning in ECG diagnosis: A review. *Knowledge-Based Systems* **227**, 107187 (2021).
18. Levick, J. R. *An introduction to cardiovascular physiology* (Butterworth-Heinemann, 2013).
19. Francis, J. ECG monitoring leads and special leads. *Indian pacing and electrophysiology journal* **16**, 92–95 (2016).
20. Wikimedia Commons. *EKG leads* Available at: https://upload.wikimedia.org/wikipedia/commons/0/0e/EKG_leads.png.
21. Ashley, E. A., Ashley, E. & Niebauer, J. Cardiology explained (2004).
22. Savalia, S. & Emamian, V. Cardiac arrhythmia classification by multi-layer perceptron and convolution neural networks. *Bioengineering* **5**, 35 (2018).
23. Gupta, V., Mittal, M. & Mittal, V. A simplistic and novel technique for ECG signal pre-processing. *IETE Journal of Research*, 1–12 (2022).
24. Xie, L., Li, Z., Zhou, Y., He, Y. & Zhu, J. Computational diagnostic techniques for electrocardiogram signal analysis. *Sensors* **20**, 6318 (2020).
25. Shekatkar, S. M., Kotriwar, Y., Harikrishnan, K. & Ambika, G. Detecting abnormality in heart dynamics from multifractal analysis of ECG signals. *Scientific reports* **7**, 15127 (2017).
26. Li, Z. & Zhang, H. *Parallel Multi-scale convolution based prototypical network for few-shot ECG beats classification* in *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)* (2022), 1–4.
27. Tantawi, M. M., Revett, K., Salem, A.-B. & Tolba, M. F. A wavelet feature extraction method for electrocardiogram (ECG)-based biometric recognition. *Signal, Image and Video Processing* **9**, 1271–1280 (2015).
28. Jagtap, S. K. & Uplane, M. *The impact of digital filtering to ECG analysis: Butterworth filter application* in *2012 International Conference on Communication, Information & Computing Technology (ICCICT)* (2012), 1–6.
29. Selesnick, I. W. & Burrus, C. S. Generalized digital Butterworth filter design. *IEEE Transactions on signal processing* **46**, 1688–1694 (1998).

30. Kumar M, A. & Chakrapani, A. Classification of ECG signal using FFT based improved Alexnet classifier. *PLOS one* **17**, e0274225 (2022).
31. Chan, H.-L., Siao, Y.-C., Chen, S.-W. & Yu, S.-F. Wavelet-based ECG compression by bit-field preserving and running length encoding. *Computer methods and programs in biomedicine* **90**, 1–8 (2008).
32. Mazomenos, E. B. *et al.* A low-complexity ECG feature extraction algorithm for mobile healthcare applications. *IEEE journal of biomedical and health informatics* **17**, 459–469 (2013).
33. Martis, R. J., Acharya, U. R. & Min, L. C. ECG beat classification using PCA, LDA, ICA and discrete wavelet transform. *Biomedical Signal Processing and Control* **8**, 437–448 (2013).
34. Safdar MF Nowak RM, P. P. Pre-Processing techniques and artificial intelligence algorithms for electrocardiogram (ECG) signals analysis: A comprehensive review. *Comput Biol Med* **170**, 107908. ISSN: 0010-4825 (2024).
35. Biran, A. & Jeremic, A. *ECG based Human Identification using Short Time Fourier Transform and Histograms of Fiducial QRS Features*. in *BIOSIGNALS* (2020), 324–329.
36. Ge, Z. *et al.* *ECG-signal classification using SVM with multi-feature* in *2019 8th International Symposium on Next Generation Electronics (ISNE)* (2019), 1–3.
37. Chashmi, A. J. & Amirani, M. C. An efficient and automatic ECG arrhythmia diagnosis system using DWT and HOS features and entropy-based feature selection procedure. *Journal of electrical bioimpedance* **10**, 47–54 (2019).
38. Singh, A. K. & Krishnan, S. ECG signal feature extraction trends in methods and applications. *BioMedical Engineering OnLine* **22**, 22 (2023).
39. De Godoy, M. F. Nonlinear analysis of heart rate variability: a comprehensive review. *Journal of Cardiology and Therapy* **3**, 528–533 (2016).
40. Castaldo, R., Melillo, P. & Pecchia, L. *Acute Mental Stress Detection via Ultra-short term HRV Analysis* in *World Congress on Medical Physics and Biomedical Engineering, June 7-12, 2015, Toronto, Canada* (Springer International Publishing, 2015), 1068–1071.
41. Coutts, L. V., Plans, D., Brown, A. W. & Collomosse, J. Deep learning with wearable based heart rate variability for prediction of mental and general health. *Journal of Biomedical Informatics* **112**, 103610 (2020).
42. Henriques, T. *et al.* Nonlinear methods most applied to heart-rate time series: a review. *Entropy* **22**, 309 (2020).
43. Castells, F., Laguna, P., Sörnmo, L., Bollmann, A. & Roig, J. M. Principal component analysis in ECG signal processing. *EURASIP Journal on Advances in Signal Processing* **2007**, 1–21 (2007).

44. Aggarwal, R. & Kumar, S. HRV based feature selection for congestive heart failure and normal sinus rhythm for meticulous presaging of heart disease using machine learning. *Measurement: Sensors* **24**, 100573 (2022).
45. Llamedo, M. & Martínez, J. P. Heartbeat classification using feature selection driven by database generalization criteria. *IEEE Transactions on Biomedical Engineering* **58**, 616–625 (2010).
46. Nasiri, J. A., Naghibzadeh, M., Yazdi, H. S. & Naghibzadeh, B. *ECG arrhythmia classification with support vector machines and genetic algorithm* in *2009 Third UKSim European symposium on computer modeling and simulation* (2009), 187–192.
47. Lyon, A., Mincholé, A., Martínez, J. P., Laguna, P. & Rodriguez, B. Computational techniques for ECG analysis and interpretation in light of their contribution to medical advances. *Journal of The Royal Society Interface* **15**, 20170821 (2018).
48. Shi, Z. *Improving k-nearest neighbors algorithm for imbalanced data classification* in *IOP Conference Series: Materials Science and Engineering* **719** (2020), 012072.
49. Pinho, A., Pombo, N., Silva, B. M., Bousson, K. & Garcia, N. Towards an accurate sleep apnea detection based on ECG signal: The quintessential of a wise feature selection. *Applied Soft Computing* **83**, 105568 (2019).
50. Moavenian, M. & Khorrami, H. A qualitative comparison of artificial neural networks and support vector machines in ECG arrhythmias classification. *Expert Systems with Applications* **37**, 3088–3093 (2010).
51. Kiranyaz, S. *et al.* 1D convolutional neural networks and applications: A survey. *Mechanical systems and signal processing* **151**, 107398 (2021).
52. Ahmed, A. A., Ali, W., Abdullah, T. A. & Malebary, S. J. Classifying cardiac arrhythmia from ECG signal using 1D CNN deep learning model. *Mathematics* **11**, 562 (2023).
53. Taherkhani, A., Cosma, G. & McGinnity, T. A Deep Convolutional Neural Network for Time Series Classification with Intermediate Targets. *SN Computer Science* **4**, 832 (2023).
54. Degirmenci, M., Ozdemir, M. A., Izci, E. & Akan, A. Arrhythmic heartbeat classification using 2d convolutional neural networks. *Irbm* **43**, 422–433 (2022).
55. Snell, J., Swersky, K. & Zemel, R. Prototypical networks for few-shot learning. *Advances in neural information processing systems* **30** (2017).
56. Erickson, B. J. & Kitamura, F. *Magician's corner: 9. Performance metrics for machine learning models* 2021.
57. Goldberger, A. L. *et al.* PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation* **101**, e215–e220 (2000).

58. Makowski, D. *et al.* NeuroKit2: A Python toolbox for neurophysiological signal processing. *Behavior research methods*, 1–8 (2021).
59. Zhao, Z. & Zhang, Y. SQI quality evaluation mechanism of single-lead ECG signal based on simple heuristic fusion and fuzzy comprehensive evaluation. *Frontiers in physiology* **9**, 727 (2018).
60. Do Vale Madeiro, J. P., Cortez, P. C., da Silva Monteiro Filho, J. M. & Rodrigues, P. R. F. in *Developments and Applications for ECG Signal Processing* 53–87 (Elsevier, 2019).
61. Mian Qaisar, S. Baseline wander and power-line interference elimination of ECG signals using efficient signal-piloted filtering. *Healthcare technology letters* **7**, 114–118 (2020).
62. Burrus, C. S., Gopinath, R. A. & Guo, H. Wavelets and wavelet transforms. *rice university, houston edition* **98** (1998).
63. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).
64. Welch, P. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics* **15**, 70–73 (1967).
65. Bartlett, M. S. Periodogram analysis and continuous spectra. *Biometrika* **37**, 1–16 (1950).
66. Solomon Jr, O. M. PSD computations using Welch's method. *NASA STI/Recon Technical Report N* **92**, 23584 (1991).
67. Shannon, C. E. A mathematical theory of communication. *The Bell system technical journal* **27**, 379–423 (1948).
68. Delgado-Bonal, A. & Marshak, A. Approximate entropy and sample entropy: A comprehensive tutorial. *Entropy* **21**, 541 (2019).
69. Lewis, M. & Short, A. Sample entropy of electrocardiographic RR and QT time-series data during rest and exercise. *Physiological measurement* **28**, 731 (2007).
70. Horie, T., Burioka, N., Amisaki, T. & Shimizu, E. Sample entropy in electrocardiogram during atrial fibrillation. *Yonago Acta Medica* **61**, 049–057 (2018).
71. Alcaraz, R., Abásolo, D., Hornero, R. & Rieta, J. J. Optimal parameters study for sample entropy-based atrial fibrillation organization analysis. *Computer methods and programs in biomedicine* **99**, 124–132 (2010).
72. Tayel, M. B. & AlSaba, E. I. Poincaré plot for heart rate variability. *International Journal of Biomedical and Biological Engineering* **9**, 708–711 (2015).
73. Guzik, P. *et al.* Correlations between the Poincaré plot and conventional heart rate variability parameters assessed during paced breathing. *The Journal of Physiological Sciences* **57**, 63–71 (2007).

74. Barabási, A.-L. & Stanley, H. E. *Fractal concepts in surface growth* (Cambridge university press, 1995).
75. Katz, M. J. Fractals and the analysis of waveforms. *Computers in biology and medicine* **18**, 145–156 (1988).
76. Cervantes-De la Torre, F., González-Trejo, J. I., Real-Ramirez, C. A. & Hoyos-Reyes, L. F. *Fractal dimension algorithms and their application to time series associated with natural phenomena* in *Journal of Physics: Conference Series* **475** (2013), 012002.
77. Higuchi, T. Approach to an irregular time series on the basis of the fractal theory. *Physica D: Nonlinear Phenomena* **31**, 277–283 (1988).
78. Wanliss, J. & Wanliss, G. E. Efficient calculation of fractal properties via the Higuchi method. *Nonlinear Dynamics* **109**, 2893–2904 (2022).
79. Kraskov, A., Stögbauer, H. & Grassberger, P. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* **69**, 066138 (2004).
80. Ross, B. C. Mutual information between discrete and continuous data sets. *PloS one* **9**, e87357 (2014).
81. Azizjon, M., Jumabek, A. & Kim, W. *1D CNN based network intrusion detection with normalization on imbalanced data* in *2020 international conference on artificial intelligence in information and communication (ICAIIC)* (2020), 218–224.
82. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **25** (2012).
83. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15**, 1929–1958 (2014).
84. Diederik, P. K. Adam: A method for stochastic optimization. (*No Title*) (2014).
85. Singh, S., Pandey, S. K., Pawar, U. & Janghel, R. R. Classification of ECG arrhythmia using recurrent neural networks. *Procedia computer science* **132**, 1290–1297 (2018).
86. Śmigiel, S., Pałczyński, K. & Ledziński, D. ECG signal classification using deep learning techniques based on the PTB-XL dataset. *Entropy* **23**, 1121 (2021).
87. Khan, F., Yu, X., Yuan, Z. & Rehman, A. U. ECG classification using 1-D convolutional deep residual neural network. *Plos one* **18**, e0284791 (2023).
88. Cook, D. A., Oh, S.-Y. & Pusic, M. V. Accuracy of physicians' electrocardiogram interpretations: a systematic review and meta-analysis. *JAMA internal medicine* **180**, 1461–1471 (2020).

APPENDIX A

The format of the header file containing the metadata pertaining to patient 001 can be seen in Figure A.1. The 12 different ECG lead files can be seen above whilst the extra relevant information about the patient is contained below. Smoking status was also going to be investigated as a potential covariate, however not enough patients and information regarding their smoking status for this to be a worthwhile inclusion.

```
s0010_re 15 1000 38400
s0010_re.dat 16 2000 16 0 -489 -8337 0 i
s0010_re.dat 16 2000 16 0 -458 -16369 0 ii
s0010_re.dat 16 2000 16 0 31 6829 0 iii
s0010_re.dat 16 2000 16 0 474 4582 0 avr
s0010_re.dat 16 2000 16 0 -260 11687 0 avl
s0010_re.dat 16 2000 16 0 -214 -16657 0 avf
s0010_re.dat 16 2000 16 0 -88 -12469 0 v1
s0010_re.dat 16 2000 16 0 -241 5636 0 v2
s0010_re.dat 16 2000 16 0 -112 -14299 0 v3
s0010_re.dat 16 2000 16 0 212 -17916 0 v4
s0010_re.dat 16 2000 16 0 393 -6668 0 v5
s0010_re.dat 16 2000 16 0 390 -17545 0 v6
s0010_re.xyz 16 2000 16 0 -3 -13009 0 vx
s0010_re.xyz 16 2000 16 0 120 7109 0 vy
s0010_re.xyz 16 2000 16 0 -18 -1992 0 vz

# age: 81
# sex: female
# ECG date: 01/10/1990

# Diagnose:
# Reason for admission: Myocardial infarction
# Acute infarction (localization): infero-latera
# Former infarction (localization): no
# Additional diagnoses: Diabetes mellitus
# Smoker: no
# Number of coronary vessels involved: 1
# Infarction date (acute): 29-Sep-90
# Previous infarction (1) date: n/a
# Previous infarction (2) date: n/a
```

FIGURE A.1: An example of the header file containing the metadata pertaining to patient 001.

APPENDIX B

```
cnn = Sequential()

for _ in range(depth):
    cnn.add(Conv1D(filters=filters, kernel_size=k, padding='same', input_shape = (60000, 1)))
    cnn.add(BatchNormalization())
    cnn.add(Activation('relu'))
    cnn.add(MaxPooling1D(pool_size=2)) # takes max out of every two
    cnn.add(Dropout(0.5))

cnn.add(Flatten())
cnn.add(Dense(1, activation='sigmoid')) # use 'sigmoid' for binary classification

# compile model
cnn.compile(optimizer=Adam(), loss='binary_crossentropy', metrics=['accuracy']) #binary cross entropy
```

FIGURE B.1: The code use to initialise and compile the 1D convolutional neural network model.

The code used to initialise and compile the 1D CNN is shown in Figure B.1. The *for* loop seen allows easy variation of the depth of the model, where in this case one single loop was used. It can also be seen that the inputted shape is set at $(60000, 1)$ in line with the length of the ECG signals used within the study. The rest of the code appears as is explained in the main text. The complete code can be viewed on GitHub as desired, at: <https://github.com/patcourts/ES98C-ECG-Project>.

APPENDIX C

The violin plots within Figure C.1 shows the different distributions of calculate parameters from each domain, for healthy and unhealthy patients. The comparison of the healthy and unhealthy distributions are as expected from literature. For example, the Shannon entropies of the signals for healthy patients are greater than those for the unhealthy, this is in line with the research discussed in chapter 2 suggesting that a heart constrained with disease loses complexity. Despite this, the challenge in classification is evident from these plots. The strong overlap in distributions between the healthy and unhealthy pateints illustrate that distinguishing between them is a non-trivial task.

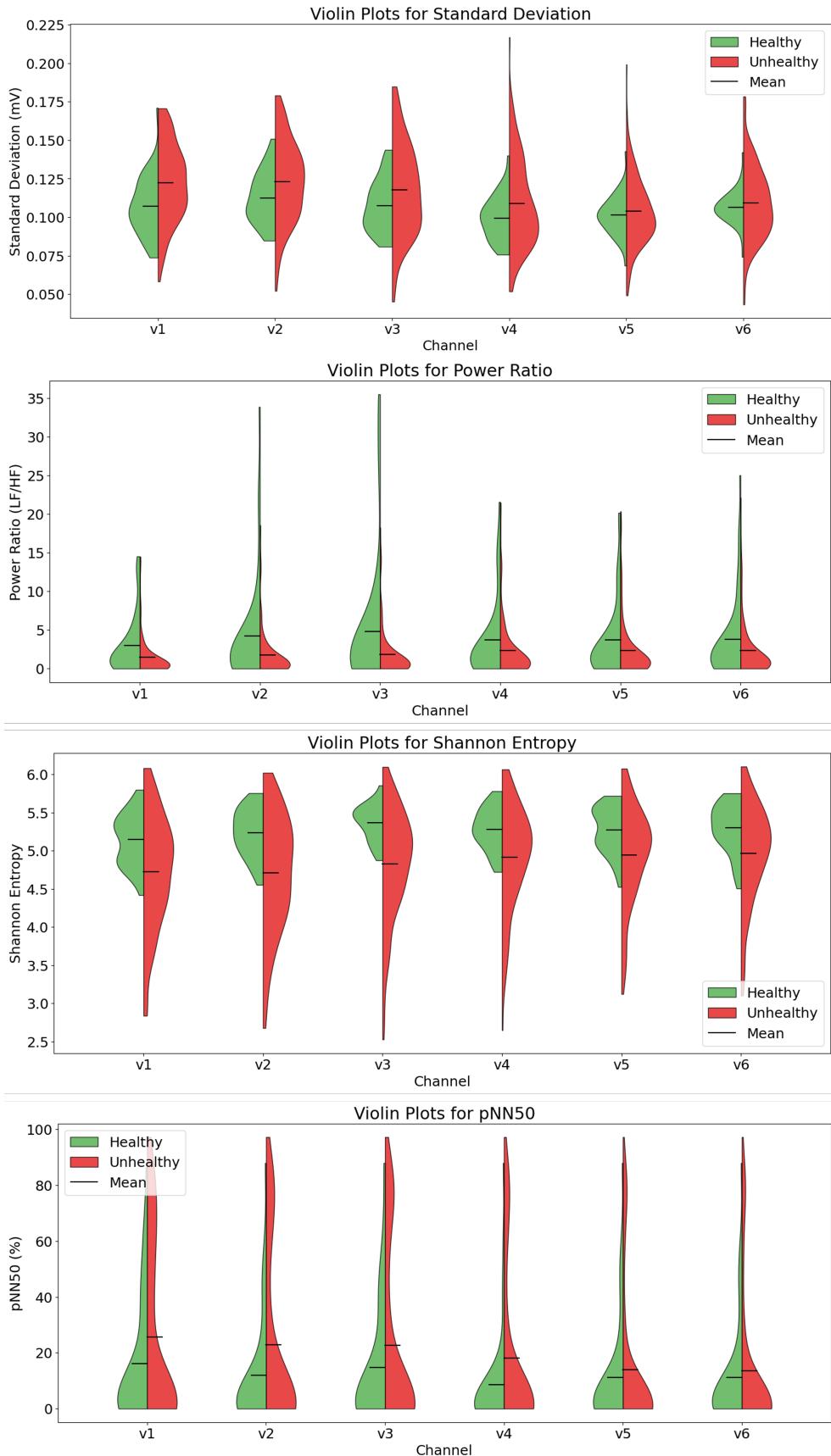


FIGURE C.1: Violin plots for different parameters calculated to discretise the ECG signals