School of Computing

FACULTY OF ENGINEERING

UNIVERSITY OF LEEDS

Deep Learning Classification of ECG Signals to detect Arrthymia

Usman Baba Usman

Submitted in accordance with the requirements for the degree of
MSc Advanced Computer Science

2021/2023

The candidate confirms that the following have been submitted*:*

Items	**Format**	**Recipient(s) and Date**
Deliverables 1	*Report*	*SSO (23/08/2022)*
Deliverable 2	*Github*	*Toni Lassila, Ali Gooya (23/08/2022)*

Type of Project: Empirical Investigation

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) ______________________

Summary

Arrhythmia is a heart rhythm disorder that affects the rate and pattern of the heartbeat. Atrial fibrillation (AF) is the most common of the arrhythmia conditions. A chaotic, and frequently fast, heartbeat is associated with AF. Furthermore, AF raises the risk of cardioembolic stroke and other heart-related issues like heart failure. As a result, it is critical to screen for AF and receive proper treatment before the condition worsens.

The goal of this project is to develop a classifier model that can distinguish and classify different the different types of arrhythmias based on dataset. This report introduces the relevant research in this application field, followed by a CRISP-DM (Cross-industry standard process for data mining) procedure to apply the experiment. To be specific, there are several divided objectives:

1. Identify an appropriate data set and make data understanding on it.
2. Data pre-processing and preparation for the model experiment. This task is challenging and will be done more than one time in order to improve coping with models.
3. Build appropriate classification models.
4. Experiments and evaluations.

Acknowledgements

"In the Name of Allah, the Most Beneficent, the Most Merciful"

First and foremost, I would like to thank Allah SWT for providing me the knowledge and understanding throughout the project.

I would like to express my deep sense of gratitude to my supervisor **Dr. Toni Lassila** for introducing me to the research area of computational biology and his dedication in helping me throughout the year to complete this dissertation and giving me valuable feedback and encouragement. As well as, providing me with the datasets and software resources.

I am also sincerely grateful to my parents ***Baba Usman*** and ***Raliya Usman*** for their continued love and support during the time spent working on this project and throughout my postgraduate degree.

I am also sincerely grateful to my friends **Umar, Khalid and Hafiz** for being my support systems throughout this period.

Table of Contents

Introduction

1.1 Background

1.1.1 The Human Heart

In order to understand the background of ECG signals and to investigate the cause of arrhythmias, the fundamental principles of the human heart structure and functions are briefly introduced in this section.

Humanities insatiable curiosity has explored every component of the human body. Particularly great attention has been concentrated of the architecture and function of the heart. Over the past decades there has been an explosion of research in both the anatomical and molecular aspects of the cardiac organ. This is due to the fact that the heart is one of the most vital organs in the body as it is imperative for the heart to function properly in order to maintain a healthy body.

One of the most interesting and most lasting fields of research is the study of The heart which is a muscle that is divided into two sides: right and left. The right side of the heart receives oxygen-poor blood from the veins and pumps it into your lungs where the blood picks up oxygen and gets rid of carbon dioxide. The left side of your heart picks up oxygen-rich blood from the lungs and pumps it through the arteries to the rest of your body. Each side consists of two chambers, the atrium and the ventricle as illustrated in Figure 1.

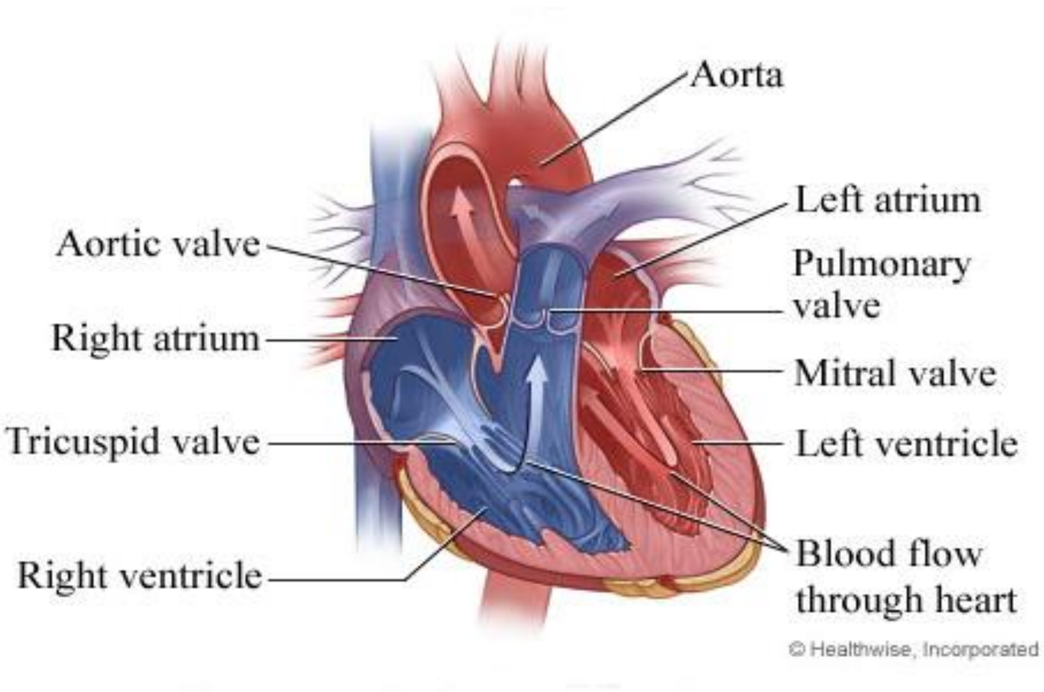


Figure 1: The Human Heart

The hearts' distribution of oxygenated blood around the human body through veins and arteries is what creates the efficiency in human productivity. Any slight interference in the performance of the heart affects the whole body in itself. The main cause of irregular heart function are called Cardiovascular diseases. These are a class of diseases which involve the heart or blood vessels. Cardiovascular diseases(CVDs) are the leading cause of illness and deaths globally, being the cause of a quarter of all deaths in the UK i.e., 160,000 deaths each year according to the British heart foundation. This paper aims to focus on a specific group of CVDs called Cardiac Arrhythmias.

1.2 Motivation

The World Health Organization states that the greatest cause of death worldwide is cardiovascular disease (CVDs). For the purpose of providing resources, tactics, and other best practises for lowering the occurrences of first and recurring cardiac disease, numerous programmes and policies have recently been put into place in a growing number of varied communities. The electrocardiogram (ECG) has emerged as the most popular bio signal for the early diagnosis of CVDs in order to meet this objective.

Cardiac arrhythmias are abnormal cardiac rhythms. Cardiac arrhythmias are classed as a type of CVD and are characterised by abnormal cardiac excitation wave start, abnormal wave propagation, or some combination of the two. Cardiac arrhythmias can present in a variety of ways, and it is still occasionally impossible to pinpoint the cause of an arrhythmia. The NHS outlines that the most effective way to detect an arrhythmia is with an electrical recording of the patients heart rhythm called an electrocardiogram(ECG) using a method called electrocardiography.

Atrial fibrillation(AFib) is defined as heart rhythm disorder originating from the upper chambers of the heart(atria) causing an abnormal and often fast rhythm due to the irregular and rapid activation of the atria. This is the most prevalent sustained cardiac rhythm disturbance i.e., Arrhythmia, which is frequently linked to a significant risk of morbidity and mortality from heart failure, stroke and thromboembolic complications.

Some of the onsets of Atrial fibrillation are barely distinguishable therefore making it very hard to detect without the use of technology, some of which include; Heart palpitations which is a sensation of fluttering or irregular or chaotic beating, light-headedness, dizziness, shortness of breath and fatigue, meaning that many people are unaware they have atrial fibrillation. Although the cause of Atrial Fibrillation are not fully understood and the disease isn't usually life threatening, the sheer fact that it is hard to detect in itself without the use of an ECG and in some people it is a serious medical condition that requires urgent treatment. Patients suffering from AFib have an increased likelihood of developing blood clots within the atrial chambers of the heart which get pumped into other organs such as the brain and in turn stop blood and oxygen from being delivered and cause significant damage. These patients are 4 to 5 times more likely to develop further conditions such as stroke.

Depending on how frequently and how long an episode lasts, atrial fibrillation can be described in a variety of ways. Paroxysmal atrial fibrillation is the term used to describe episodes that come and go and often last less than 7 days, lasting minutes or hours in many people. Long-standing persistent atrial fibrillation is defined as continuous atrial fibrillation for a year or longer. Persistent atrial fibrillation events last longer than 7 days. When atrial fibrillation is accepted to be permanent and no efforts are made to treat it, this condition is referred to as permanent atrial fibrillation.

The long lasting effect of AFib can cause the weakening of the heart muscle eventually ending in heart failure. Unfortunately with a large part of AFib cases going undiagnosed, the Stroke Association of the UK states that ‘If AFib was adequately treated, around 7000 strokes would be prevented and over 2000 lives would be saved in England alone’. With the lives of people on the line, there have been several campaigns created towards promoting awareness on AFib and Strokes with the UK having its own Stroke Awareness month which has been highly effective in raising funding towards stroke research.

Unfortunately with AFib, patients experiencing heart abnormality may plan to visit a doctor. Typically, the doctor will take an ECG, assess the patient's blood pressure, and possibly send those results to a specialist for additional evaluation. It could take weeks or even longer to wait for the results. In some instances, more specialist testing, such as donning a Holter monitor for a day or longer, may be required following the receipt of the test findings. With special cases of AFib taking longer further inconveniencing the patient, numerous techniques and tools have been created for monitoring vital signs in order to allay these worries, including wearable, portable, long-term monitoring devices that also offer real-time signal interpretation. As a result, the patient can wear a vital signs monitoring device while at home, allowing for contemporaneous analysis of the results using Machine learning methods to classify ECG signals.

1.3 Aims and Objectives

Automatic ECG classification technologies provide essential aid when it comes to early detection of arrhythmias. The importance of ECG classification is very high now due to many current medical applications. There are numerous machine learning (ML) tools available right now that can be used to analyse and categorise ECG data. The combination of heuristic hand-crafted or manufactured features with shallow feature learning architectures, however, is the fundamental drawback of these ML outcomes.

The aim of this research is to train deep learning models to classify ECG signals so as to identify different types of AFib and assist in differential diagnosis.

Project Objectives, to be described as;

- Conduct comprehensive literature review on Arrhythmias and ECG classification
- Develop pre-processing Algorithms to increase signal to noise ratio of ECGs

- Identify appropriate tools to build models using deep learning algorithms
- Evaluate the performance of selected algorithm on the provided dataset

Advanced objective 1: Analyse the performance of different deep learning models on dataset based on the accuracy and speed of classification.

Advanced objective 2: Investigate the performance of the classifiers on data taking with a different numbers of leads, i.e., 12 leads, 6 leads and 2 leads.

1.4 Project Management

Initial preparatory work has been started early in March 2022 but was paused due to exams and study commitments. Three months of full-time effort spent on this project, from May to August. Due to uncertainties about the project direction and the lack of background concerning what techniques are used at the start time of the project, some items were scheduled unrealistically. The initial plan was also unrealistic in the length of time assigned to some tasks; therefore, the timescale was adjusted appropriately.

	April				May				June				July				August			
Tasks	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identify topics	■	■																		
Aims and requirements			■	■																
Project outline					■															
Background Research						■	■													
Project scoping and planning								■												
Data Preparation									■	■	■									
Modelling												■	■							
Progress Review Meeting														■						
Evaluation of Results															■	■	■			
Final Report Complete																		■	■	■

Table 1: Project Gantt Chart

1.5 Risk Management

Identifier	Description of Risk	Likelihood	Severity	Mitigation
1	Computer Malfunctioning, resulting in losing work	Medium	High	Work is done in a cloud application such as google colab.
2	The physio net dataset is too difficult to implement	Medium	Medium	Using Internet resources to learn as well as get advice from supervisor.
3	Poor Model design resulting in low accuracy	Medium	Medium	Discussing with supervisor and getting help on model structure.
4	Lack of Time management	Medium	High	Use of project management tools like trello and Gantt charts to design an adequate project plan and ensure it is followed.
5	Minor/Major illness	High	High	Provide adequate time in project plan to account for minor inconveniences and apply for extension if needed.
6	Incomplete dissertation	Low	High	Enforce project plan and give provide proper back up options in case of failure.

Table 2: Risk Assessment

1.6 Report Outline

The report of this project studies data pre-processing, feature extraction and data classification based on ECG signals. The paper is divided into six chapters. The first chapter is the introduction which mainly introduces the research background and a breakdown of the projects motivation was detailed to aid the reader understand why this paper addresses deep learning based models to detect and classify Arrhythmias. Furthermore, the aims and objectives were established for report clarification. Finally, a project management section was written to break down the milestones needed to be accomplished for this project.

In the second chapter of the report, a literature review on similar past papers were appraised. Firstly, background research on the working principle of the ECG signal and how they are captured and then introduces machine learning methods most commonly used for the data classification algorithm. This chapter is the basic content of this paper and provides data support for research work.

The third chapter discusses the experimental setup. Firstly, the data set is introduced in detail, including data set collection, sensor location, data format and data analysis tools. Secondly, data processing and feature extraction are carried out.

Chapter four describes the experimental setup. Firstly the data set is introduced in detail including data source, data format and data analysis. In this section, a brief overview of performance metrics to be used for evaluating the results and the significance of the is to be discussed.

Chapter five analyses and optimizes the results of chapter four. Firstly, the results will be judged based on the performance metrics discussed in the earlier chapter.

Finally, chapter six summarizes the research content and results of this project, and then summarizes the deficiencies and possible future research directions of this paper.

Chapter 2

Background Research

In this chapter, the main concepts and technologies regarding arrhythmias and ECG classification are reviewed and comprehensively discussed. Thereafter, the review considers existing ECG classification models and pre-processing algorithms used, and the major challenges posed to successfully apply it will be discussed in detail.

2.1 Cardiovascular Disease

Heart diseases or cardiovascular diseases (CVD) are a class of diseases that involve the heart and blood vessels. Cardiovascular disease has long been a leading cause of death in the medical field. The number of people suffering from cardiovascular diseases is increasing year by year as people's living standards improve and life pressures rise. Acute cardiovascular diseases and chronic cardiovascular diseases are the two main types of cardiovascular diseases. Heart disease can be caused by a variety of factors. Changes in lifestyle, such as smoking, eating habits, physical activity, obesity, and diabetes, as well as biochemical factors such as blood pressure or glycemia. As a result, it is mandatory to record critical heart behaviour based on each type of heart disease and provide a system that assists doctors in making correct and efficient decisions while diagnosing. Because traditional wet-lab experiments for identifying cardiovascular diseases such as taking measurements of blood sugar, blood pressure and cholesterol are inefficient and time-consuming, the computer method plays an important role in cardiovascular disease treatment. A lot of research has been done on the use of machine learning in CVD detection; Warner et al studied the role of mathematical programs for the diagnosis of congenital heart disease, this paper which proposed a mathematical model based on Baye's theory of probability for the clinical diagnosis of congenital cardiac disease. Their method allows for enhanced congenital heart disease diagnosis with accuracy comparable to that of a doctor and improved with modifications to the symptoms and physical indicators(Warner, Toronto and Veasy, 1964). Gavhane et al. also tried to predict heart disease using a machine learning algorithm. They began by gathering the user's age, gender, blood pressure, heart rate, and other data, and then used a multi-layer net to build a machine learning model to predict heart disease. The model performed well in 30 trials and can be used to predict heart disease (Gavhane *et al.*, 2018). The use of machine learning for cardiovascular disease detection is a very complex task, with the subsequent amount of data needed from a patient proving rather exhaustive. This project aims to detect a specific type of CVD called cardiac arrhythmias using as little data as possible.

2.2 Arrhythmias

Cardiac arrhythmias are common in people of all ages and can occur in the presence of underlying heart disease as well as in structurally normal hearts. Cardiac arrhythmias arise as a result of abnormal impulse initiation or conduction, it is a condition that alters the rhythm and regularity of the heartbeat. Atrial fibrillation (AFib), one of the many arrhythmia disorders, is the most common. An erratic, frequently rapid heartbeat is linked to AFib. Additionally, AF raises the risk of heart-related issues such heart failure and cardioembolic stroke(Yuki, Hagiwara *et al.*, 2018) . Therefore, it's important to check for AFib and get the right care before the issue worsens. AFib causes irregular and frequently very rapid or sluggish heartbeats. The electrical impulses in the atria, the heart's upper chambers, are chaotic and out of rhythm as shown in the figure below with the ventricles in AFib (lower chambers of the heart).

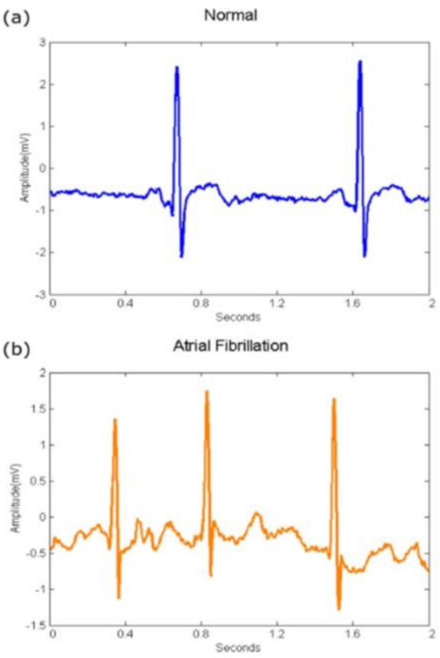


Figure 2: Atrial Fibrillation Rhythm vs Normal Heart Rhythm (Yuki, Hagiwara *et al.*, 2018)

AFib is classified as paroxysmal, persistent, or permanent (longstanding persistent AF) (*Atrial fibrillation - Symptoms and causes*, no date).

- **Paroxysmal Atrial Fibrillation**: Paroxysmal AFib refers to AFib episodes that occur infrequently and usually stop on their own. Episodes can last a few seconds, hours, or even days before stopping and returning to the heart's normal sinus rhythm. This is the hardest to detect as you need the patient to wear a portable device to monitor it when it occurs.

- **Persistent Atrial Fibrillation**: A persistent AFib episode is one that lasts more than 7 days. It does not go away unless treated. Medication or electric shock treatment can help restore normal rhythm.
- **Permanent Atrial Fibrillation:** This could go on for a long time. Typically, the decision has been made not to restore sinus rhythm with medication or electric shock therapy.

The gold standard for the diagnosis of AFib as of right now is electrocardiogram (ECG) , other alternatives which healthcare professionals typically use are clinical examination, Holter monitors, an event monitor, stress test and an echocardiogram.

2.3 Electrocardiogram

The ECG was the first and most popular bioelectric signal to be analysed by a computer, and it later became the most widely used cardiac diagnostic test. (Rivera-Ruiz, Cajavilca and Varon, 2008). The Electrocardiogram(ECG) was first introduced in the year 1877 by a man called Augustus D. Waller. In its basic definition, ECG is the electrical representation of the contractile activity of the heart, and can be recorded fairly easily by using surface electrodes on the limbs or chest of the patient. The ECG is a type of medical diagnostic equipment used to measure heart rate, specifically the electrical current produced by the heart as it beats. ECGs are classified according to the number of leads they have, that is, the number of electrical signals that are monitored at the time. The standard ECG is A 12-leads, which uses six electrodes on the chest and one on each limb for a total of 10 electrodes on the chest. The configuration of the electrodes produces 12 leads by the way signals are transmitted between them.

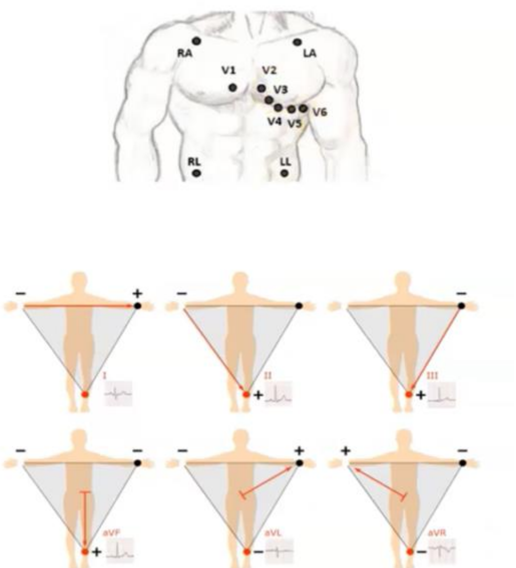


Figure 2.3: Location of 12 Leads on the body surface

The rhythm of the heart in terms of beats per minute (bpm) can be easily calculated by counting the R peaks of the ECG wave during one minute of recording. More crucially, cardiovascular disorders and abnormalities such cardiac arrhythmias can modify the rhythm and morphology of the ECG waveform and this paper's main focus is on the automatic detection and categorization of these conditions. The ECG waveform contains vital information about the health of the heart.
A typical ECG waveform consists of the P wave, QRS complex, and T wave as shown in Figure 2.4 below. In an AFib ECG signal however, the P wave is absent and is replaced by numerous and inconsistent fibrillatory waves. Nonetheless, the QRS complex can be seen in the waveform.

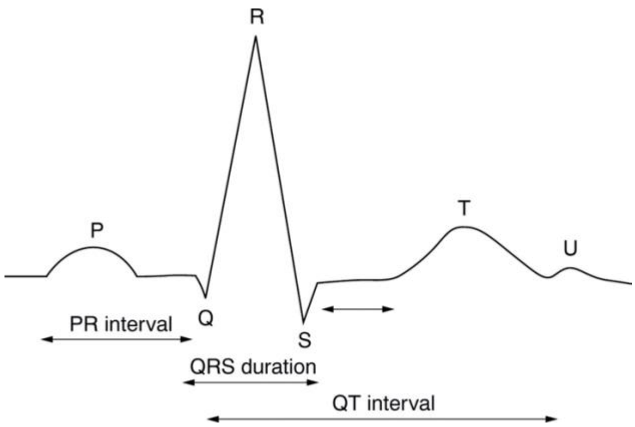


Figure 2.3: Components of an ECG signal

ECG recordings are normally characterised by a lot of interference and noise by power line disturbance, instrumentation noise, electrode contact noise and electromyography(EMG) noise.(Chang, Ko and Chang, 2010). As a result, the ECG signals become complex and difficult to interpret. Additionally, it could be difficult to interpret the signals based on the observations made during constrained time frames due to the ECG signal's time-varying nature making it difficult to spot changes. Even among cardiologists, manual evaluation of ECG signals may be arbitrary and subject to variation.
There has been extensive research on many different methodologies to develop a system for the automated classification of AF and ECG signals. For instance Martis et al. (Martis *et al.*, 2013), utilised an independent component analysis (ICA) technique to retain only the most significant features after decomposing the ECG signals with DWT up to eight levels to denoise the ECG segments. The naive Bayes (NB) classifier was then fed these reduced characteristics, and it produced results with a high accuracy of 99.33%. Similarly, Wavelet packet transform (WPT) was used by Daqrouq et al. to breakdown the ECG signals and extract features with average framing percentage energy. After being included to the probabilistic neural network (PNN) classifier, these signals produced a diagnostic accuracy of 97.72% (Daqrouq *et al.*, 2014).

Further work by Kora et al. offered a quick and effective method for classifying the two classes of ECG signals by using the CS-SCHT technique to extract key features. To choose the best set of features to feed the LMNN classifier, they used a mix of two optimizer techniques, the firefly and particle swarm optimization (FFPSO). The diagnostic performance of the suggested approach was over 99%.

What the papers introduced common is the need to use feature extraction techniques so as to prepare the ECG signals for classification, this project aims to build a classifier that uses a type of machine learning technique called deep learning to increase the overall robustness of the current system, as with deep learning, R-peak detection and removal of noises and artifacts from ECG signals are not required thereby simplifying the overall process.

2.4 Machine Learning

Machine learning is the subfield of computer science that, according to Arthur Samuel in 1959, gives "computers the ability to learn without being explicitly programmed”. Tom Mitchell defines machine learning as ‘A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance a tasks in T, as measured by P, improves with experience E.” Zhi-Hua Zhou defines machine learning as the technique that improves system performance by learning from experience using computational methods.(Zhou, 2021). Data is used to represent experience in computer systems, and the basic goal of machine learning is to create learning algorithms that create models from data. We create a model that can make predictions by feeding it experience data.

Machine learning, put simply, is the process of using data to train a model, which is then used to make predictions. Evolved from research based on computational learning theory and pattern recognition in artificial intelligence(William L Hosch, 2021). Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Examples of applications for machine learning include email filtering, detection of network intruders or malicious insiders attempting a data breach, optical character recognition (OCR),learning to rank, and computer vision. Machine learning is used in a variety of computing tasks where designing and programming explicit algorithms with good performance is challenging or impractical.

The two most widely adopted machine learning methods are supervised learning and unsupervised learning. Sometimes semi-supervised and reinforcement learning techniques are used.

2.5 Supervised Learning

In supervised learning, a mapping between a set of input variables X and an output variable Y is learned, and this mapping is then used to forecast the results for hypothetical data.
The ability to analyse fresh data with more accuracy depends on the number of samples that are made available to the supervised learning algorithm. The fundamental difficulty with supervised learning is that it takes a lot of time and labour to create huge data from labelled samples.

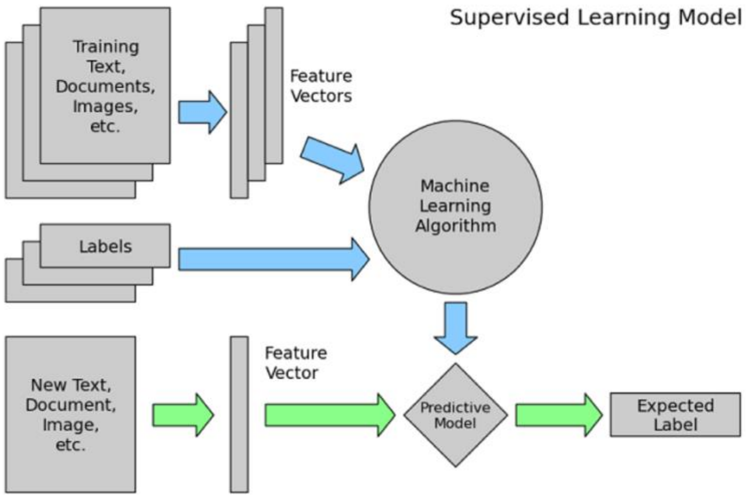


Figure 2.5: Supervised Learning (Alzubi, Nayyar and Kumar, 2018)

The figure above shows the structure of a supervised machine learning process. To train a Predictive Model, you must first prepare training data, which can be text, image, or audio, and then extract the required features to form feature vectors, and then send these feature vectors, along with the corresponding markers/targets, into the learning algorithm. The feature vector for testing is then obtained by applying the same feature extraction method to the new test data. Finally, the prediction model is used to predict the feature vectors to be tested and the results are obtained.
There are two main applications for supervised machine learning: classification problems and regression problems.

2.5.1 Classification

Classification is the most common supervised model application. This refers to taking an input value and mapping it to a discrete value. This typically occurs when our output consists of multiple classes or categories. This could range from trying to predict what objects are present in an image or whether it is going to rain or not.

2.5.2 Regression

Regression is related to continuous data (value functions) which is characterized by an annotated data set with numerical target variables. In Regression, the predicted output values are real numbers. It deals with problems such as predicting the price of a house or the trend in the stock price at a given time, etc.

2.6 Unsupervised Learning

Unsupervised Learning is a type of machine learning in which the algorithms are provided with data that does not contain any labels or explicit instructions on what to do with it. The goal is for the learning algorithm to find structure in the input data on its own.

Unsupervised learning, to put it simply, is a type of self-learning in which the algorithm may discover previously unknown patterns in unlabelled datasets and provide the necessary output without any interference.

Finding these underlying patterns aids in data clustering, association, and anomaly and error identification.

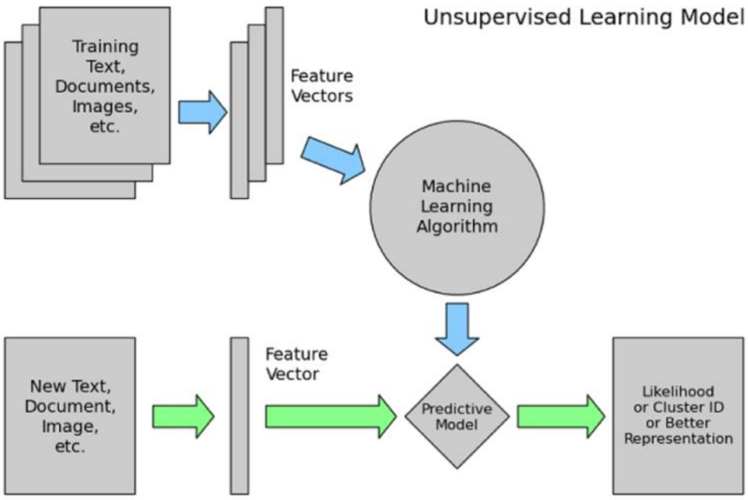


Figure 2.6: Unsupervised Learning Model

Unsupervised models can perform more complex tasks than supervised models but they are often unpredictable. A few ways in which unsupervised models handle complex problems are as follows;

2.6.1 Clustering

Clustering is the type of unsupervised learning technique where the model tries to differentiate data by finding hidden patterns based on their similarities or differences. These patterns can relate to the shape, size, or colour and are used to group data items or create clusters. For example, given a similarity criterion, find out which ones are more similar to the other.

2.6.2 Association

Association is a type of unsupervised learning where the model tries to find a relationship between one data item and another. We are now able to build dependencies based on this relationship and map them in a way that benefits us. Using association we begin to understand customer habits so as to know what to market/advertise to them, for example, there exists a product A and a product B, after analysing shopping carts of multiple customers we begin to realise that people who have bought product A are very likely to buy product B, then we can easily start recommending product B to them when they have product A in their cart.

The association rule is used to find the probability of co-occurrence of items in a collection. These techniques are often utilized in customer behaviour analysis in e-commerce websites such as Amazon, Ali Baba and OTT platforms like Netflix and Disney+.

2.6.3 Dimensionality reduction

The method attempts to minimise the dimensions of the data, as its name suggests. For feature extraction, it is employed. A key component of machine learning algorithms is identifying the key features in the dataset. By removing pointless features, this reduces the amount of random variables in the dataset.

2.7 Semi-Supervised Learning

These algorithms offer a method for combining the strengths of supervised and unsupervised learning. In the first two categories of output, labels are either given for every observation or none are given at all. There may be instances where certain observations are given labels, but the majority of observations are left unlabelled because labelling is expensive and requires specialised human knowledge. Semi-supervised algorithms are the most appropriate for generating models in these circumstances. Classification, regression, and prediction issues can all be solved using semi-supervised learning.

2.8 Deep Learning

The majority of modern civilization is powered by machine learning, including social network content filtering, e-commerce website suggestions, and a growing number of consumer goods like cameras and smartphones. Machine-learning algorithms are used to choose relevant search results, recognise objects in photos, convert speech to text, match news articles, posts, or products with users' interests, and more. These applications are increasingly using a group of methods known as deep learning. Although traditional machine learning algorithms are sufficient for dealing with the relatively large number of data, it is not enough fast and effective in dealing with modern days big data and hard to improve the result.

Deep learning is a set of algorithms in machine learning which attempts to learn important features from raw data automatically. Like traditional neural networks, the deep neural networks (DNN) also consist of artificial neurons arranged in the form of input, hidden, and output layers (Hinton *et al.*, 2012). DNNs typically include more hidden layers than ordinary neural networks do, frequently more than one. Deep neural networks' hierarchical structure enables them to learn features at several levels, each of which corresponds to a different level of abstraction.

At the first layers, fundamental concepts are learned, which are subsequently aggregated at the deeper layers to create higher level notions. Deep learning has a wide variety of applications and can be applied directly to different data types such as images, signals, audio and text.(Ahmad, Farman and Jan, 2019)

The potential for deep learning algorithms has been dramatically altered in the last ten years by the emergence of supercomputers, GPUs, and big data(Raina, Madhavan and Ng, 2009). These developments make it possible for deep learning techniques to efficiently use both labelled and unlabelled data, complicated compositional nonlinear functions, and automatically learn distributed and hierarchical features.(Ahmad, Farman and Jan, 2019).

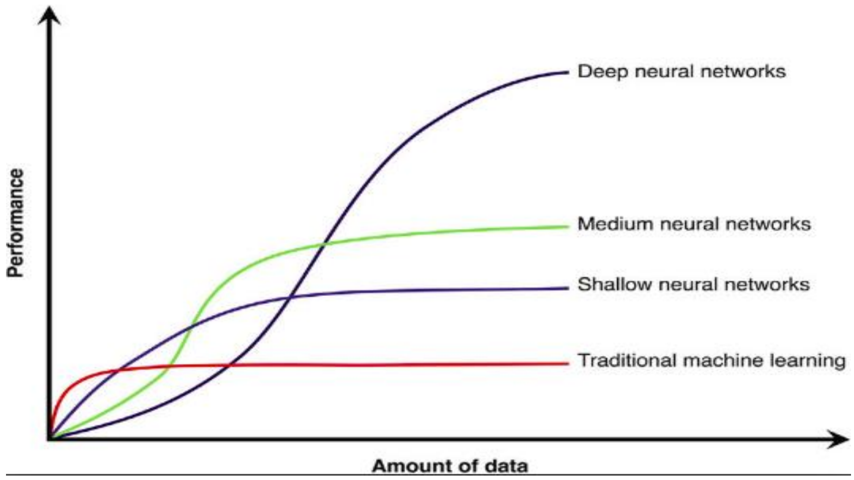


Fig 2.8: Deep Learning vs Machine Learning

Deep learning has demonstrated its advantage over conventional machine learning time and time again as computing power has increased. Deep learning is superior not only in terms of processing speed but also in terms of input variety, data diversity, data sizes, and other factors. Fig. 2.7 provides an example comparison.

The image above contains four curves. The bottom red curve, for example, represents the performance of traditional machine learning algorithms such as SVM, logistic regression, decision tree, and so on.

When the amount of data is relatively small, the traditional learning model performs better. However, when the amount of data is large, the performance trend is essentially stable; The smaller neural network model is represented by the yellow curve above the red curve (Small NN). When the amount of data is large, it outperforms traditional machine learning algorithms. When the amount of data is larger, the blue curve represents a medium-sized neural network model (Media NN) that outperforms Small NN. Finally, the top green curve represents a larger neural network (Large NN), the deep learning model, which, as shown in the figure, performs best when the amount of data is large, and basically maintains a relatively fast rising trend.

Additionally, feature engineering is necessary for conventional machine learning algorithms to reduce the dimension of the data. Because of its multi-layer structure, deep learning can feed data directly into the network and perform internal feature learning and data normalisation, which solves the problems associated with feature engineering and dimensionality reduction.

A well-trained model can be easily transferred to new data in related applications because to the separation of the model structure and data inputs in deep learning.

2.9 Multi-Layer Perceptron

As the need for real time environmental modelling increases, so does the complexity of the approach towards understanding and solving the problems. A comprehensive numerical model is maybe the most ideal approach if there are sufficient data and computational resources available, as well as if a solid theoretical understanding of the issue is available. However, in general, as a problem's complexity rises, theoretical comprehension declines (due to poorly defined interactions between systems), necessitating the use of statistical methodologies. Recently, it has been demonstrated that neural networks, specifically the multilayer perceptron, are efficient substitutes for more established statistical methods.(Gardner and Dorling, 1998). It can be trained to approximate any smooth, measurable function(Hornik, Stinchcombe and White, 1989). When given fresh, untested data, it may be trained to generalise with accuracy and represent extremely non-linear functions. (Gardner and Dorling, 1998).

The multilayer perceptron is a model that represents a nonlinear mapping between an input vector and an output vector and is made up of a system of simple interconnected neurons, or nodes, as shown in Fig. 2.8. Weights and output signals that are a function of the sum of the node's inputs modified by a simple nonlinear transfer, or activation, function connect the nodes. The simplest MLP has 3 layers and the number of neurons in the input layer would typically show the dimensions of the input data, and the output layer which signifies the amount of classes possible to predict normally has just one neuron. The middle of an MLP consists of a number hidden layers as well as a number of neutrons in each layer. Typically the more complex the task, the more complex the middle of an MLP is, meaning more hidden layers and more neutrons to be able to accurately perform the task. The downside to having a more complex MLP is the amount of time it would take to train the network itself.

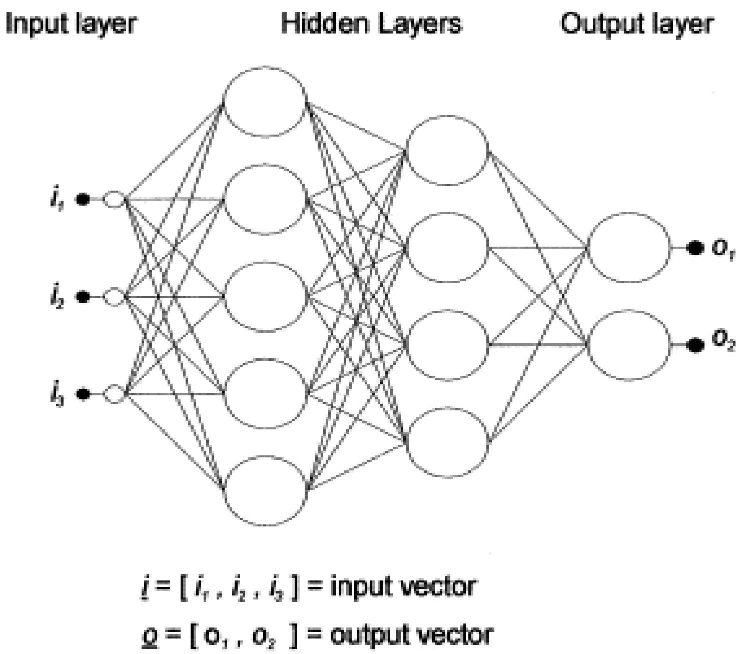


Fig 2.9: Multi-layer Perceptron (Gardner and Dorling, 1998)

2.10 Recurrent Neural Network(RNN)

A linear programming problem attempts to optimise a linear objective function subject to a set of linear functional constraints and non-negativity constraints. Large-scale linear programming problems must be solved in real time for many large-scale systems, including human and equipment manoeuvres in military operations and machine vision in robotic operations. Due to the limitations of sequential processing, conventional sequential algorithms and basic neural networks for example; MLP have proven not to be efficient due to the limitation of signal processing.(Wang, 1993).

Due to the nature of this project, the input data provided is in form of a time series. For this reason, we used Recurrent Neural Network (RNN) to overcome the constraints of basic MLP. Fig 2.9 shows the structure of a basic RNN:

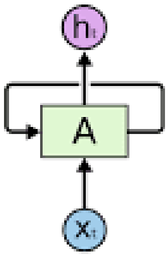


Figure 2.10: Simple RNN

The RNN employs a feed-forward multi-layer neural network with three hidden layers positioned between the input and output layers, each of which has n units, to represent the n features in the training data. The RNN produces an output for each moment's input combined with the current mode's state. In the figure above, At each time step t, the recurrent layer receives input x(t) as well as the output from the previous time step. RNN can be viewed as the result of replicating the same neural network structure in time series like the concept graph shows in Fig 2.6.

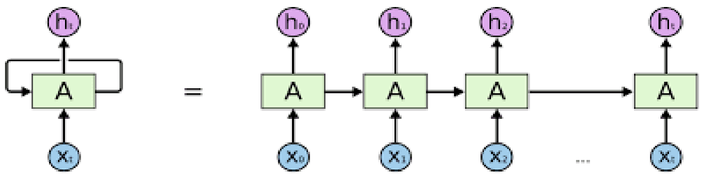


Fig 2.10: RNN network

RNNs are trained by unrolling them through time as in Fig 2.10 and then using regular backpropagation. This process is called backpropagation through time(BPTT). The output of the whole recurrent layer can be computed as follows

$$y_{t} = \Phi(W_x^T \cdot X_{(t)} + W_y^T \cdot Y_{t-1} + b)$$

2.11 Long Short Term Memory

The typical feature of the RNN architecture is a cyclic connection, which enables the RNN to possess the capacity to update the current state based on past states and current input data. However, although the simple RNN has had incredible success on some problems, when the gap between the relevant input data is large, the simple RNN is unable to connect the relevant information to where it is needed. These problems are called 'long-term dependencies'.

In order to handle these 'long-term dependencies', Hochreiter and Schmidhuber (Hochreiter and Schmidhuber, 1997) proposed the long short-short term memory(LSTM). The LSTM built upon the initial remembering capacity of the basic RNN cell by introducing a 'gate' into the cell. The LSTM recurrent neural network is made up of a memory cell module that can learn data features in the time domain. Because of its superior processing performance for time series data, it has found widespread application. The initial LSTM proposed by Hochreiter and Schmidhuber consisted of two gates as shown in figure 2.11 which only consisted of an input gate and an output gate:

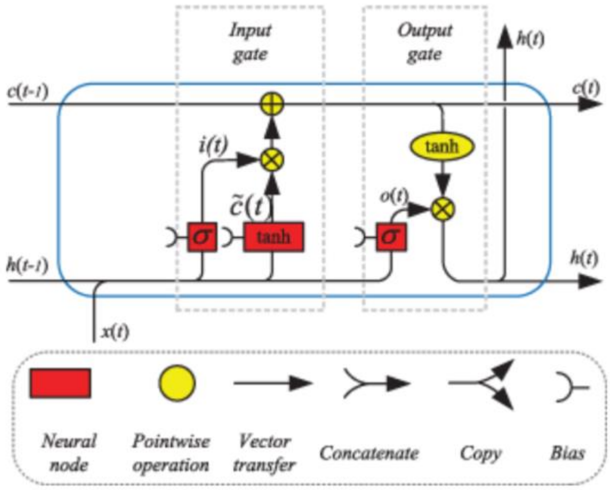


Figure 2.11: LSTM without Forget Gate

The LSTM cell with only an input and output gate as shown above can be mathematically expressed as:

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i)$$
$$\check{c}_t = \tanh(W_{\check{c}h}h_{t-1} + W_{\check{c}x}x_t + b_{\check{c}})$$
$$c_t = c_{t-1} + i_t \cdot \check{c}_t$$
$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o)$$
$$h_t = o_t \cdot \tanh(c_t)$$

where;

c_t *denotes the cell state of the LSTM*

$W_i, W_{\check{c}}, W_o$ *are the weights*

When the cell state is updated, the input gate determines what new information can be stored in the cell state, and the output gate determines what information can be output based on the cell state.

The current LSTM recurrent neural network memory module introduced by Gers et al. contains three multiplication units: an input gate, a forget gate, and an output gate. These gates, in turn, control information input, update, and output, allowing the network to perform a specific memory function. In the meantime, the network has more learning parameters as a result of these gates. Figure 2.12 below shows the structure of an LSTM cell:

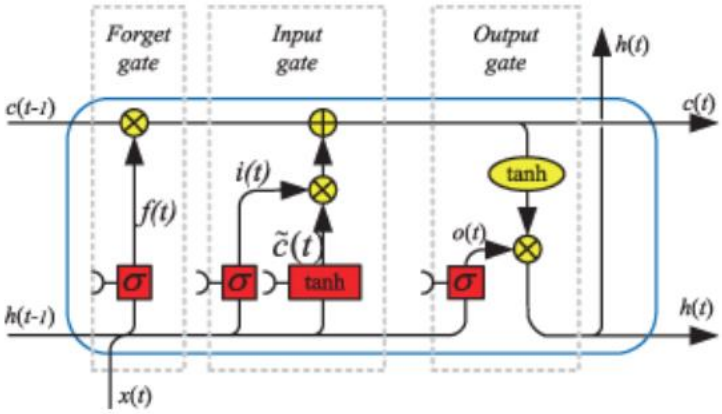


Figure 2.12: LSTM with Forget Gate

The forget gate has the function of deciding what information will be thrown away from the current state of the cell and can be mathematically expressed as follows:

$$f_t = \sigma\left(W_{fh}h_{t-1} + W_{fx}x_t + b_f\right)$$

When the forget gate, ft, is set to 1, it keeps this information; a value of 0 means it discards all information.

2.12 Gated Recurrent Units

The LSTM cell outperforms the standard recurrent cell in terms of learning capacity. However, the additional parameters add to the computational load. As a result, the gated recurrent unit (GRU) was developed by Cho in 2014 (Cho *et al.*, 2014). The GRU is a special case of the LSTM which reduces the amount of gates from three to two in order to reduce the number of parameters. The GRU cell integrates the forget gate and input gate of the LSTM cell as an update gate. The GRU cell only has two gates: an update gate and a reset gate. The structure of the GRU is show in figure 2.9 below:

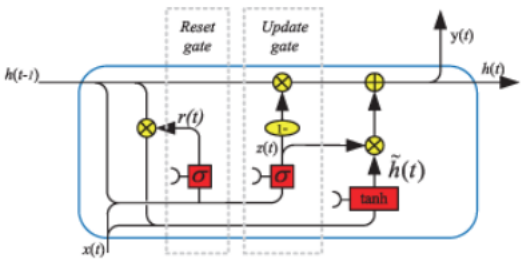


Figure 2.9: GRU cell

The GRU cell shown in figure 2.9 is mathematically expressed as follows:

$$r_t = \sigma(W_{rh}h_{t-1} + W_{rx}x_t + b_r)$$
$$Z_t = \sigma(W_{zh}h_{t-1} + W_{zx}x_t + b_z)$$
$$h_t = \tanh(W_h\,(r_t \cdot h_{t-1}) + W_h x_t + b_z)$$
$$h_t = (1 - Z_t) \cdot h_t + Z_t \cdot h_t$$

Chapter 3

Experimental Setup

In this chapter, all the necessary steps to set up the experiments and the tools to be used are presented. this chapter focuses on the data understanding and data preparation phases to relate the selected dataset. Along with a justification for choosing the dataset, major problems of the dataset are addressed. Further, the software tools for analysis are explained.

3.1 Dataset Selection

The data used for this project is acquired from the Physio net which is an extensive archive of well-characterised digital recordings of physiologic signals, time series and related data for use by the biomedical research community. The data contains ECG signal files, header files and attribute files for each patient, Which is captured using 12 lead Holter or 3-lead wearable ECG monitoring devices. The signals are extracted from lead I and II of the long-term dynamic ECGs, each sampled at 200 Hz. The data was acquired from 49 AFib Patients with a subclass of 23 Paroxysmal AFib Patients and 56 non AFib patients(usually including other abnormal and normal rhythms).

3.2 Data Structures

There is in total, 1436 records from 91 patients in this dataset --- each signal file is stored in a ‘.dat. file which is the standard WFDB format for ECG signal files. The total dataset file size is 1.0 GB. There are three types of data files, for each recording there exists a signal file, a header file and an attributes file. The signal file contains a 2d array with the first value of the array corresponding to data from Lead I of the Holter device and the second corresponding to II from the device.
The name of each file is in the format the header file which describes the length of ECG signal in the signal file as well the amount of leads that were used to record it and the frequency in which the data was recorded. The header file also contains the age, gender, signal names, the units of each signal as well as the diagnosis of the signal in the comments section as seen in Figure 3.1 below:

```
{'record_name': 'data_0_10',
 'n_sig': 2,
 'fs': 200,
 'counter_freq': None,
 'base_counter': None,
 'sig_len': 156768,
 'base_time': None,
 'base_date': None,
 'comments': ['non atrial fibrillation'],
 'sig_name': ['I', 'II'],
 'p_signal': None,
 'd_signal': None,
 'e_p_signal': None,
 'e_d_signal': None,
 'file_name': ['data_0_10.dat', 'data_0_10.dat'],
 'fmt': ['16', '16'],
 'samps_per_frame': [1, 1],
 'skew': [None, None],
 'byte_offset': [None, None],
 'adc_gain': [36960.66594624795, 23321.574588417934],
 'baseline': [-2430, -18661],
 'units': ['mV', 'mV'],
 'adc_res': [16, 16],
 'adc_zero': [0, 0],
 'init_value': [-4229, -25017],
 'checksum': [41165, 7554],
 'block_size': [0, 0]}
```


Figure 3.1: Header file sample

3.3 Tools

3.3.1 Python

Python is an object-oriented programming language with a simple structure and easy to learn. One of its biggest advantages is its wide range of libraries and good compatibility. Therefore, it is often used as a tool for data processing, which can handle data from KB to T, with high development efficiency and maintainability. Python also has one of the largest package repositories, allowing it to be used for almost all areas of computer science. Python satisfies all the technical specifications for this project, including deep learning, computer vision, and data visualisation. Therefore, Python can do all the work in one type of code and at no cost, as opposed to utilising one tool to process the data, another to analyse it, and finally a third tool to visualise it.. Python also has a library called WFDB is which is needed to decode and load in the ECG signal files and merge them with the header files to create a record. This will prove very useful in the data processing part of this task.

3.3.2 Google Colab

Google colab is a software as a service(SOAS) platform which allows anybody to write and execute arbitrary python code through the browser, and is especially well suited for machine learning , data analysis and education. It also provides access to computing resources like GPUs which we require for computing intensive tasks like this as it reduces the amount of time needed to train and validate our model.

Chapter 4

Model Design

The Model this project aims to design should achieve the objective to classify the different types of arrythmia as well as the different types of Atrial Fibrillation using deep learning. The model is constructed in accordance with the criteria stated in the previous literature study in Chapter 2 that it should be able to handle the temporal data.

4.1 Implementation

4.1.1 Deep Learning Framework

There is wide variety of deep learning frameworks as of recent, most of these frameworks are open source and support python so we have a wide array of options to use for this project. This project uses Keras to implement our experimental model. This is because of Keras' ability to support data parallelism and handle massive volumes of data while accelerating the training time for user coupled with the fact that it is designed to be straightforward and easy to use which is ideal for this project. Keras also integrates and builds upon other deep learning frameworks like Theano and TensorFlow an also python packages like NumPy.

4.1.2 LSTM

As described in the literature review, Recurrent Neural Networks with Long Short-Term Memory cells were created to understand the temporal relationships in data, as discussed in Chapter 2. So, the simplest and most basic technique is to build an LSTM model using the Keras API. The order in which a model is built using the Keras sequential API is shown in Figure 4.1:

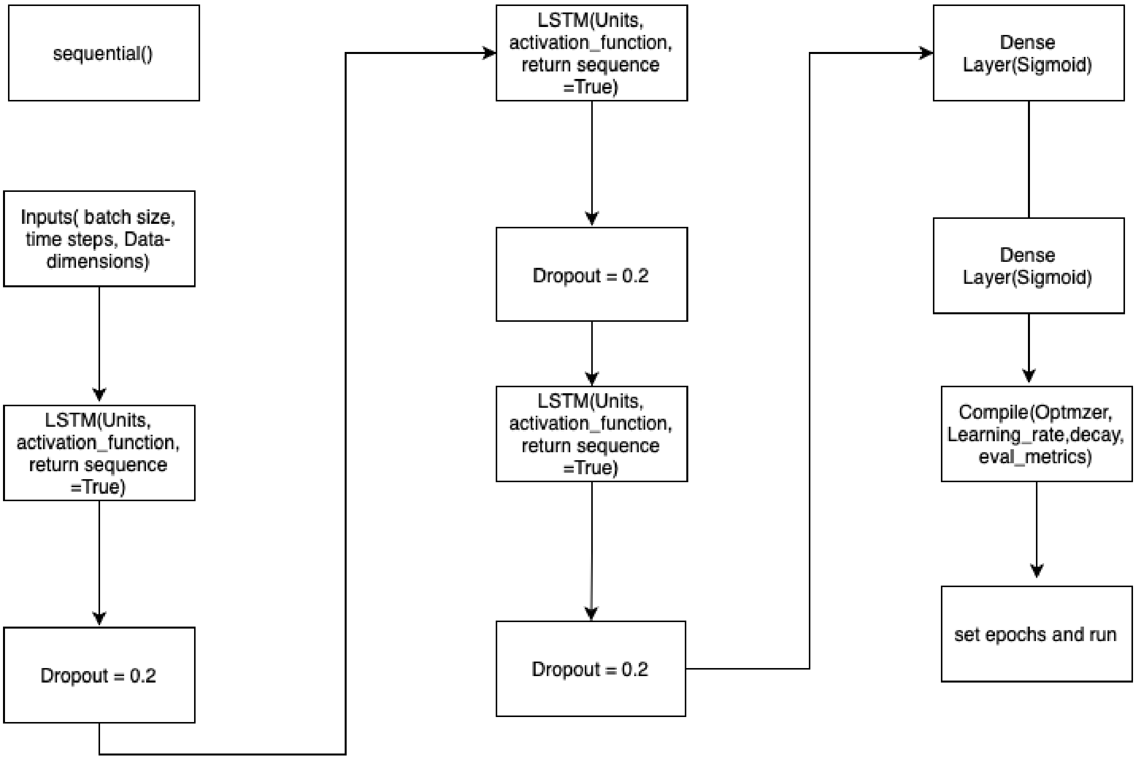


Figure 4.1: Structure of LSTM layer using sequential API

1. Firstly we start by defining a Sequential model API from the Keras framework, this allows us to create layers and add them on to our model therefore creating the structure for our model.
2. A key point towards designing a sequential model is defining our input layer, the model must know the shape of our input data, in this case it is a 2 dimensional array with a fixed batch size.
3. After the definition of the input layer, follows the main part of the model definition which is the LSTM layer. The number of hidden units must be defined for each layer and in the case of the first layer the input shape must be defined and the return sequence set to true so as to ensure the data is passed to the next layer and learning is continued in order to make it more consistent.
4. After each LSTM layer, we employ a dropout layer which randomly sets inputs to zero with a frequency of 0.2 at each step during training, which helps to prevent overfitting. Inputs not set to 0 are scaled up by 1/(1 - rate) such that the sum over all inputs is unchanged.
5. The output must match the label after the LSTM layers in order to produce a result. In the data processing stage covered in Chapter 5, labels are intended to be handled in one-hot encoding, which would result in a list of labels with only the label number's index indicated as 1. To make a prediction to the most likely class, define a dense layer with Sigmoid and set

only the maximum probability to 1 while setting all other probabilities to 0 to fit the one-hot encoding of the original label.

6. The user must define the model itself and compile it when all layers are defined. Since we have 3 classes of actions, we may define additional hyperparameters in the build API such as learning rate and its decay over epochs, dropout rates, and evaluation metrics.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 cu_dnnlstm (CuDNNLSTM)      (None, 2500, 64)          17408

 dropout (Dropout)           (None, 2500, 64)          0

 cu_dnnlstm_1 (CuDNNLSTM)    (None, 2500, 256)         329728

 dropout_1 (Dropout)         (None, 2500, 256)         0

 cu_dnnlstm_2 (CuDNNLSTM)    (None, 100)               143200

 dropout_2 (Dropout)         (None, 100)               0

 dense (Dense)               (None, 100)               10100

 dense_1 (Dense)             (None, 3)                 303

=================================================================
Total params: 500,739
Trainable params: 500,739
Non-trainable params: 0
_________________________________________________________________
```

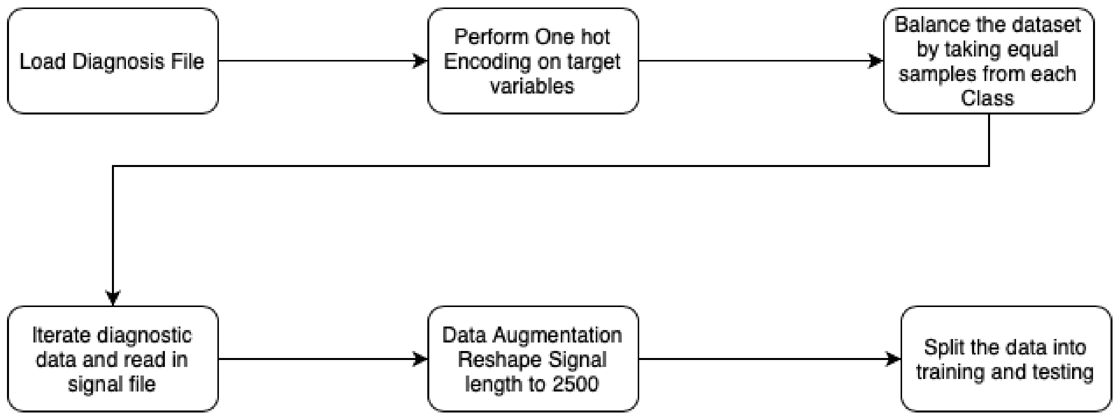
Figure 4.2: LSTM Model Design

Chapter 5

Data Processing

5.1 Steps of Processing

As introduced in Chapter 3, the ECG signals are stored in '.dat' files separate from the header data and the attributes as well as the ECG signals being different lengths . In order to use the data in the model experiments, several processing steps must be taken. The brief steps are shown in the figure below.



5.2 Dataset Loading

The WFDB python API was developed for reading in the physio net dataset. However, it only provides methods for the operation, not the entire process. Therefore we need to build an iterator for the data we need to be loaded in and saved for processing

5.3 Data Pre-processing

Data pre-processing is an essential phase before performing classification. This section describes different actions that take place in the pre-processing stage and returns a new transformed dataset. The initial data for this project contained a record file which contained the name and the file path for all the files in our dataset. the record file was used to load in the data based on the path from the system, there were a few inconsistencies with the record file as about 7 patient files mentioned were not able to be accounted for, this required us to remove these files from the record file so as to be able to load in all the files properly.

5.3.1 Data Cleaning

The process of re-examining and validating data in order to remove duplicate information, repair existing errors, and ensure data consistency is known as data cleaning. Deletes irrelevant and duplicate data in the original data, smoothes noise data, filters out data irrelevant to mining subjects, and handles missing values and outliers. The simplest technique to deal with invalid and missing numbers, for example, is to replace them with the mean, median, or mode value.
There wasn't much to do towards cleaning for this dataset as our signal values were arrays.

Errors: Before the labels could be extracted the diagnosis was stored in the form of arrays in our dataset, the value of the labels had to be extracted and delimited by underscores, for example the label, 'non atrial fibrillation' is replaced with 'non-atrial-fibrillation'. Use the replace function in python to change the entire dataset into an underscore delimited format.

5.3.2 Data Augmentation

Data Augmentation is required prior to data analysis for modern deep learning and ECG deep learning is no exception, data augmentation overall helps to alleviate overfitting and over confidence for neural networks. In signal processing, it mostly pertains to data normalisation, resampling and conversion into appropriate formats to fulfil the needs of classification tasks and algorithms. Data normalisation is the process of scaling data into a specified interval. The unit limit of the data is removed after normalisation, and all variables are in the same order of magnitude, allowing for complete comparative comparison.

There are typically 3 ways to do normalization for ECG signals

- Naïve Normalization
- Z-score normalization,
- Min-Max normalization

For this project the Z-score normalization technique was chosen as it is the most commonly used method for the pre-processing of ECGs, although the min-max method has recently started being adopted for pre-processing it was decided that the z-score was the safest bet. The naïve normalization was not considered for this project as it is best suited to computer vision and image classification due to the range of values being fixed within a certain range, although this is not the case for ECG signals. The z-score normalization is given by the equation below:

$$Z - score = \left(\frac{x - mean(x)}{std(x) + eps}\right) \cdot s + m$$

Where *x* is an ECG signal, *m*, *s* are given values, a small value of eps is added to avoid decision by zero.

For the augmentation of the dataset in this project gaussian noise was added to each signal. Finally, before classifying the data, the ECG signals are resampled to a fixed sampling rate so as to provide consistency as the signal length for the ECGs vary, it is imperative to find a standard length to which we can truncate the data without losing any relevant information.

Chapter 6

Experiments and Evaluations

After the data processing in the previous chapter, the data is to be passed through the model that was built in chapter 4. In this chapter we would conduct numerous experiments and examine the results.

6.1 Experimental Setup

6.1.1 Hyper Parameter Tuning

When working with neural networks it is essential to set up hyper parameters for our models such as number of epochs and learning rate, so as to get the best possible variation for the training of our models. Unlike the regular parameters in neural networks like the weights and bias which a tweaked by the algorithm during training the hyper parameters need to be set before the learning process begins. The hyper parameters we’ll be looking at in this section will be the number of epochs and the batch size because although the model does not automatically tweak these parameters, finding the best value for these parameters essentially aid in improving the accuracy of the model.

Number of Epochs

An epoch in machine learning means one complete pass of the training dataset through the algorithm. This epochs number is an important hyperparameter for the algorithm. It specifies the number of epochs or complete passes of the entire training dataset passing through the training or learning process of the algorithm. With each epoch, the dataset’s internal model parameters are updated, and the model tries to improve its performance by minimising the loss per epoch. When training, it is not uncommon to see epochs ranging up to thousands or even higher, depending on how difficult the task is. To determine the best number of epochs for our LSTM and GRU models we employed a Keras function called ‘Grid search’. This method allows us to pass an array with a different values for the epoch parameter and runs the model using each parameter to determine which epoch provides the best accuracy score. Furthermore, after a while the changes in test accuracy are very minimal even with increased epochs so it is very imperative to select a value where the accuracy starts slowing down as the increase in epochs the run time would also increase, thereby creating a trade-off between performance and speed which do need to be considered for this experiment as well. Generally with a more diverse dataset, the number of epochs would need to be large to make it perform better but since our dataset is fairly uniform, the number of epochs we need is not too high.

Batch Size

The batch size is a hyper parameter that defines the number of samples to work through before updating parameters. Batch sizes are choses due to the size of the data that is needed to go into the model. Practically, this is represented in the code by the batch size setting, which indicates how many examples we want to load in a single batch. The process by which we load a single batch is referred to as iteration. The model will calculate the gradient descent with error functions after completing one batch of calculations in order to carry out a backpropagation and update its weights and bias. Therefore, the quantity of iterations also refers to the number of times the network upgrades itself during a training epoch. It's interesting to note that accuracy does not always increase as batch size increases; in fact, when batch size exceeds a certain threshold, performance degrades. Iteration times, which decrease as batch size increases, caused the weights and bias to not update as frequently as they should have, which affected the learning outcome

Cross Validation

The K fold cross validation has become essential when training to classify multiple classes. This is because of its ability to split the data into different folds and ensure that each class has equal representation in each fold. For example, there are K subsets, one for each test set, and the rest for training. Cross validation was repeated k times, with each time a subset chosen as the test set, and the average recognition accuracy of cross validation for k times is used to calculate the result, the advantage of this being that all samples are used as training and test set. The value of 3 was chosen for this experiment as the size of our dataset was not that large in itself.

Layers

For the consistency of this project, we provided the same amount of layers for each model, although stacking more layers to the network would increase the models accuracy by giving it more parameters to learn. However just like the other parameters, there is a trade-off between the number of layers and run time. Due to this we settled on a maximum of 4 layers for this experiment.

6.2 Experimental Evaluations

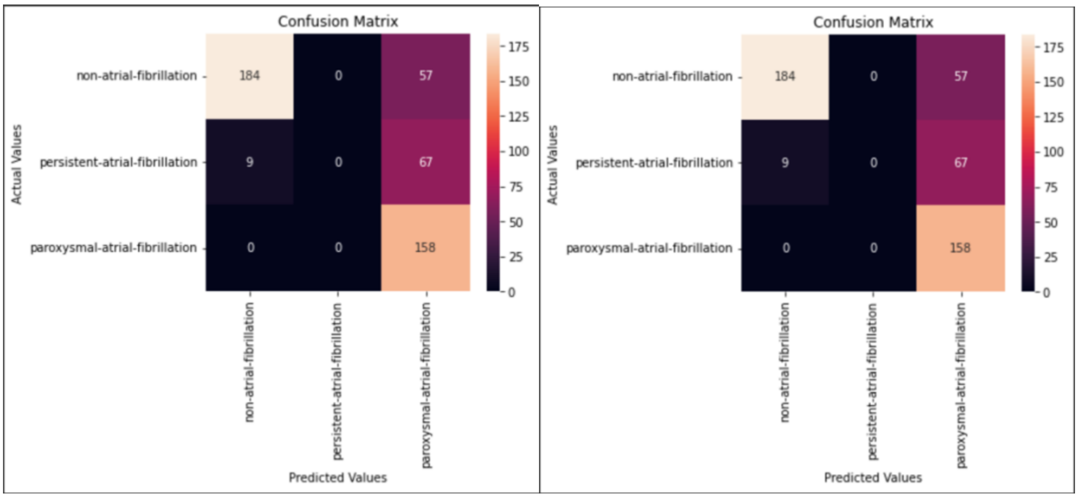


Figure 6.1: Result from both LSTM (left) and GRU(right) model

The models were tested using 475 labelled ECG signals as test data. Overall, the performance of both models were fairly similar with both having an equal amount of layers to ensure consistency. From the confusion matrix in figure 6.1 above, we can see that both models are fairly capable of distinguishing between AFIB and non AFIB but lack accuracy when it comes to distinguishing between the different types of AFIB classes we have present i.e., persistent vs Paroxysmal. For this iteration, we fed the model with an ECG signal of Length 1675, which is the smallest possible signal in our dataset. We also tried various signal lengths in order to determine if we were able to distinguish between the two to see if the length of the signal was the reason for the inaccurate classification of the two classes but we had to trade off with speed of execution and also had no difference in accuracy when experimenting with signal length. Although in the context of electrocardiography, it is more valuable to correctly classify true Atrial fibrillation that to correctly classify a normal heart rhythm, so a high sensitivity(recall) indicates that both models have learned the crucial features for distinguishing between AFib and Non AFib.

The models performance was evaluated using the criteria in figure 6.2 below:

	Precision	Accuracy	Recall	F1
LSTM model	0.67	0.72	0.72	**0.67**
GRU	0.67	0.72	0.72	**0.67**

Table: 6.1: Models performance judged on different metrics

Chapter 7

Conclusion

This project implements a deep learning networks to classify 2 lead ECG signals and differentiates between two different types of AFib and a normal human heart rhythm. The implemented networks are Recurrent Neural Networks which are perfect for signal processing and classification like ECGs. Overall I believe not all the main objectives of this project have been met and evaluated as our model was not able to accurately differentiate between the different types of Atrial Fibrillation as shown in our experiments and results due to various reasons such as shortage of time due to illness and other challenges.

The evaluation shows that both the LSTM and GRU models were on par when it came to testing our models, which was not expected as the LSTM was expected to perform better from initial observations.

7.1 Future Work

Since the primary objectives of this project have not been met, there are many areas of improvements that are needed for the next iteration of this project:

1. Improving Model accuracy: Exploring better model designs and enhancing neural network to improve accuracy and reduce false negatives as well as be able to classify persistent atrial fibrillation by enhancing our signal augmentation methods, as having an overfitted model is one of the possible reasons for poor accuracy as well as adjusting the threshold for the classes.
2. Model Varieties: More models could have been implemented and used to evaluate performance such as the bidirectional LSTM and convolutional networks like ResNet and DenseNet. Also, the implemented models are not fine-tuned, leaving space for optimisation therefore allowing a more comprehensive evaluation.
3. Computer Aided Diagnosis(CAD) system: when the issues with the model are fixed, this project could further be extended by developing a CAD system which uses this model and provides a very

7.2 Project Reflections

While working on this project there were a lot of concepts that were particularly new to me, even though I had the theoretical knowledge on those areas, I had barely scratched the surface when it came to actual implementation and working on a machine learning project from start to finish. The project was challenging in various aspects, some of them were because of the lack of solid background in signal processing and deep learning. This led to a lot of overthinking instead of actual implementation and testing out different things which delayed the initial timeline of this project.

Having no signal processing background meant I had to double my efforts to get a grasp of the processing techniques and methods I needed to use to actually train my models and then after getting to the training I was having an accuracy of about 40% and couldn't understand why. This wasted further time on trying to figure out what the source of the low accuracy was. I had to test my model by implementing it on a 12 Lead Arrhythmia dataset to try and find out what the issue was. It took me about a week and half to realise the issue was in the data being passed in to the model after processing which could have been easily fixed.

Another Issue I had was working with the length of the data as the length of the data was not consistent which hindered me when I was trying to pick the best length to pass to my model, the various iterations cost more time as having to load in the data and pad or cut the signals based on their length was way too time consuming, in the end it did not amount to much as I was unable to increase the accuracy with longer signals.

The project timeline outlined in the initial scoping and the planning phase mentioned in chapter 1, were not closely followed throughout the project due to illness as well as the initial work in chapters 5 and 6 taking longer than initially planned for.

In my personal opinion, now that the project has been somewhat complete, I don't believe it to be as difficult as I thought it to be, if only had communicated with my supervisor more about my issues, this project would have been implemented correctly. I am sad now due to the fact that I believe I could have taken this project in a better direction and implemented more models to test my data on. I saw a lot of models based on using a 1 dimensional convolutional model but didn't have enough time to implement it. While working on this project, I have learned a lot about machine learning, signal processing and computational biology which I enjoyed and it most definitely expanded my interest in the mentioned subjects for future work.

List of References

Ahmad, J., Farman, H. and Jan, Z. (2019) ‘Deep Learning Methods and Applications’, in Khan, M., Jan, B., and Farman, H., *Deep Learning: Convergence to Big Data Analytics*. Singapore: Springer Singapore (SpringerBriefs in Computer Science), pp. 31–42. Available at: https://doi.org/10.1007/978-981-13-3459-7_3.

Alzubi, J., Nayyar, A. and Kumar, A. (2018) ‘Machine Learning from Theory to Algorithms: An Overview’, *Journal of Physics: Conference Series*, 1142, p. 012012. Available at: https://doi.org/10.1088/1742-6596/1142/1/012012.

Atrial fibrillation - Symptoms and causes (no date) *Mayo Clinic*. Available at: https://www.mayoclinic.org/diseases-conditions/atrial-fibrillation/symptoms-causes/syc-20350624 (Accessed: 15 August 2022).

Chang, C.-H., Ko, H.-J. and Chang, K.-M. (2010) ‘Cancellation of high-frequency noise in ECG signals using adaptive filter without external reference’, in *2010 3rd International Conference on Biomedical Engineering and Informatics*. *2010 3rd International Conference on Biomedical Engineering and Informatics*, pp. 787–790. Available at: https://doi.org/10.1109/BMEI.2010.5639953.

Cho, K. *et al.* (2014) ‘Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation’. arXiv. Available at: http://arxiv.org/abs/1406.1078 (Accessed: 13 August 2022).

Daqrouq, K. *et al.* (2014) ‘Neural network and wavelet average framing percentage energy for atrial fibrillation classification’, *Computer Methods and Programs in Biomedicine*, 113(3), pp. 919–926. Available at: https://doi.org/10.1016/j.cmpb.2013.12.002.

Gardner, M.W. and Dorling, S.R. (1998) ‘Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences’, *Atmospheric Environment*, 32(14), pp. 2627–2636. Available at: https://doi.org/10.1016/S1352-2310(97)00447-0.

Gavhane, A. *et al.* (2018) ‘Prediction of Heart Disease Using Machine Learning’, in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1275–1278. Available at: https://doi.org/10.1109/ICECA.2018.8474922.

Hinton, G. *et al.* (2012) 'Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups', *IEEE Signal Processing Magazine*, 29(6), pp. 82–97. Available at: https://doi.org/10.1109/MSP.2012.2205597.

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

Hornik, K., Stinchcombe, M. and White, H. (1989) 'Multilayer feedforward networks are universal approximators', *Neural Networks*, 2(5), pp. 359–366. Available at: https://doi.org/10.1016/0893-6080(89)90020-8.

Martis, R.J. *et al.* (2013) 'Automated detection of atrial fibrillation using Bayesian paradigm', *Knowledge-Based Systems*, 54, pp. 269–275. Available at: https://doi.org/10.1016/j.knosys.2013.09.016.

Raina, R., Madhavan, A. and Ng, A.Y. (2009) 'Large-scale deep unsupervised learning using graphics processors', in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09. the 26th Annual International Conference*, Montreal, Quebec, Canada: ACM Press, pp. 1–8. Available at: https://doi.org/10.1145/1553374.1553486.

Rivera-Ruiz, M., Cajavilca, C. and Varon, J. (2008) 'Einthoven's String Galvanometer', *Texas Heart Institute Journal*, 35(2), pp. 174–178.

Wang, J. (1993) 'Analysis and design of a recurrent neural network for linear programming', *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 40(9), pp. 613–618. Available at: https://doi.org/10.1109/81.244913.

Warner, H.R., Toronto, A.F. and Veasy, L.G. (1964) 'Experience with Baye's Theorem for Computer Diagnosis of Congenital Heart Disease*', *Annals of the New York Academy of Sciences*, 115(2), pp. 558–567. Available at: https://doi.org/10.1111/j.1749-6632.1964.tb50648.x.

William L Hosch (2021) *machine learning | artificial intelligence | Britannica*. Available at: https://www.britannica.com/technology/machine-learning (Accessed: 9 August 2022).

Yuki, Hagiwara *et al.* (2018) *Computer-aided diagnosis of atrial fibrillation based on ECG Signals: A review | Elsevier Enhanced Reader*. Available at: https://doi.org/10.1016/j.ins.2018.07.063.

Zhou, Z.-H. (2021) *Machine Learning*. Springer Nature.

Appendix A
External Materials

Keras: This library is used to build model and customise cells and layers. Link: https://keras.io

Tensorflow: This library is used to create TFRecord files and as the backend of Keras. Link: https://www.tensorflow.org

Scikit-Video: This library is used to visualise the data. Link: http://www.scikit-video.org

Numpy: This library is used to implement array operations. Link: https://numpy.org

Keras: This library is used to build model and customise cells and layers. Link: https://keras.io

Tensorflow: This library is used to create TFRecord files and as the backend of Keras. Link: https://www.tensorflow.org

Scikit-Video: This library is used to visualise the data. Link: http://www.scikit-video.org

Numpy: This library is used to implement array operations. Link: https://numpy.org

Appendix B

Ethical Issues Addressed

No ethical issues apply for this project as no person related data was processed. All the patient data was scrubbed by the data provider for anonymity