ECG Arrhythmia Classification Using 1-D Convolution Neural Network Leveraging the Resampling Technique and Gaussian Mixture Model

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering of the University of Asia Pacific

Submitted By

Fahmeda Akter
Student ID: 17101060

Mst. Farzana Akhtar Lubna
Student ID: 17101064

Md. Remon Hasan Apu
Student ID: 17101086

Supervised By

Tanjina Helaly
Assistant Professor

Co-Supervised By

Tanmoy Sarkar Pias
Lecturer

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF ASIA PACIFIC

June, 2021

CERTIFICATION

This is to certify that the work presented in the thesis is an outcome of the investigation carried out by the authors under the supervision of Assistant Professor Tanjina Helaly and Co-Supervision of Lecturer Tanmoy Sarkar Pias, Department of Computer Science and Engineering , University of Asia Pacific. It is declared that this thesis has been submitted only for the award of graduation.

Authors

Fahmeda Akter

Mst. Farzana Akhtar Lubna

Md Remon Hasan Apu

Signature of the Supervisor

Tanjina Helaly

Assistant Professor
Department of Computer Science and Engineering
University of Asia Pacific (UAP)

Signature of the Co-Supervisor

Tanmoy Sarkar Pias

Lecturer
Department of Computer Science and Engineering
University of Asia Pacific (UAP)

Acknowledgments

First of all thanks to Almighty Allah for giving the patience to successfully complete this research work. Then We want to thank Tanjina Helaly madam for deciding to oversee our study with us. Her patience and trust in us have been a source of motivation and without her great help, inspiration and influence this research would not have been possible. We think her contribution to this research work is worthy of being reciprocated with great gratitude. We would also like to express our special thanks to our co-supervisor Tonmoy Sarkar Pias sir for his work assistance and encouragement. A special thanks to the Study Committee for taking the time as part of our undergraduate program, to review and assess our study.

ABSTRACT

The electrocardiogram (ECG) is one of the simplest and oldest tools to assess the heart condition of cardiac patients. Heart diseases have emerged as one of the leading causes of death all over the world. According to the world health organization (WHO), millions of people are dying every year from heart-related diseases. A classification model that can early detect Arrhythmia will be able to reduce this number by manyfold. Many researchers are working in this area and proposed many deep learning and Machine Learning based models for Arrhythmia classification. These models have high accuracy but require a machine with high computational power. Hence, these models are not sustainable options for the practical field. In this paper, we have proposed a 1D Convolutional Neural Network (CNN) model with high accuracy and low computational complexity. Our proposed methodology is appraised on the MIT-BIH arrhythmia dataset. We achieved overall 98.25% accuracy into five classes with an f1 score of 98.24%, precision 97.58%, and recall 96.79% which is better than previous results classifying arrhythmias. We can claim that our proposed method is better than most other existing models because of the higher accuracy with a simple architecture that can be run on an edge device with relatively low hardware configuration.

Keywords: ECG signal, Classification, Arrhythmia, 1D CNN.

TABLE OF CONTENTS

LIST OF TABLES

List of Figures

CHAPTER 1

INTRODUCTION

Heart disease is one of the leading causes of human death [44]. It can affect all genders, all races, and all ethnic groups. Arrhythmia, a type of heart disease, where the heart beats too quickly, too slowly, or with an irregular pattern. Arrhythmia beats are uncomfortable and can be life-threatening. Arrhythmia may cause symptoms like chest pain, shortness of breath, palpitation, light-headedness, etc. Arrhythmia leads to other kinds of heart diseases. Therefore, studies of Arrhythmia are important in this research. Early detection of Arrhythmia can help the cardiologist and doctors to prevent the death rate due to heart diseases [45]. The electrocardiogram (ECG), a test that captures the variability of the heart, helps to detect abnormalities in heart or heart rhythm.

ECG is one of the simplest and oldest tools to assess the heart condition of cardiac patients. In ECG, a tiny electrical impulse is produced by the heart with each beat which helps to regulate the different chambers of the heart and pump the blood out to the whole body. This electrical impulse is recorded by the ECG and displays as a trace on paper. The ECG signal has three main components; P wave, QRS complex, and T wave. Each of these waves can have variable amplitude and duration. The length and intervals between components provide useful information about heart conditions. Each of these components has its normal range which may vary a little between males and females. By tracking the variation of R peaks from ECG different kinds of arrhythmia can be detected.

Many researchers are working in this area and proposed many deep learning and Machine Learning based models, described in chapter 3, for Arrhythmia classification. Some of these models have high accuracy but require a machine with high computational power. Hence, these models are not sustainable options for the practical field.

1.1 Research Statement

In this research, we have proposed a 1D Convolutional Neural Network (CNN) model with high accuracy and low computational complexity. Our proposed methodology is appraised on the MIT-BIH arrhythmia dataset. We achieved overall 98.25% accuracy into five classes with an f1 score of 98.24%, precision of 97.58%, and recall of 96.79% which is better than the existing arrhythmia classification models. We can claim that our proposed method is better than most other existing models because of the higher accuracy with a simple architecture that can be run on an edge device with relatively low hardware configuration. The main contributions of our proposed system are:

- A combination of resampling and the gaussian method is used. Accordingly, it contains the simple method of an algorithmic model for pre-processing.
- We achieved state-of-the-art performance in the Arrhythmia classification of ECG by using a computationally light 1D convolution neural network.

1.2 Paper Organization

The other parts of the paper are organized as follows: Background in chapter two, Related research works in chapter three, Methodology in chapter four, Introduction to our dataset in the experiments section, Classification results, and discussion in chapter five. Comparison with existing algorithms in Table 5.5 and as follows Table 3.1. The conclusion and future work in chapter six and the rest of the part are references and implementation in chapter seven.

Chapter 2

Background

2.1 ECG

ECG is the graphical representation of electrical activity which is generated by the cells of the heart [fig 2.1]. Thus, ECG stands for Electrocardiogram. A revolutionary innovation by the great Willem Einthoven, a Dutch physician in 1895. Our body contains elements like sodium, potassium, magnesium, and calcium that have electrical characteristics which generate electrical activity in our body. Electrical activity is the basis to make any contraction in the human body thus maintaining the rhythm and activity of the heart [1]. Any abnormality in the heart hence can be easily identified by measuring this electrical activity of the body. To measure these pinpoint sensitive electrical signals around the heart ECG plays a vital role and determining any abnormalities in cardiac function. As it is a graphical representation the character of the graph is not like a normal graph [2]. The graphs are 1mm by 5mm in size, the machine is voltage remains at 10mm/mV and the printer speed is limited at 25 mm/sec which is a standard value. ECG recording is a remote monitoring system. It has different types such as the Holter monitor. It

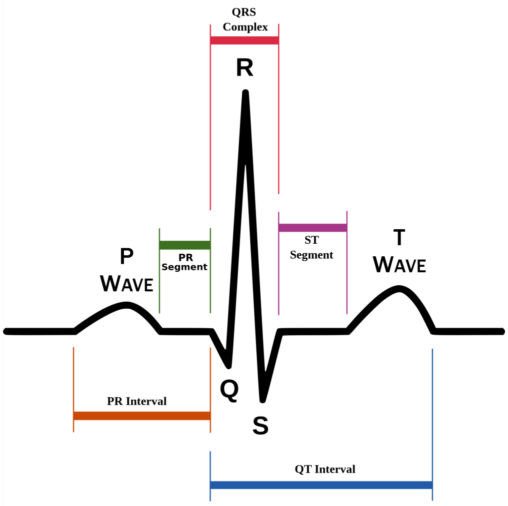

Figure 2.1: Basic of Electrocardiogram (ECG) Signal

is a wearable device that records a continuous ECG. It usually records for 24 to 48 hours. There is a portable device like a Holter monitor. It records the condition of the heart for only a few minutes.

When anyone feels symptoms it is needed to push a button. Doing ECG has no risk because it is a safe procedure and it has no electrical shock. There are 12 leads in a standard ECG machine. The electrodes are define as leads. It measure the variation of electrical potential between two different points of the body. It is known as bipolar and the virtual zero electrical point known as unipolar. There are 3 standard and 3 augmented limb leads and 6 precordial leads. The particular axis of these leads represents the viewpoint of the heart from which they look for. There are two types of leads Gold Berger (aV frontal) and Wilson and Co-workers (chest lead) [3].

The ECG represents simultaneously the P wave, QRS complex, and T wave. The P wave is the first electrical signal. It is originated from the atria. The sum of the electrical signals from the two atria. Depolarisation of the ventricles Einthoven Triangle Lead (Standard) is the output of the largest part of the ECG signal. It is defined as the QRS complex. The first negative deflection is the Q wave. The next upward deflection is the R wave. The crosses of the isoelectric line are provided by the R wave. Therefore, it becomes positive. Also, the next deflection downwards is called the S wave. The crosses of the isoelectric line are provided by it which becomes negative. It occurs before the isoelectric baseline. The most leads in an upright deflection of variable amplitude are called T waves. It has some normal intervals such as the PR interval. It measured from the starting of the P wave and the rest of the deflection of the QRS complex.

The normal range of the QRS complex is between 120ms to 200ms. It is measured from the first deflection at the isoelectric line. The normal range is up to 120ms. From the first deflection of the QRS complex to the end of the T wave, the QT interval is measured. The normal range of T waves is up to 440ms. It has the variation with heart rate. Also, the duration is longer in females. Standard ECG provides a maximum estimation of the heart rate from the recording of ECG. Along the horizontal axis, it represented the duration of 250mm each second of time [40].

2.2 Clinical ECG Interpretation

The morphological assessment of the waves and the intervals in an ECG graph is known as ECG interpretation [fig 2.2]. It is a vital skill to interpret an ECG as from this result a doctor or specialist could find the abnormalities and treatment plans [4]. There are logical order to interpret an ECG to avoid any misinterpretations those are; rate and rhythm, the intervals of conduction, the axis of cardiac, ST segments, QRS complex, and T waves. From the sinoatrial node, the rate and rhythm depolarisation is originated. It is known as sinus rhythm. If the depolarization is originated from a different part of the heart, the rhythm is defined as Arrhythmia. The rate of the heart is between 60 and 100 beats per minute. The P-R interval should be between 120-200ms long. It represents the time between the ventricular contraction and atrial contraction. It is a representation of the delay in conduction at the atrioventricular (AV) node. It prevents the atria and ventricles from contracting at the same time [7].

The conduction from the atria to the ventricles is abnormally quick when the P-R duration is less than 120ms. The PR interval detects the conduction defect which is known as first-degree heart block. The duration PR interval should be between 120-200ms. The ventricular tachycardia is caused by prolonged QT duration[39]. The average direction of the electrical wave traveling

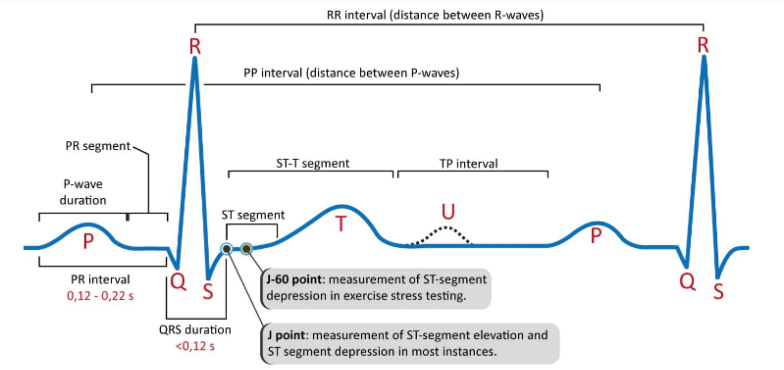

Figure 2.2: ECG arch with common waveform

through the ventricles in the vertical plane is defined by the cardiac axis. The duration of the normal cardiac axis is between ninety-degree using lead I as the zero degrees reference point [36]. It represented that a wave of depolarization is enlarging towards leads. There are two axes of it which are right and left. Left axis deviation represents left ventricular hypertrophy and conduction defects. The right axis deviation represents the right pulmonary embolism, ventricular hypertrophy, and congenital heart diseases [37].

2.2.1 The QRS Complex

The QRS complex [fig 2.3] constitutes the distribution of the ventricles. However, the QRS complex may not always define the presentation of all three waves. It is called a Q wave when the duration of the QRS complex is negative.

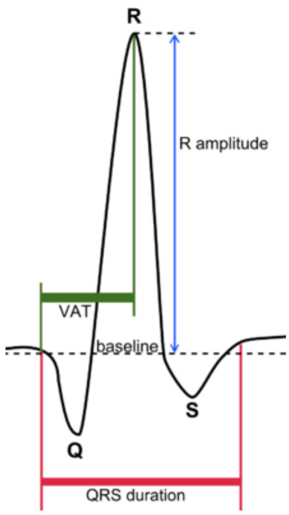

Figure 2.3: The QRS Complexes

The first positive deflection in the QRS the complex is called the R wave. The negative deflection which follows the R wave is called an S wave. By the left ventricle, the electrical vector is rendered. It is many times higher than the vector rendered by the right ventricle. The demonstration of left ventricular depolarization is represented the QRS complex [8].

2.2.2 The ST Segment

The ST segment is the flat wave. Isoelectric represents flat on the baseline of the wave. It is neither a positive nor negative section of the ECG between the rest of the S wave and the starting of the T wave [35]. The ST segment defines the starting of ventricular duration and the junction between the rest of the QRS complex. Sometimes it is called the J point [fig 2.4] when it starts from the ST segment. The most important cause of ST-segment abnormality is called myocardial Arrhythmia. There are two kinds of ST-segment deviations. The ST-segment depression indicates

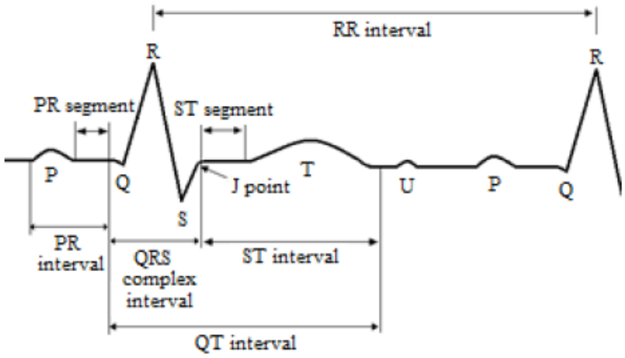

Figure 2.4: Arch of J point and ST segment

that the ST segment was misplaced. It is below the level of the PR segment. It is in a higher place to the level of the PR segment. The J point represents the point where the ST segment starts [9].

2.2.3 The T-Wave

The T wave illustrates the replication of the ventricles [fig 2.5]. The most features leveled wave in the ECG is the T wave. The patch of wave should be fluent from the ST segment to the T-wave. The T-wave is asymmetric when it produced descending arch. The absolute refractory period is obtained from the starting of the QRS complex and the T wave.

Due to the replication of the membrane, the T wave is obtained positive [34]. Because of changing the T wave many cardiac and non-cardiac conditions may occur. It is including the low-amplitude T waves and inverted T waves which are abnormal. Except in the right precordial leads the T wave is in the same direction as the QRS. Also, when the first half moving more slowly than the second half the T wave is asymmetric. The T wave is always upright in leads I, II, V3-6 in the normal ECG. It is usually inverted in lead aVR. The direction of the QRS may vary according to the age of the patient [10].

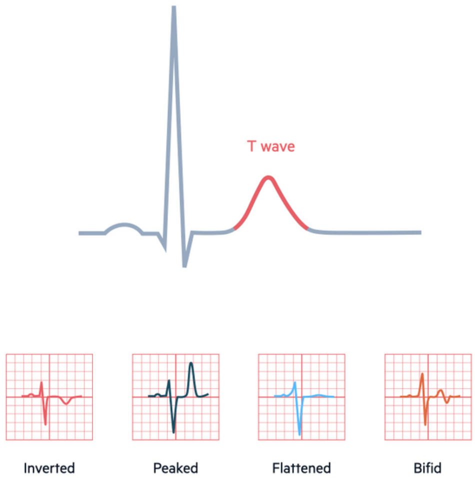

Figure 2.5: Arch of T-wave

2.2.4 The U-wave

The last phase of ventricular replication is defined as U waves [fig 2.6]. The direction of the U wave is positive usually. It is the same as the T wave [33]. The ratio of the magnitude of the U-wave is one-fourth of the T-waves magnitude. The U-wave sometimes remains elusive because of the slow heart rates. Normally, positive T waves appear with the negative U waves. And the abnormal finding can be noted in left ventricular hypertrophy and myocardial Arrhythmia. In the ECG the U wave is the deflection that represents the waves of duplication (QRS) and replication (T) of the heart chambers. U waves are thought to define the replication of the Purkinje fibers.

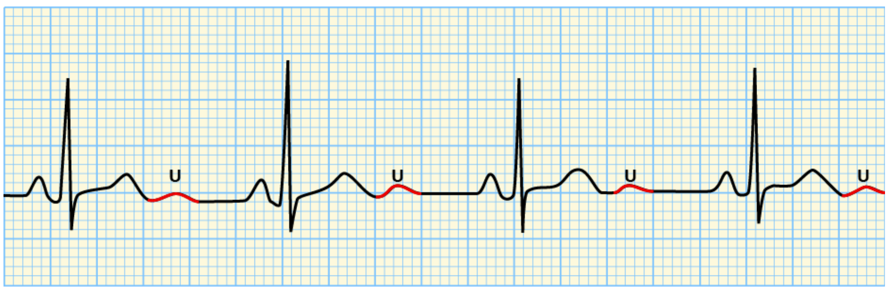

Figure 2.6: Arch of U-wave

2.2.5 The QT and QTc Duration

The final duration of ventricular duplication and replication is presented by The QT duration [fig 2.7]. It estimates The onset of the QRS complex to the end of the T-wave. With the QT duration, the heart rate is inversely connected. The slower heart rates are enlarged by the QT arrangement. The higher heart rates are decreased by the QT arrangement also. When the heart rates are synthesized with QT arrangement that refers to the corrected QTc arrangement [32]. Therefore, The QTc arrangement enhances the risk of ventricular arrhythmias.

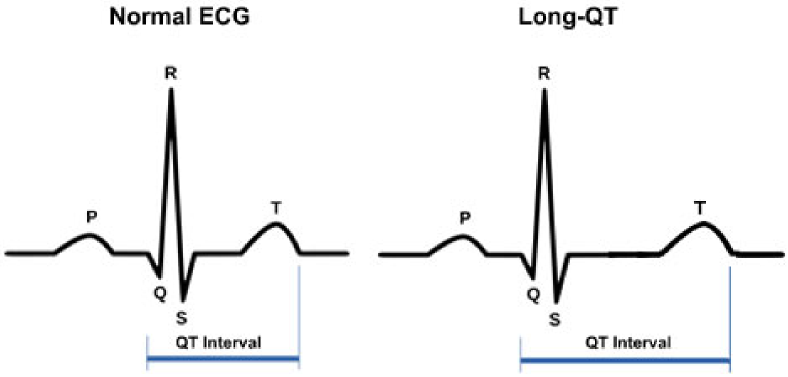

Figure 2.7: Arch of QT Duration

2.2.6 The P-wave

Usually, the ECG explanation starts with an arch of a P-wave [11]. The P-wave describes as a smooth arch. As muscle mass is small by the atria, the P-wave is small. The P-wave vectors go towards in descending order and to the left of the front plane at the time of sinus rhythm [Fig 2.8].

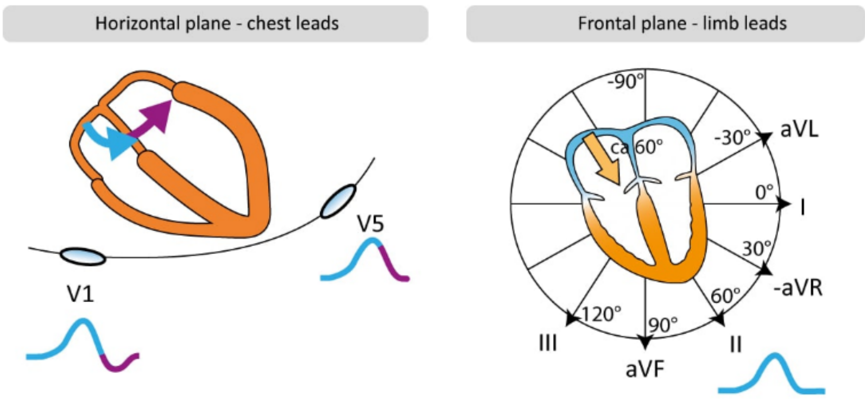

Figure 2.8: Horizontal and Front plane P wave

2.2.7 The PR Interval and PR Segment

From the onset of the P-wave, the PR interval start. at the onset of the QRS complex it is finished [fig 2.9]. By the PR interval, atrial depolarization is started. And, also the start ventricular depolarization is presented in the PR interval. To determine the natural speed of the impulse conduction from the atria to the ventricles the PR interval [fig 2.10] is picked. The duration of the PR interval is between 0.12s to 0.22s. The PR segment is the flat line between the end of the P-wave and the onset of the QRS complex. Through the atrioventricular node, slow impulse conduction is obtained [20]. The PR segment extracts the Arrhythmia features from the baseline to the ECG arch.

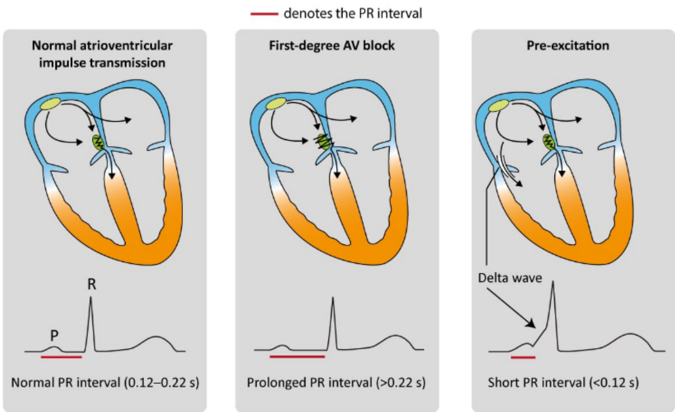

Figure 2.9: The PR Interval

The PR segment represents the flat line between the end of the P-wave and the end of the QRS complex. It represents the slow impulse conduction of the atrioventricular node.

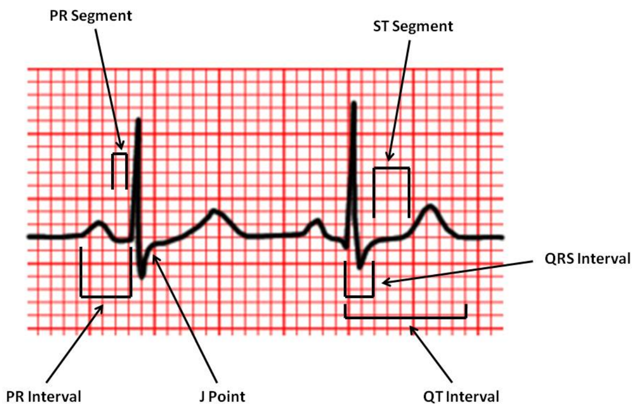

Figure 2.10: The PR Segment

2.3 Heart Rhythm

A rhythm has represented the heartbeats displaying the waveforms on the ECG [30]. The law of similarity defines the same origin of the impulse. The heart pacemaker with normal circumstances which is regarded as sinus rhythm is defined as a sinoatrial node [22].

An arrhythmia means an abnormal heart rhythm that is not physiologically measured. The concluding process is described with importance because rhythms that are physiologically measured should not be described as abnormal. Sinus bradycardia is usually found in athletes during sleep. Therefore, for that scenario, it is not regarded as abnormal. Also, sinus bradycardia is growing during physical exercise which is regarded as abnormal because heart rate increases during physical exercise [31].

2.4 Arrhythmia Background

Arrhythmia is nothing but abnormal rhythm or irregularity of the heartbeat [fig 2.11]. Another meaning of heartbeat is that it works too quickly, too slowly, or with an irregular pattern. It is a disorder of the heart. Arrhythmia beats are distressing, it feels like the heart is beating beyond the chest [21]. Arrhythmia indicates the heartbeat is running too fast or too slow. When the heartbeats are too fast it is defined as tachycardia. And when it is slow that is defined as bradycardia. There are many symptoms such as palpitation which is caused by arrhythmia. Therefore, many more have more serious consequences. It also causes sudden death. Anxiety is easily created by Arrhythmia.

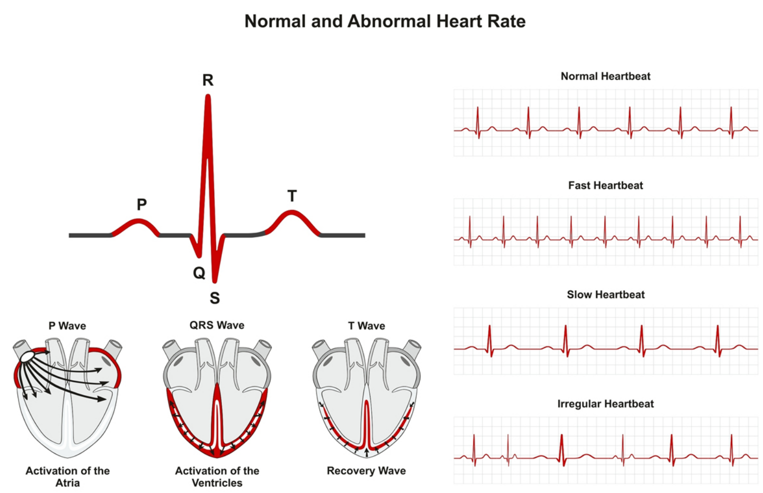

Figure 2.11: Difference between Normal and Abnormal Heartbeats

Heart attack is occurring because of Arrhythmia. A heart attack is happened because of changes in the heart structure. The most of reasons for heart diseases which are caused by Arrhythmia are blocked arteries in the heart, having high blood pressure, having diabetes, sleep apnea, infection with COVID-19. Too much smoking and drinking are also responsible for heart diseases. The adoption of the drug, certain medications, and supplements are also responsible. Including the effection of cold and taking nutritional supplements are responsible for Arrhythmia [29].

There are two kinds of Arrhythmia (i) Tachycardia caused by the fast heartbeat when the heartbeat rate is greater than 100 beats a minute, (ii) Bradycardia caused by the slow heartbeat when the heart rate is less than 60 beats a minute. Arrhythmia symptoms can be a start from the chest. The symptoms are chest pain, breath shortness, anxiety, etc. Arrhythmia can refrain from maintaining a heart-healthy diet. Also, remaining physically active and keeping a healthy weight. It also refrains from smoking, limiting or avoiding caffeine and alcohol, reducing stress. Severe stress and anger can cause heart rhythm problems[23].

2.4.1 Mechanisms of Arrhythmia

Arrhythmia begins [fig 2.12] when the impulse formation is not normal or when the impulse transmission is not normal or both are not normal [28]. Two mechanisms of abnormal impulse formation occur Arrhythmia. They are: (i) abnormal automaticity and (ii) triggered activity. Abnormal automaticity refers to four structures that are narrated step by step:

- The Sinoatrial Node: The sinoatrial node is defined as a primary part of the heart. It manages the heart rhythm during natural aspects of acting. The rhythm is defined as sinus rhythm [26].
- The Atrial Myocardium: The aspects of atrial myocardial cells are surrounded by the crista terminals. It enters into the coronary. Naturally, the rhythm is not able to pass through the cells of the conduction system [27].

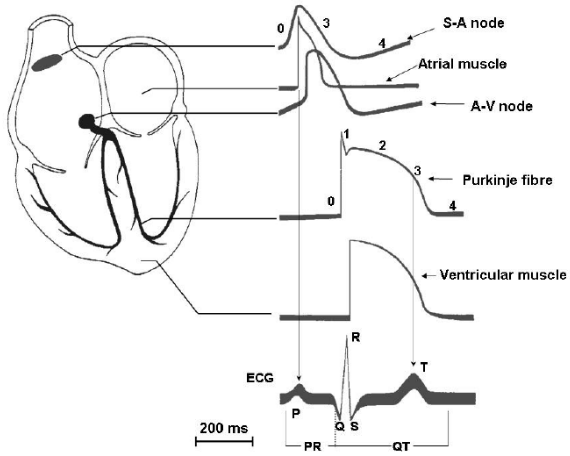

Figure 2.12: Electric Activity in the Myocardium

2.4.2 Sinus Arrhythmia

There are the same reasons phenomenon of sinus arrhythmia and respiratory sinus arrhythmia [fig 2.13]. Except for the fact of heart rhythm irregularity, sinus arrhythmia fulfills the criteria of sinus rhythm [24]. The phenomenon is defined by the heart rate variation caused by respiration [25]. Because of increasing vagal tone the heart rate decreases during inspiration. Among older individuals, there is no sinus arrhythmia which is found normal.

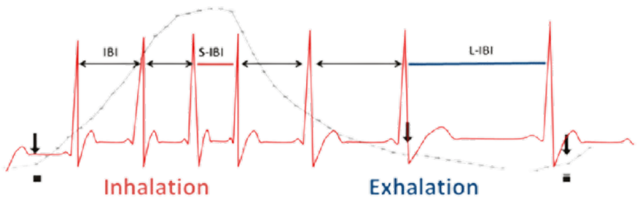

Figure 2.13: Respiratory Sinus Arrhythmia

CHAPTER 3

LITERATURE REVIEW

One dimension convolution neural network is a deep learning model. Deep learning is the newest achievement of machine learning. One dimension convolution neural network is focused

Table 3.1: Related research works

Paper	**Publication Year**	**Class**	**Preprocess Method**	**Feature Extraction**	**Classification Method**	**Accuracy**
Jiang et al. [17]	2007	N,S,V,F,Q	bandpass filters	hermite transform	blocked nerural network	96.66%
Zadeh et al. [18]	2011	N,L,R,A,V	bandpass filter	continuous wavelet transform	support vector machine+genetic algorithm	97.20%
Martis et al. [15]	2013	N,L,R,A,V	Wavelet	pan-tompkins and principle component analysis	neural network+support vector machine	93.00%
Joshi et al. [12]	2014	N,S,V,F,Q	wavelet	wavelet component analysis	support vector machine	86.40%
Ismaiel et al. [16]	2015	N.L.R.A.V	digital filters	discrete wavelet	nnws	94.00%
Zubiar et al. [14]	2016	N,S,V,F,Q	bandpass filter	convolution neural network	softmax	92.70%
Jose et al. [13]	2017	N,L,R,A,V	wavelet	wavelet	probabilistic neural network	92.70%
Dan et al. [19]	2017	N,L,R,A,V	wavelet Combination	1D-cnn	softmax	97.50%

on one dimension signal. It also focused on the dimension data repositories. There are different dimensions of convolution neural network is present in the deep learning era. Different dimension of convolution neural network is used for manipulation and feature extraction of a different dimension of data. In this work, we use one dimension of ECG data for the classification of Arrhythmia into five major classes.

The arrhythmia classification is mainly based on two parts. They are (i) Data Preprocessing (ii) Classification. In one dimension convolution neural network, several preprocessing techniques are used. In this work, we proposed a mixture of resampling and gaussian methods as a preprocessing technique. 1D CNN contains the convolution layers which are responsible extract features from the raw input of 1D ECG signals. It consists of different sizes of kernels or filters. It is also responsible for extracting the high levels features of edges. Different kinds of activation functions are used in 1D CNN. Rectified Linear Unit (ReLU) as an activation function. When the peaks of the signal are too large the pooling layers segment can reduce the number of parameters. Spatial grouping is also known as subsample or downsample that reduces the dimensionality of a map but keeps important features. Finally, a fully connected layer originates the output of the classification.

3.1 Preprocessing and Classification Methods

A one-dimensional method is proposed in Dan et al. [19] which achieved an overall accuracy of 97.50%.

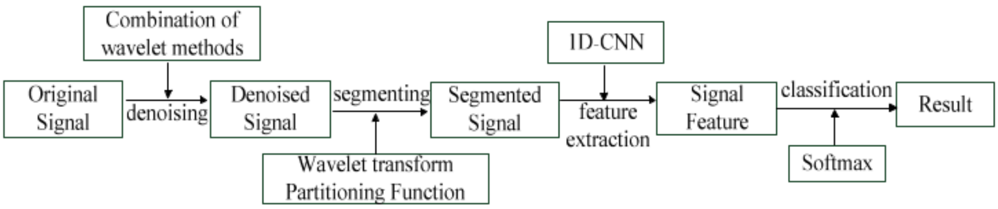

Figure 3.1: The flow chart of model building of Dan et al.

In [fig 3.1] the whole classification method is described. It contains the four major steps: data preprocessing, segmentation, feature extraction, and arrhythmia classification. At first, the high-frequency noises are filtered by using the wavelet threshold method. For achieving the baseline drift with low-frequency wavelet transform and reconstruction algorithm is used. Therefore, the processed segments of data are used for the input data for arrhythmia classification. Mainly, an ECG signal contains three types of noise. They are baseline drift, normal noise, and power line interference. For getting the proper R peak the signals need to be denoised. In this work, wavelet threshold and wavelet decomposition are used for denoising the signals [fig 3.2].

A convolution neural network has two parts. They are feature extraction and classification. In the Dan [17] work a 1D CNN is used as feature extraction and softmax is used as a classification method. Mainly, feature extraction extracts effective features. Ten layers of architecture are proposed for the arrhythmia classification into five classes. The architecture is described in detail in [fig 3.3].

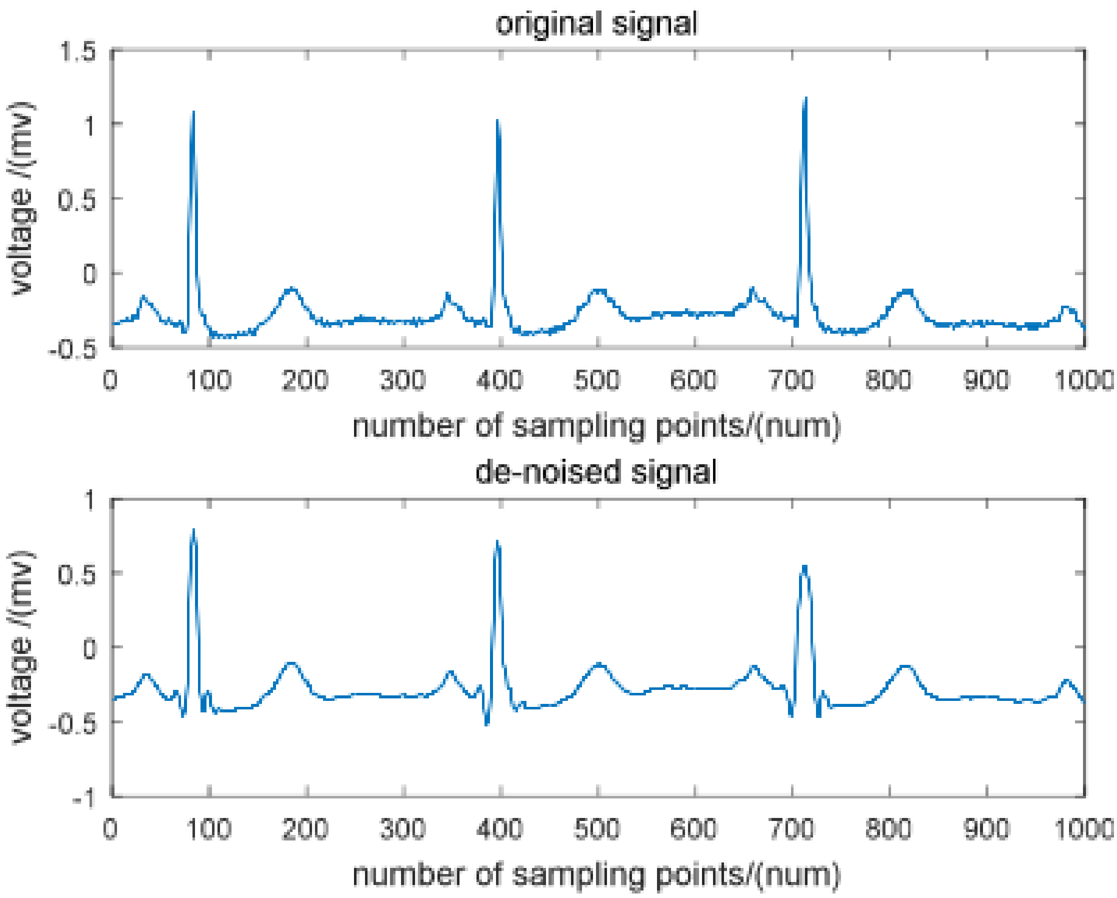

Figure 3.2: The denoising diagram

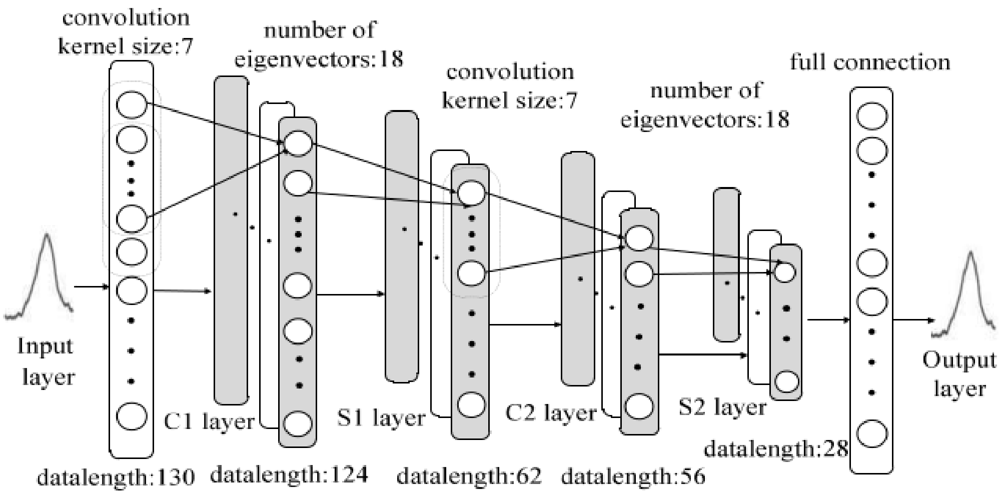

Figure 3.3: Proposed 1D CNN architecture

Preprocessing method is an efficient step for the data mining process in Joshi et al. [12]. ECG signals can contain various kinds of noise. To improve the Signal Noise Ratio (SNR), those noises need to be reduced. To detect the subsequent fiducial point, those improving (SNR) helps a lot. The pre-processing is based on baseline wander correction. And different kinds of baselines wander detached by the wavelets-based approach in [12]. The Heartbeat segmentation method is responsible for extract ECG features for QRS detection. The preprocessing step is responsible for detaching various kinds of noise from ECG signals. It can identify individual heartbeats. A Heartbeat represents the P wave, the QRS complex, and the T wave. There is also extra waves, the U waves which are sometimes visible, but not all the time.

In the work of Joshi [12], independent component analysis (ICA) is a computational method to extract the hidden factors that underlie sets of random variables of signals. ICA represents an efficient model for the observed multivariate data. Those are typically given as a large database of samples. ICA is used for feature extraction [fig 3.4]. In signal processing, ICA is widely used. And it can solve blind source separation problems.

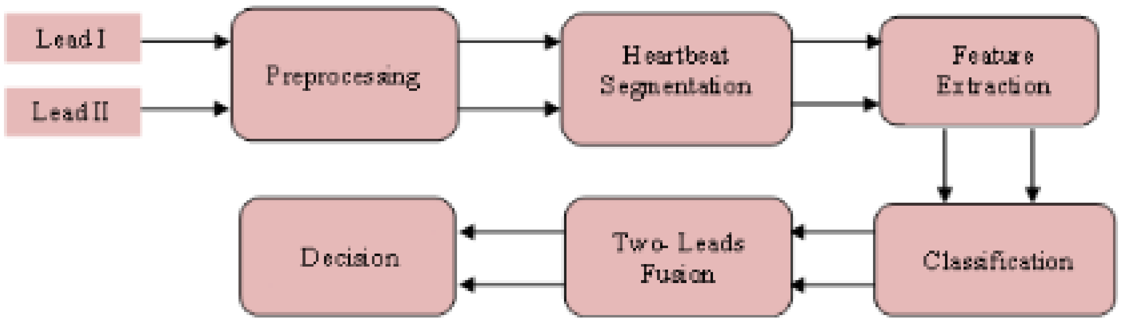

Figure 3.4: The flow chart of model building of Joshi et al.

In the Joshi et al. [12] work, the wavelet series represents the integrable function. The main advantage of the Wavelet Transform is compared to the Fourier Transform. The practical application of the Wavelet Transform is analyzing the ECG signals. A wavelet is a computational function. It is used to separate a function into different components. Wavelet transform is capable of performing analysis in both the time frequency domains. ECG signals can be analyzed by using wavelet transform. There are different kinds of purposes of using WT in ECG signals. These include de-noising, heartbeat detection, and feature extraction.

Principal component analysis (PCA) is a method that reduces the dimension of the dataset. At the same time, it minimizes information loss. To maximize the variance successively it can make a new uncorrelated variable. To extract the important one from a large pool PCA is a technique that is used to reduce the number of variables from data. To retain as much information as possible reduces the dimension of data. PCA can also utilize the accuracy of the classification model.

RR intervals are a function of the basic properties of the sinus node. It represents the duration between the next heartbeat. This duration is used to calculate the heart rate. For obtaining the efficient features from the heartbeat signal input, RR interval features need to be extracted. These are defined as dynamic features. The RR interval features are divides into four types which are previous RR, post RR, local RR, and average RR. The variation of RR intervals is responsible for detect different kinds of Arrhythmia.

In the work of Joshi [12], a support vector machine model is proposed and achieved 86.40% of overall accuracy into five classes. For the classification and regression analysis, SVM analyzes data. They can categorize new text. The SVM model extracts the labeled training data for each class of beats. To draw a straight line between two classes, the linear SVM classifier works. Support vectors represent the data points that are closer to the hyperplane and influence the position. To enlarge the margin of the classifier, SVM is used. The aspects of the hyperplane will change by deleting the support vectors from the signals. These points help to build the SVM.

A probabilistic neural network is used and achieved 92.70% overall accuracy into five classes in Josh et al. [13]. Wavelet is used for preprocessing and feature extraction. And PNN is used for the classification method.

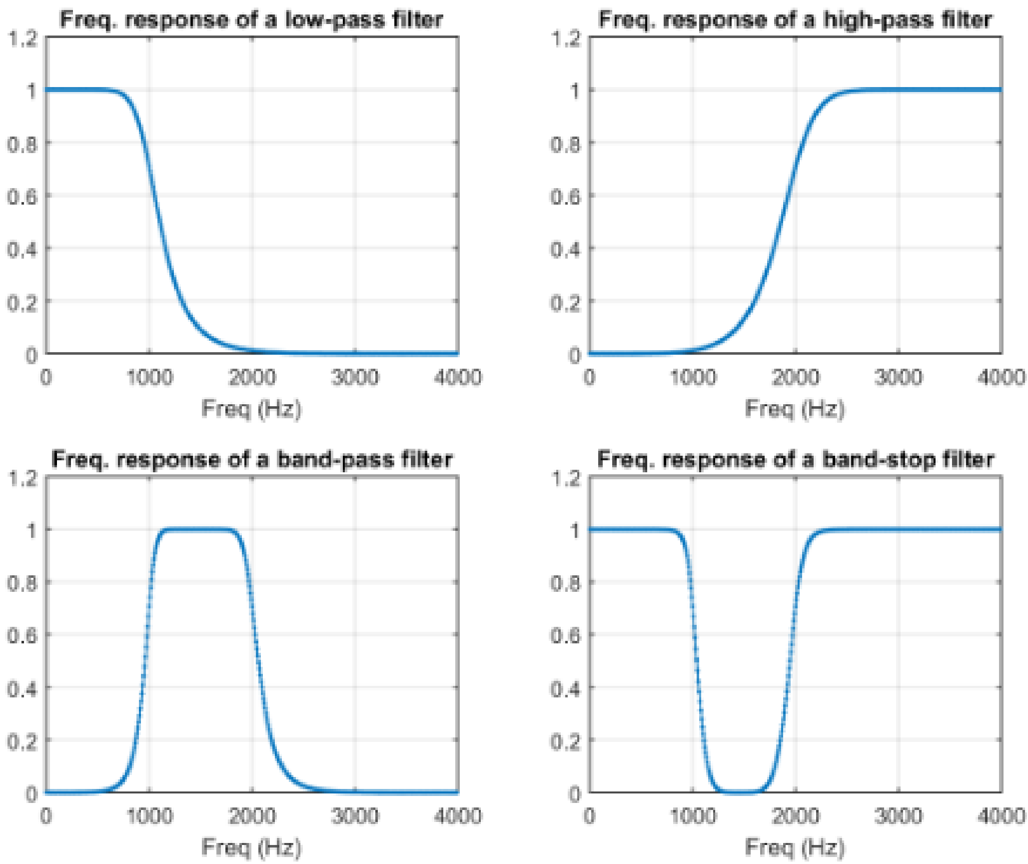

Figure 3.5: The arch of filtering of Martis

The PNN architecture represents the backpropagation network with an activation function derived from statistical data. It is a classifier that maps input patterns in several class levels. There are four types of layer in PNN; an input layer, a summation layer, a pattern layer, and a decision layer. Bandpass filters are used to remove the data by the elimination of factitious noise. There are many kinds of filter which remove noise differently. Bandpass Butterworth is the easiest approach. This filter is used to clean up any signals within a specific frequency range. It also rejects the frequencies which are outside range. The low pass filter is responsible for isolating the signals which have frequencies higher than the cutoff frequency. Using bandpass filters is responsible for remove the data, particularly where large amounts of gain have been added. And typically where the survey took place over lossy or uneven ground.

A neural network model has achieved an overall accuracy of 92.70% into five classes in the work of Zubiar [14]. In this work, a bandpass filter is used as preprocessing method, CNN as feature extraction, and softmax as a classification method. The softmax function is called a normalized exponential function that takes as input a vector z of K real vectors. It normalizes the K vectors into a probability distribution. It consisting of K probabilities proportional to the exponentials of the input numbers. The softmax function is used to determine the losses. It is needed when training a data set. The efficient use-cases of softmax regression are in discriminative models such as Cross-Entropy.

A neural network(NN) and support vector machine approach are achieved an overall accuracy of 93.00% for arrhythmia classification into five classes in the work of Martis [15]. In this work, wavelet is used as preprocessing method. Pantompkins and principal component analysis are responsible for feature extraction. Also, neural networks and support vector machines are responsible for classification.

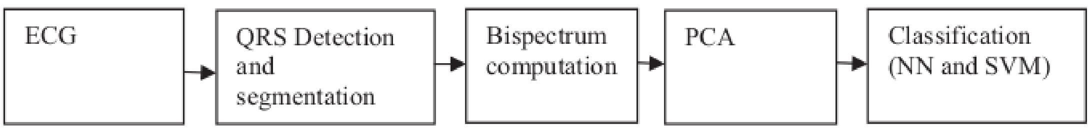

Figure 3.6: The flow diagram of model building of Martis

Firstly, QRS detection is completed. Then the segmentation processes have been completed. For detecting the QRS Pantompkins algorithm is used. This algorithm includes the average filtering of signals and also the operations of the threshold. It removes the high-frequency noise. And the threshold operation contains the rectangular pulses for detecting the R peak.

For the detection of abnormal heart activity, the Bispectrum Computation method is used in Martis [15]. It filters the higher-order functions of signals to detect various kinds of arrhythmia [fig 3.7]. The neural network method is used which includes the input layer, two hidden layers, and also the output layer. A gradient descent algorithm is responsible for updating each layer of functional features [fig 3.8].

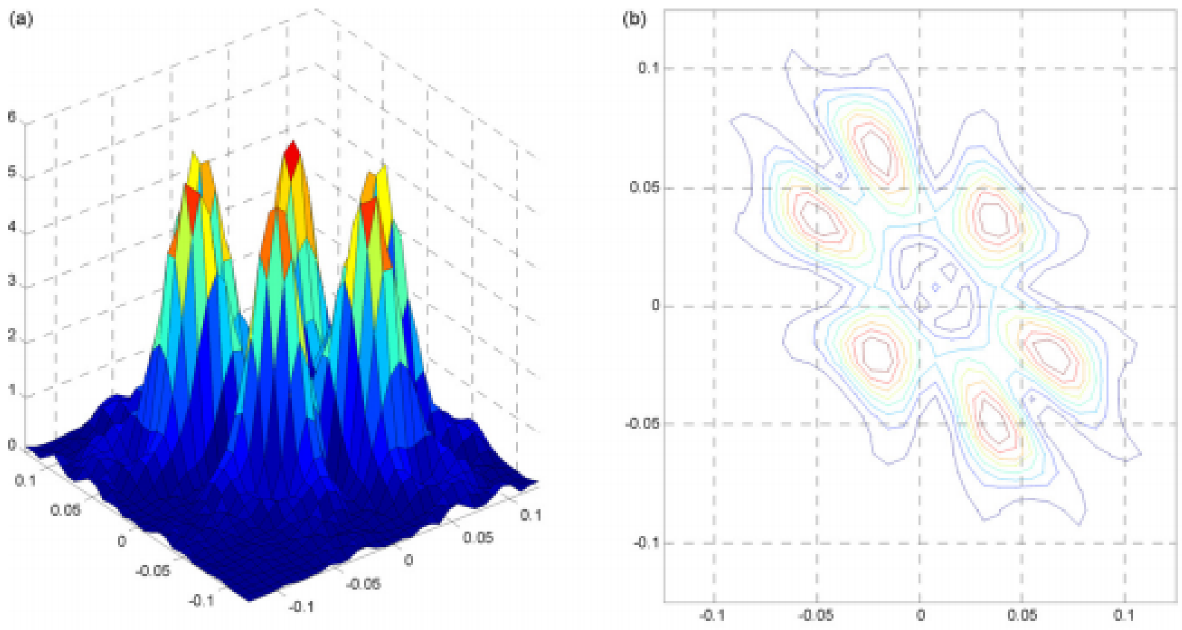

Figure 3.7: Normal beat: (a) Bispectrum graph, (b) Contour of (a) graph

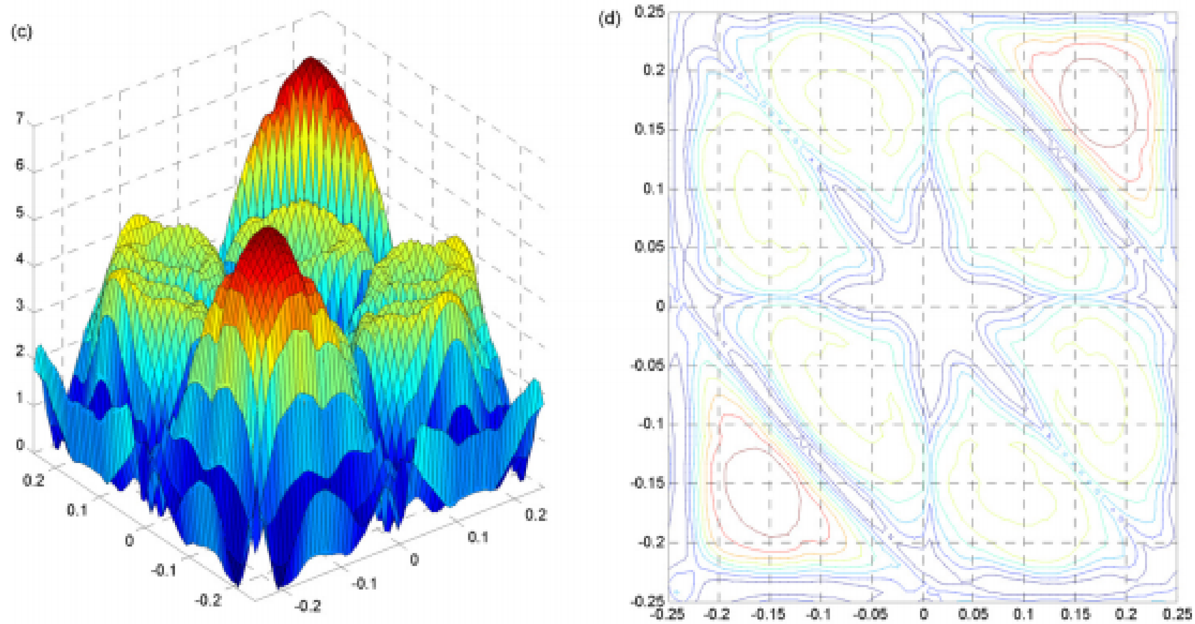

Figure 3.8: Normal beat: (c) Bicoherence graph, (d) Contour of (c) graph

Principle component analysis is used for the reduction technique in Martis [15]. It indicates the highest variabilities of each layer of signals [fig 3.9]. In this work, the first 12 components are responsible for the feeding the classifier and the other 12 components are responsible for represents the ECG beats [fig 3.10]. Support vector machine is used as the non-linear network. It minimizes the structural risk and classifies the unseen data efficiently. Also, it has the higher ability to generalize the data.

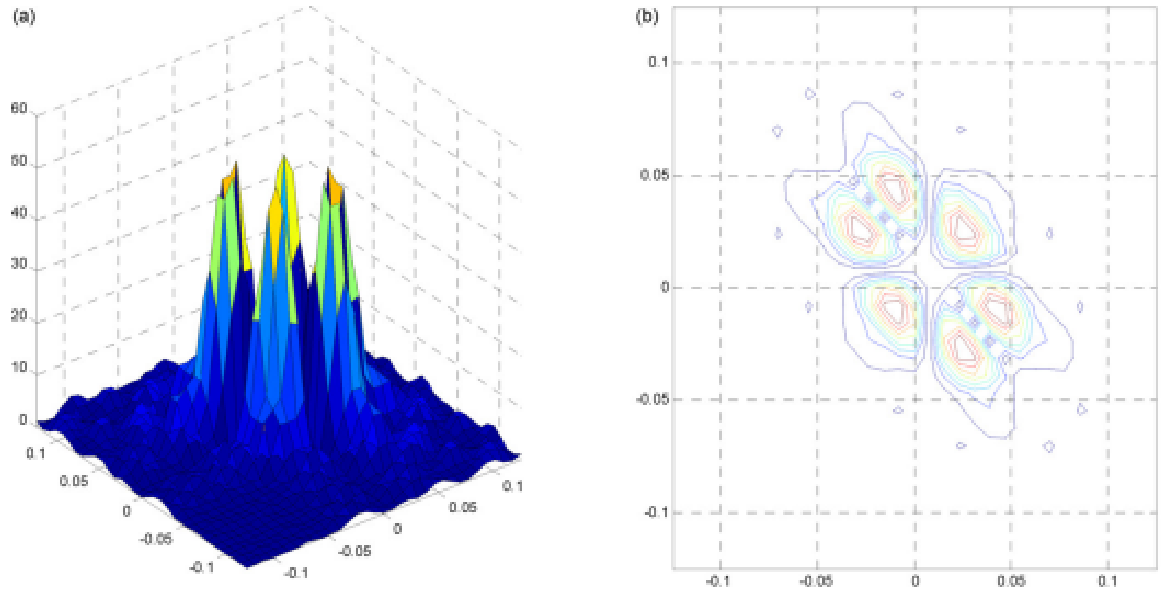

Figure 3.9: RBBB beat: (a) Bispectrum, (b) Contour of (a)

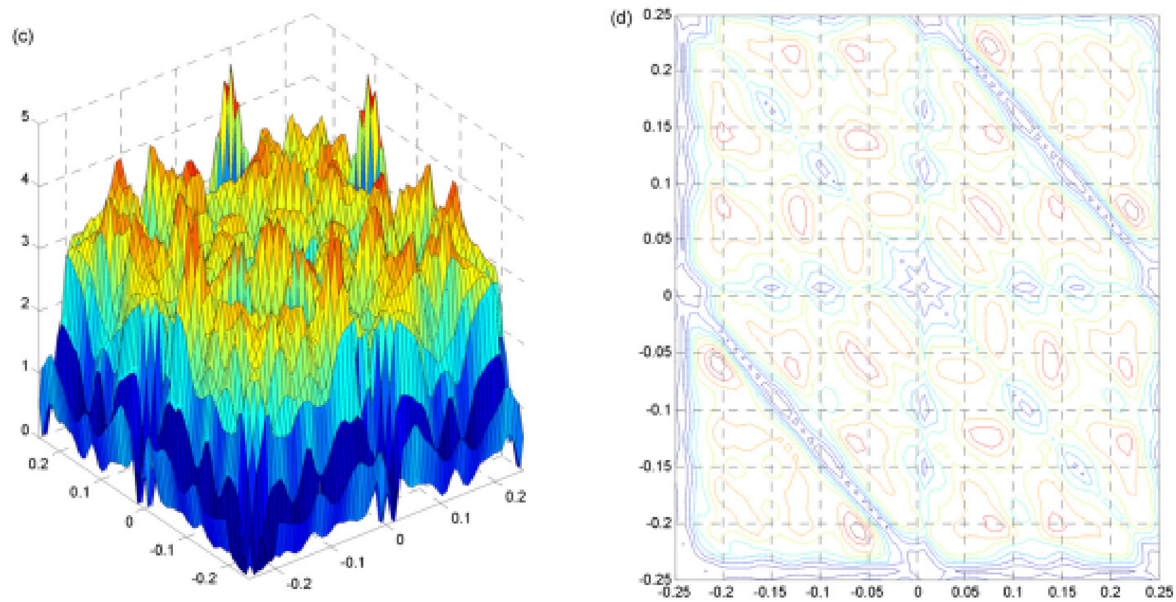

Figure 3.10: RBBB beat: (c) Bicoherence, (d) Contour of (c)

An additive neural network is achieved an overall accuracy of 94.00% for arrhythmia classification into five classes in Ismaiel [16]. In this work, digital filters are used as preprocessing method. Discrete wavelet is used as a feature extraction method. Also, an additive neural network is responsible for classification.

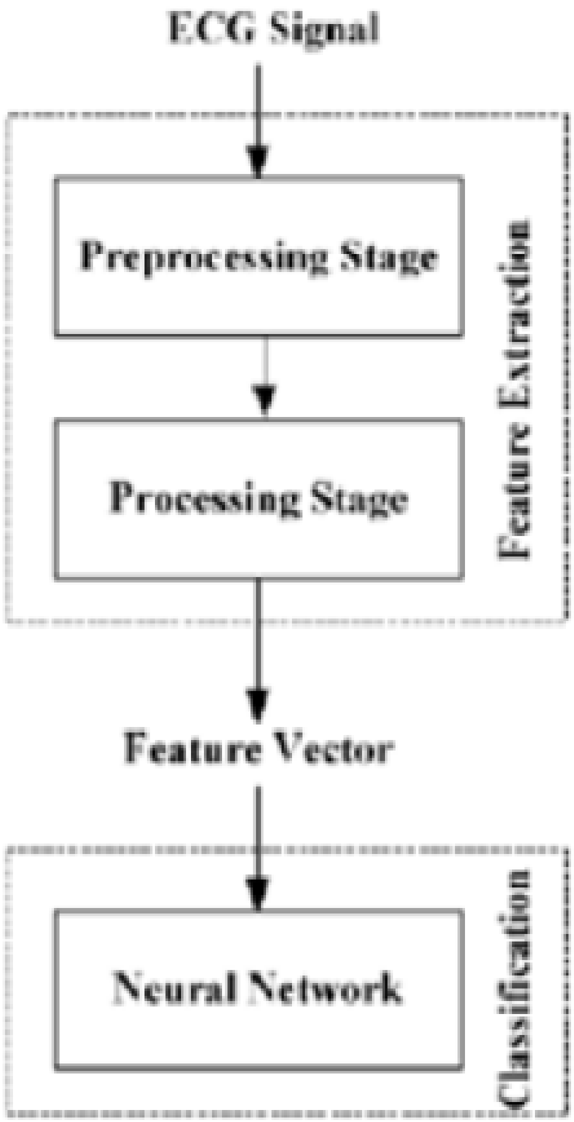

Figure 3.11: The flow chart of model building of Ismaiel

For accurate analysis of the signal, noise is needed to be clean up. Adaptive filtering has the self-learning ability to clean up the noise from the ECG signal. It takes two input signals and merges the signals with noise and desired signal. The MIT-BIH database is imported into Matlab for the processing of adaptive filtering.

Wavelet transform is responsible for parameter extraction. Multilevel Discrete Wavelet Transform is responsible for parameter extraction.

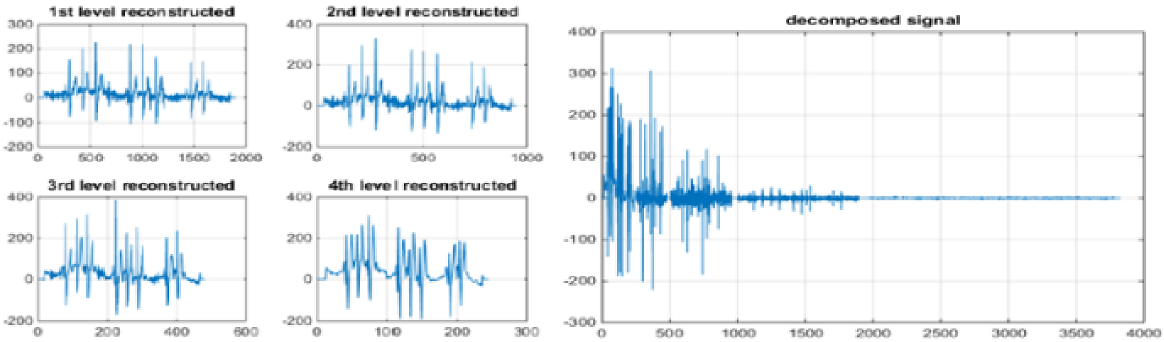

Figure 3.12: Decomposed signal in Ismaiel

A neural network is used for arrhythmia classification in Ismaiel [16]. It contains backpropagation approaches with an input layer, hidden layer, and output layer. Sigmoid is used as an activation function. The classification is used 10 neurons in the hidden layer.

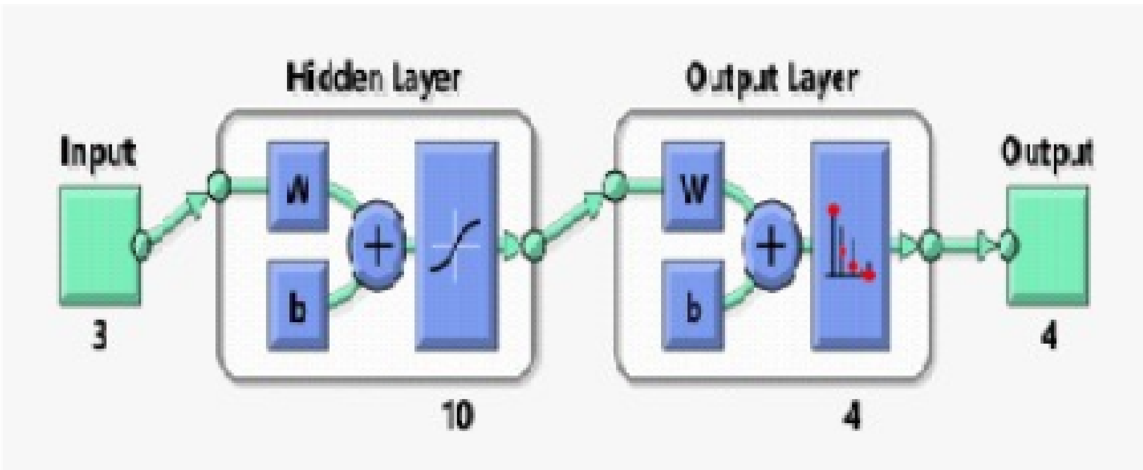

Figure 3.13: Neural network architecture of Ismaiel

A Neural Network-based algorithm in Jiang [17] is achieved 96.66% overall accuracy in five classes. As a preprocessing technique bandpass filter is used. Also, the Hermite transform is used as a feature extraction method. A support vector machine approach is achieved an overall accuracy of 97.20%. As preprocessing techniques bandpass filter and continuous wavelet transform are used for feature extraction. The arrhythmia classification is into five classes in Zadeh [18].

CHAPTER 4

PROPOSED METHOD

This chapter consists of the proposed methodology of our work. We proposed a method for ECG Arrhythmia Classification Using 1-D Convolution Neural Network Leveraging the Resampling Technique and Gaussian Mixture Model into five classes.

4.1 Preprocessing

In this thesis, several preprocessing techniques are used for balancing the dataset and generalizing the train set with Gaussian distribution [Fig 3.3]. ECG signals are combined with the functional channel of the node which is extended with electrical attachment. They mislead the aspect of ECG and peak critical segment that causes the abnormal heartbeats. Generalization is a familiar process used to increase the aspect of ECG signal. Accordingly, the preprocessed data is used for input data in our CNN model to obtain the classification of arrhythmia. Resampling is the technique that consists of sampling repeated samples from our MIT-BIH original data samples. It defines the nonparametric method of statistical inference.

4.1.1 Resample

It is an efficient technique for balancing highly imbalanced datasets. It contains the set of methods where samples are repeated from a given sample or classify the precision with efficient statistics. The duplication of ECG examples from the dataset of minority classes is called oversampling [fig 4.1]. It selects the ECG examples from the minority class, then adds the examples to the training dataset by the following replacement. Also, the deletion of ECG examples from the dataset of majority classes is called undersampling [fig 4.2]. It takes the majority of examples of ECG until the balanced dataset distribution is achieved.

In this thesis, we combined the two techniques, and resample technique is chosen more than the class weights with several 109446 samples which include 20,000 samples [Fig 4.4] of Normal beats (N) Supraventricular ectopic beats (S), Ventricular ectopic beats(V), Fusion beats (F) and Unknown beats (Q).

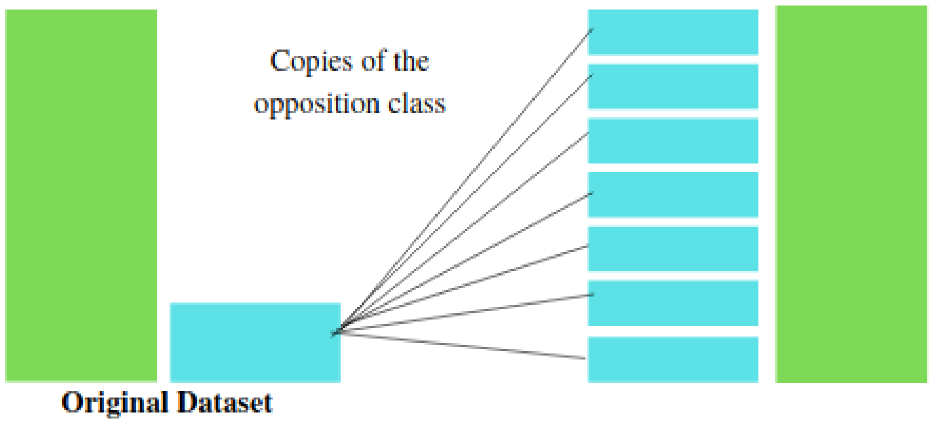

Figure 4.1: Oversampling

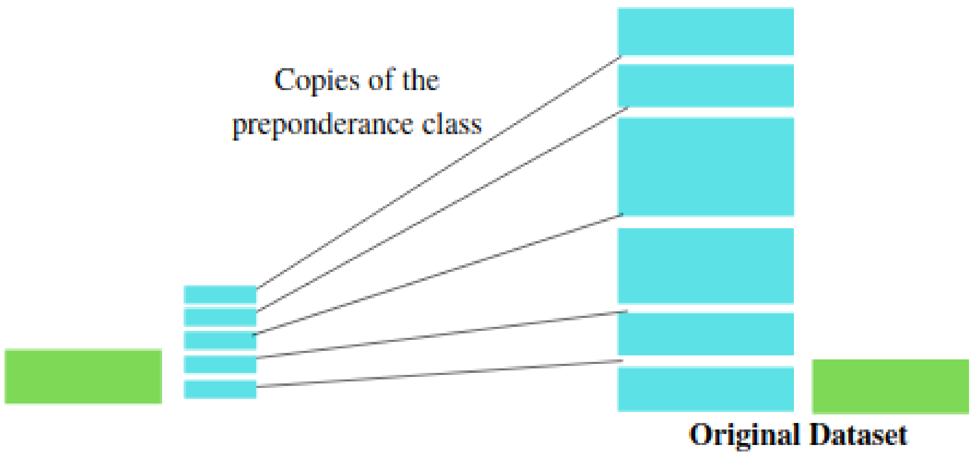

Figure 4.2: Undersampling

4.1.2 White Gaussian Noise

Gaussian noise is one kind of analytical noise that is estimated from the probability density function (PDF) which corresponds to the normal distribution. White Gaussian Noise is defined as a special case where the values of signals at any pair are uniformly distributed. In our dataset, the enlarged portion of the signal includes the interval of the R wave peak. Signal-to-Noise (SNR) is a measurement that is the ratio between the flattening of the desired noise signal and the flatten of background noise. The SNR value can indicate how much noise is present in the signal. The higher the SNR value the better to detect the desired peak for classification.

$$SNR = \frac{P_{signal}}{P_{noise}} \tag{4.1}$$

where P is the average power function. If the signal is a constant(s) the signal-to-noise(SNR) ratio of random noise turn into:

$$SNR = \frac{S^2}{E(|N|)^2} \tag{4.2}$$

where E refers to the expected value. In our paper, Gaussian White Noise is added for generalization of the train data which predetermined upper and lower hop of the co-ordinate axis, accomplishing each ECG beat is based in the right place as shown [Fig 4.3].

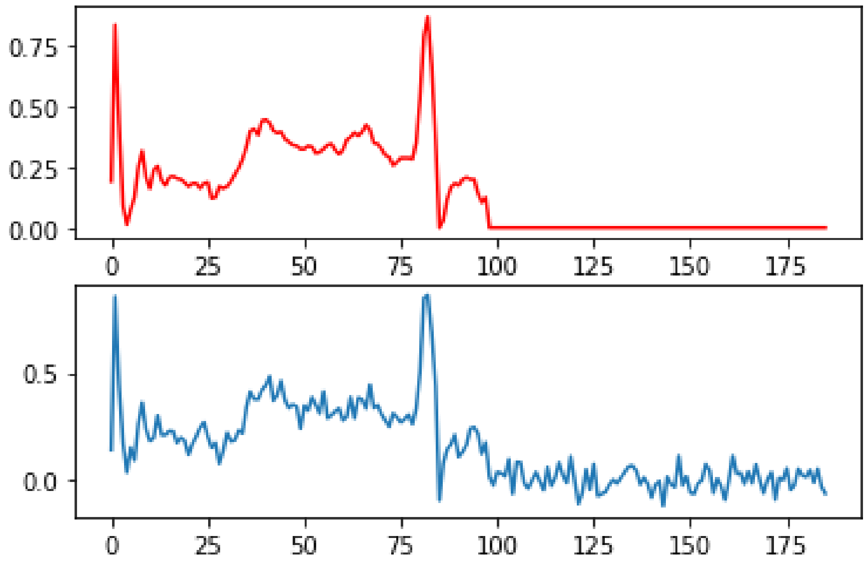

Figure 4.3: Added gaussian noise

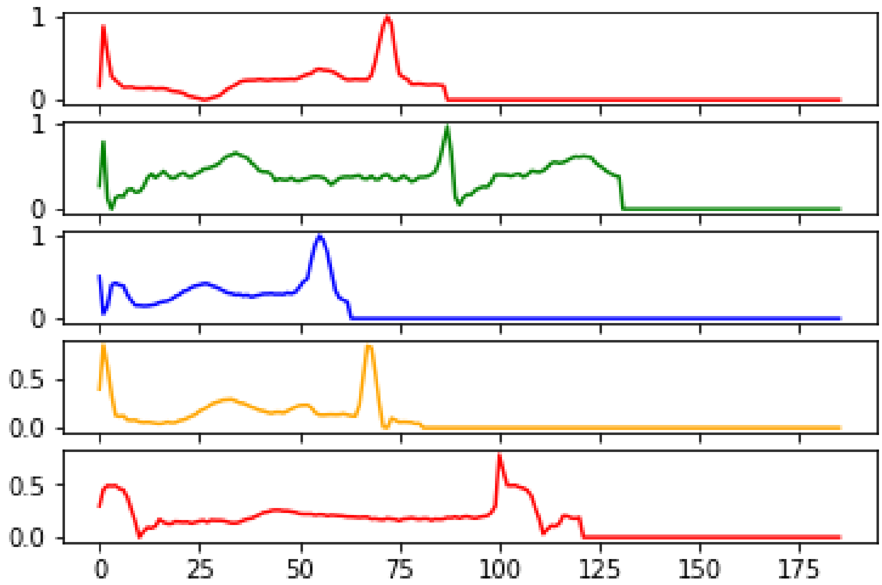

Figure 4.4: Resampled samples

4.2 Experiments

4.2.1 Dataset

The MIT-BIH Arrhythmia Dataset contains the collection of ECG recordings of twice channel in 48 half-hours. It is developed by the BIH Arrhythmia Laboratory. From the set of 24-hour ECG recordings, twenty-three recordings are picked continuously. From a combination of the samples of inpatients and outpatients, it is achieved. For including less common but clinically significant arrhythmias the rest of the twenty-five recordings are gathered from the same set. Dataset link:https://physionet.org/content/mitdb/1.0.0/

Table 4.1: relationship between AAMI and MITBIH heartbeats

AAMI Classes	**MIT-BIH Heartbeats**
Normal beats (N)	left bundle branch block beats, right bundle branch block beats, nodal escape beats, atrial escape beats
Supra Ventricular ectopic beats (S)	aberrated atrial premature beats,supraventricular premature beats,atrial premature beats
Ventricular ectopic beats (V)	ventricular flutter wave, ventricular escape beats, premature ventricular contraction
Fusion beats (F)	ventricular beats, normal beats
Unknown beats (Q)	paced beats, unclassifiable beats, fusion paced beats, ectopic beats

In this thesis, for grouping, the heartbeats into five different classes the AAMI [Table 4.1] has proposed which contains 109,446 ECG samples.

4.2.2 1D Convolution Neural Network

A Convolution Neural Network is widely familiar for recognizing visual patterns from the data. Numerous parts of every convolution layer can extract multiple types of deep features. The filter weights in the convolution layer work similarly to a visualization system which might not be meaningful for humans but those can be highly effective for classification. In the pooling layer, the pooling filters have decreased the adversity of training parameters and separate the data aspect which maintains the balance of training data with efficient features of parameters. This collection of parameters is used for ECG signal processing [5]. Accordingly, convolution neural networks have a convenient role in ECG signal processing [6]. Because of having 1D data, a convolution network of 1D is taken in this ECG arrhythmia classification.

Convolution Neural Network (CNN) is consisted of feature extraction and classification. The segment of feature extraction is compulsory for peaking efficient features from the raw signals. Architecture and the hyper-parameters are discussed in the following [fig 4.5].

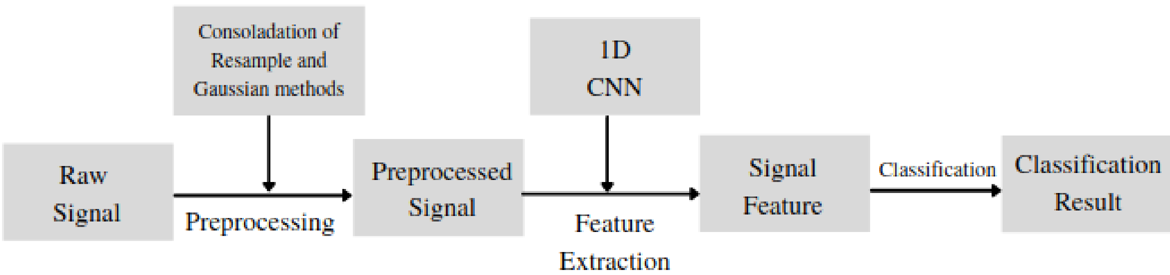

Figure 4.5: Proposed Model building flow chart

The segment of feature extraction contains the convolution layer and also the down-sampling layer. The convolution layer (Conv-layer) [Fig 4.6] is observed because of enlarging the parameters of raw signals and separating the noise. Convolution action is held for feature vectors of the bottom layer and kernel of the common layer. Accordingly, the result of convolution calculations is given by the activation function.

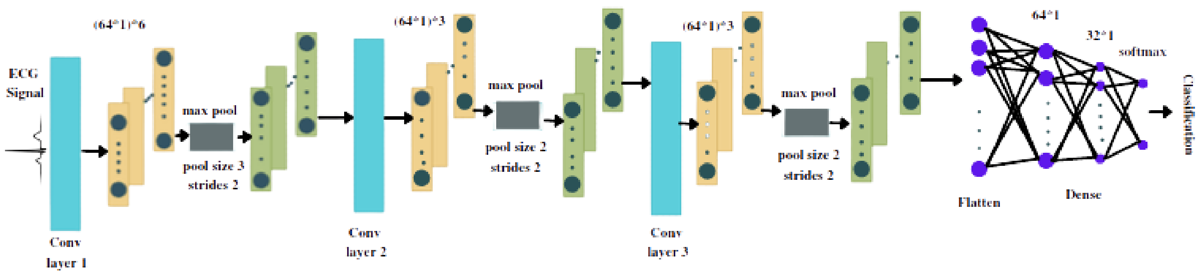

Figure 4.6: Proposed model with 1D Convolution Neural Network

Because of the non-linear feature mapping, the activation function is needed before the output of the convolution layer.

In our thesis, a ten-layer 1D CNN model is showed according to the study of train pre-processed data and getting the classification output. A CNN can extract effective features at the time of the training process. The proposed 1D CNN is constructed using three convolution layers, three pooling, one flattened, two dense, and a fully connected layer [Fig 3.6]. For the original ECG signal, the input size of the first convolution layer consists of filter size 64 with kernel size 6. The rectifier linear unit (ReLU) function is needed for activation function. A pooling layer of size 3 with the size of stride of 2 is added after the previous layer. The second convolution layer consists of filters with a size of 64 and a kernel size of 3. Also, the third convolution layer is set as the same. Therefore, the second and third pooling layers have a pool of size 2 with strides size 2 [Fig 4.6]. The dropout layer indicates dropping out units in the neural network randomly. For preventing the over-fitting problem dropout rate of 0.2 is selected. Decisively, the output of classification is obtained with the convolution and pooling layers which are appointed to the fully connected layer.

Layers	Type	Filter Size	Strides	kernel	output shape	Parameters
Layer 1	Conv1d	6 * 6	-	64	(None, 181, 64)	448
Layer 2	Pooling	3 * 3	2 * 2	-	(None, 91, 64)	0
Layer 3	Conv1d	3 * 3	-	64	(None, 89, 64)	12352
Layer 4	Pooling	2 * 2	2 * 2	-	(None, 45, 64)	0
Layer 5	Conv1d	3 * 3	-	64	(None, 43, 64)	12352
Layer 6	Pooling	2 * 2	2 * 2	-	(None, 22, 64)	0
Layer 7	Flatten	2 * 2	2 * 2	-	(None, 1408)	0
Layer 8	Dense	-	-	64	(None, 64)	90176
Layer 9	Dense	-	-	32	(None, 32)	2080
Layer 10	Output	-	-	5	(None, 5)	165

Figure 4.7: Proposed 1D CNN model architecture Layers

Batch normalization is used in the CNN network. Batch normalization is an optimization method developed by Google [9] for data standardization and normalization where a group of data is referred to as batch. Therefore, The input and output data of the medium network layer which is composed of the inner neurons, and the difference may be obtained by applying batch normalization [10]. In this paper, the number of total parameters is 118,341 and the total trainable parameter is 117,957 [Fig 4.7].

CHAPTER 5

EXPERIMENTAL SETUP AND RESULT ANALYSIS

5.1 Experimental setup

The proposed ECG arrhythmia classification using 1D convolution neural network is implemented with python. Also the open-source library TensorFlow and neural network library Keras is used which is introduced by Google for deep learning [11]. Keras is the high-level API that is built on the top of TensorFlow. The experiment setup consists of a second-generation HP Compaq notebook server with 4GB internal RAM, 320 GB internal hard drive, and no external hard drive included. The processor consisted of Intel(R) Celeron(R) CPU B815@1.6. The 1D ECG signal is divided into 80% data for training and 20% of data for testing. This low configuration device is used to demonstrate that this model is very lightweight and can be effectively implemented in an edge device at the user end. One of our objectives is to make the model as light as possible while maintaining the highest possible accuracy.

5.2 Classification Results and Discussion

5.2.1 Results

The proposed model is trained using a set of hyper-parameters which is selected empirically. The first and second convolution layer is set as kernel size 64. The sampling of three pooling layers is set as stride size 3,2,2 which is maximizing the operation with the number of iterations 150 times. Adam optimizer is used to accelerate the gradient descent process. A softmax function is used in the final layer. As a loss function categorical-cross-entropy is used. Finally, the test result achieves an accuracy of 98.25%.

The first model achieves an overall accuracy is 97.64% on the test set [Table 5.1]. In this model, the dropout rate is set to 0.2 and 0.3 for the first three Conv layers and first dense layer respectively. The convolution layers kernel and the filter size are mentioned the [fig. 4.6]. Also, the pool and strides size is set as [fig 4.6]. The accuracy of all the classes is as follows. Normal beats (N) 98.00%, Supraventricular ectopic beats(S) 85.00%, Ventricular ectopic beats(V) 96.00%, Fusion beats(F) 80.00% and Unknown beats(Q) 99.00% with 200 iterations.

Table 5.1: Confusion Matrix:Accuracy 97.64%

Class	**N**	**S**	**V**	**F**	**Q**
N	0.98	0.01	0.00	0.00	0.00
S	0.12	0.85	0.02	0.01	0.00
V	0.02	0.01	0.96	0.01	0.00
F	0.06	0.01	0.14	0.80	0.00
Q	0.01	0.00	0.00	0.00	0.99

In the second model, the test result finds that the overall accuracy is 98.00% [Table 5.2] where the dropout is 0.2 is kept in three pooling layers and also in the two dense layers. The kernel and filter size is kept the same as before. The accuracy of each class is found as Normal beats (N) 99.00%, Supraventricular ectopic beats(S) 84.00%, Ventricular ectopic beats(V) 94.00%, Fusion beats(F) 81.00% and Unknown beats(Q) 99.00% with the number of 150 iterations.

Table 5.2: Confusion Matrix:Accuracy 98.00%

Class	**N**	**S**	**V**	**F**	**Q**
N	0.99	0.01	0.00	0.00	0.00
S	0.14	0.84	0.01	0.00	0.01
V	0.03	0.00	0.94	0.02	0.00
F	0.07	0.01	0.10	0.81	0.00
Q	0.01	0.00	0.00	0.00	0.99

In the third model, the test result is achieved as overall accuracy is 98.02% [Table 5.3] where the dropout of 0.2 is used in three pooling layers and the first dense layers. The kernel and filter size is set as the same. The accuracy of each class is found as Normal beats (N) 99.00%, Supraventricular ectopic beats(S) 85.00%, Ventricular ectopic beats(V) 96.00%, Fusion beats(F) 84.00% and Unknown beats(Q) 99.00% with the number of 150 iterations.

Table 5.3: confusion matrix:accuracy 98.02%

Class	**N**	**S**	**V**	**F**	**Q**
N	0.99	0.01	0.00	0.00	0.00
S	0.12	0.85	0.02	0.01	0.00
V	0.02	0.00	0.96	0.02	0.00
F	0.06	0.01	0.09	0.84	0.00
Q	0.00	0.00	0.00	0.00	0.99

In the final model, the test result is achieved as overall accuracy is 98.25% [fig 5.1] according to the [fig 4.6] proposed architecture where the dropout 0.2 is used in three pooling layers. The kernel and filter size is set as the same. The accuracy of each class is found as Normal beats (N) 98.00%, Supraventricular ectopic beats(S) 83.00%, Ventricular ectopic beats(V) 95.00%, Fusion beats(F) 86.00% and Unknown beats(Q) 99.00% with the number of 150 iterations.

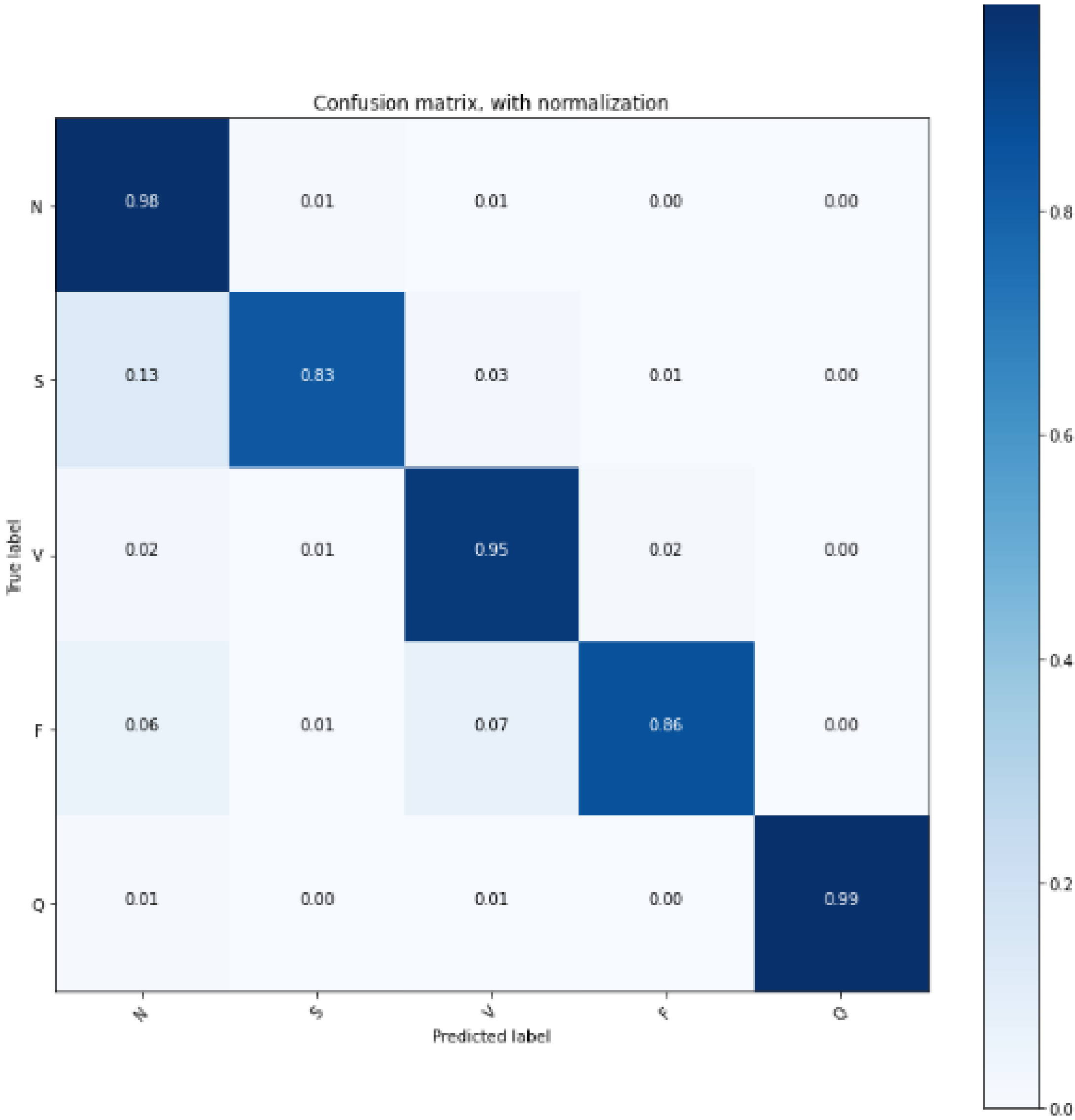

Figure 5.1: Confusion Matrix: Accuracy 98.25%

5.2.2 Discussion

The comparison between our model with other past existing models is given [Table 5.4]. The proposed method method obtains the highest accuracy of arrhythmia classification compared to the previously published classification results into five classes using one dimension convolution neural network. The proposed method represents a comparatively lightweight CNN model that is suitable for one-dimension ECG signals. The dataset is trained with a different number of epochs and values of hyper-parameters. Each model demonstrates a high classification accuracy comparing the previous model [Table 5.4]. The confusion matrix is represented in the results section in detail.

However, we have faced a problem while tracking the wave R peak for getting the desired SNR (signal-to-noise). Thus we have used the White Gaussian Noise as a solution. Also encountered another problem with the generalization of the dataset. This problem is solved by utilizing the resample techniques functionalities. Constructing a lightweight method with 1D Convolution Neural Network while keeping a high accuracy is also challenging. The previous deep learning models are very complex and take a good amount of time to train and test. To make this model lightweight and effective, we created and tested more than 50 CNN models. Surprisingly, several lightweight models achieved very high accuracy and even some model beat the previous existing model accuracy.

Table 5.4: Comparison with Existing Algorithms

Paper	**Class**	**Preprocessing**	**Feature Extraction**	**Classification**	**Accuracy**
Dan et al. [19]	N,L,R,A,V	wavelet Combination	1D-CNN	softmax	97.50%
Model 01	N,S,V,F,Q	resample and gaussian mixture	1D-CNN	Softmax	97.64%
Model 02	N,S,V,F,Q	resample and gaussian mixture	1D-CNN	Softmax	98.00%
Model 03	N,S,V,F,Q	resample and gaussian mixture	1D-CNN	Softmax	98.02%
Proposed	**N,S,V,F,Q**	**resample and gaussian mixture**	**1D-CNN**	**Softmax**	**98.25%**

The performances of selected models are shown in the [Table 5.1, 5.2, 5.3] and [fig 5.1]. Finally, our model which is given in this work is better and efficient than all of the previous algorithms that are proposed with a one-dimension convolution neural network into five classes. In this paper, the deep feature extraction is performed by the one-dimensional convolution neural network and the classification is done at the output layer with a soft-max function. Our proposed model has the highest accuracy of 98.25% which is a classification result of all five classes of arrhythmia.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

CNN can be effectively used to analyze the ECG signals to determine cardiovascular diseases. It is a very active and essential research area. The convolution neural network determines adequate features and exhaustive classification of different kinds of arrhythmia. In this paper, the classification is obtained by two approaches. They are the Resampling Technique with Gaussian Mixture Model and 1D CNN. In our paper, Preprocessing technique is worked for the generalization of balance data and track the R peak with the desired SNR. By processing the several epochs the highest accuracy is achieved. A satisfactory classification overall accuracy of 98.25% is achieved with an f1 score of 98.24%, positive predictive value(precision) 97.58%, and recall 96.79% by comparing with previous work. We are hopeful that the proposed model architecture will help medical experts diagnose cardiovascular diseases by giving the efficient classification of ECG signals which observes less computational power.

6.2 Future Work

For future work, the two-dimension peak of a signal can be used for arrhythmia classification into more classes. Also, with the growth of mobile applications, a prenotification system can be developed to notify arrhythmias probability.

BIBLIOGRAPHY

[1] P. -Y. Hsu and C. -K. Cheng, "Arrhythmia Classification using Deep Learning and Machine Learning with Features Extracted from Waveform-based Signal Processing," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), Montreal, QC, Canada, 2020, pp. 292-295, doi: 10.1109/EMBC44109.2020.9176679.

[2] M. Shahin, E. Oo and B. Ahmed, "Adversarial Multi-Task Learning for Robust End-to-End ECG-based Heartbeat Classification," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), Montreal, QC, Canada, 2020, pp. 341-344, doi: 10.1109/EMBC44109.2020.9175640.

[3] X. Xu and H. Liu, "ECG Heartbeat Classification Using Convolutional Neural Networks," in IEEE Access, vol. 8, pp. 8614-8619, 2020, doi: 10.1109/ACCESS.2020.2964749.

[4] Nurmaini, Siti; Darmawahyuni, Annisa; Sakti Mukti, Akhmad N.; Rachmatullah, Muhammad N.; Firdaus, Firdaus; Tutuko, Bambang. 2020. "Deep Learning-Based Stacked Denoising and Autoencoder for ECG Heartbeat Classification" Electronics 9, no. 1: 135. https://doi.org/10.3390/electronics9010135

[5] Hong S, You T, Kwak S, et al. Online Tracking by Learning Discriminative Saliency Map with Convolutional Neural Network[J]. Computer ence, 2015:597-606.A

[6] Shen, Yelong, He, Xiaodong, Gao, Jianfeng. Learning Semantic Representations using Convolutional Neural Network for Web Search[J]. proc www, 2014:373-374.A

[7] Kiranyaz S, Ince T, Hamila R, Gabbouj M. Convolutional Neural Networks for patient-specific ECG classification. Annu Int Conf IEEE Eng Med Biol Soc. 2015;2015:2608-11. doi: 10.1109/EMBC.2015.7318926. PMID: 26736826.

[8] Dosovitskiy A, Springenberg J T, Brox T. Learning to Generate Chairs with Convolutional Neural Networks[C]// 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015.

[9] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[C]// International Conference on International Conference on Machine Learning.

[10] Simon M, Rodner E, Denzler J. ImageNet pre-trained models with batch normalization[J]. 2016.

[11] Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv 2016, arXiv:1603.04467

[12] N. P. Joshi, P. S. Topannavar, Support vector machine based heartbeat classification. In Proc. of 4th IRF Int. Conf , pp. 140-144. 2014.

[13] J. A. DSP-based arrhythmia classification using wavelet transform and probabilistic neural network. Biomedical Signal Processing and Control, vol. 32, pp. 44-56, 2017

[14] M. Zubair, J. Kim, and C. Yoon. ‚ÄúAn Automated ECG Beat Classification System Using Convolutional Neural Networks‚Äù. In IT Convergence and Security (ICITCS), 2016 6th International Conference on , pp. 1-5. 2016. IEEE

[15] R. J. Martis, U. R. Acharya, K. M. Mandana, et al, ‚ÄúCardiac decision making using higher order spectra‚Äù. Biomedical Signal Processing and Control, vol. 8, no. 2, pp.193-203, 2013.

[16] smaiel, Fatima Osman Mohamed. "Classification of Cardiac Arrhythmias Based on Wavelet Transform and Neural Networks." Sudan University of Science Technology. 2015. IEEE

[17] W. Jiang and S. G. Kong, ‚ÄúBlock-based neural networks for personalized ECG signal classification,‚Äù IEEE Transactions on Neural Networks, vol. 18, no. 6, pp. 1750‚Äì1761, 2007. IEEE

[18] Zadeh A E, Khazaee A. ‚ÄúHigh Efficient System for Automatic Classification of the Electrocardiogram Beats‚Äù. Annals of Biomedical Engineering, vol. 39, no. 3, pp. 996-1011. 2011. IEEE

[19] D. Li, J. Zhang, Q. Zhang and X. Wei, "Classification of ECG signals based on 1D convolution neural network," 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, 2017, pp. 1-6, doi: 10.1109/HealthCom.2017.8210784.

[20] M. Kachuee, S. Fazeli and M. Sarrafzadeh, "ECG Heartbeat Classification: A Deep Transferable Representation," 2018 IEEE International Conference on Healthcare Informatics (ICHI), New York, NY, 2018, pp. 443-444, doi: 10.1109/ICHI.2018.00092.

[21] Chaur-Heh Hsieh, Yan-Shuo Li, Bor-Jiunn Hwang, Ching-Hua Hsiao Sensors (Basel) 2020 Apr; 20(7): 2136. Published online 2020 Apr 10. doi: 10.3390/s20072136 PMCID: PMC7180882

[22] Abdalla, F.Y.O., Wu, L., Ullah, H. et al. Deep convolutional neural network application to classify the ECG arrhythmia. SIViP 14, 1431‚Äì1439 (2020). https://doi.org/10.1007/s11760-020-01688-2

[23] Coast, D.A.; Stern, R.M.M.; Cano, G.G.; Briller, S.A. An approach to cardiac arrhythmia analysis using hidden markov models. IEEE Trans. Biomed. Eng. 1990, 37, 826‚Äì836.

[24] Mustaqeem, A.; Anwar, S.M.; Majid, M. Multiclass classification of cardiac arrhythmia using improved feature selection and SVM invariants. Comput. Math. Methods Med. 2018.

[25] Inan, O.T.; Giovangrandi, L.; Kovacs, G.T. Robust neural-network-based classification of premature ventricular contractions using wavelet transform and timing interval features. IEEE Trans. Biomed. Eng. 2006, 53, 2507‚Äì2515.

[26] Willems, J.L.; Lesaffre, E. Comparison of multigroup logistic and linear discriminant ecg and vcg classification. J. Electrocardiol. 1987, 20, 83‚Äì92.

[27] Trahanias, P.; Skordalakis, E. Syntactic pattern recognition of the ECG. IEEE Trans. Pattern Anal. Mach. Intell. 1990, 12, 648‚Äì657.

[28] Hu, Y.H.; Palreddy, S.; Tompkins, W.J. A patient-adaptable ECG beat classifier using a mixture of experts approach. IEEE Trans. Biomed. Eng. 1997, 44, 891‚Äì900.

[29] Dehan, L.; Guanggui, X.U.; Yuhua, Z.; Hosseini, H.G. Novel ECG diagnosis model based on multi-stage artificial neural networks. Chin. J. Sci. Instrum. 2008, 29, 27.

[30] Salvatore, C.; Cerasa, A.; Battista, P.; Gilardi, M.C.; Quattrone, A.; Castiglioni, I. Magnetic resonance imaging biomarkers for the early diagnosis of Alzheimer‚Äôs disease: A machine learning approach. Front. Neurosci. 2015, 9, 307.

[31] Kiranyaz, S.; Ince, T.; Gabbouj, M. Real-time patient-specific ECG classification by 1-D convolutional neural networks. IEEE Trans. Biomed. Eng. 2015, 63, 664‚Äì675.

[32] Rajpurkar, P.; Hannun, A.Y.; Haghpanahi, M.; Bourn, C.; Ng, A.Y. Cardiologist-level arrhythmia detection with convolutional neural networks. arXiv 2017, arXiv:1707.01836.

[33] Acharya, U.R.; Oh, S.L.; Hagiwara, Y.; Tan, J.H.; Adam, M.; Gertych, A.; San Tan, R. A deep convolutional neural network model to classify heartbeats. Comput. Biol. Med. 2017, 89, 389396.

[34] Polat. Breast cancer diagnosis using least square support vector machine. Digit. Signal Process. 2007, 17, 694701.

[35] Huertas-Fernandez, I.; Garcia-Gomez, F.J.; Garcia-Solis, D.; Benitez-Rivero, S.; Marin-Oyaga, V.A.; Jesus, S.; Mir, P. Machine learning models for the differential diagnosis of vascular parkinsonism and Parkinsons disease using [123 I] FPCIT SPECT. Eur. J. Nucl. Med. Mol. Imaging 2015, 42, 112119.

[36] S. M. Jadhav, S. L. Nalbalwar and A. Ghatol, "Artificial Neural Network based cardiac arrhythmia classification using ECG signal data," 2010 International Conference on Electronics and Information Engineering, Kyoto, Japan, 2010, pp. V1-228-V1-231, doi: 10.1109/ICEIE.2010.5559887.

[37] S. Y. SEN ECG Arrhythmia Classification By Using Convolutional Neural Network And Spectrogram, 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), Izmir, Turkey, 2019, pp. 1-6, doi: 10.1109/ASYU48272.2019.8946417.

[38] C. Sarvan and N. Ozkurt, ECG Beat Arrhythmia Classification by using 1-D CNN in case of Class Imbalance, 2019 Medical Technologies Congress (TIPTEKNO), Izmir, Turkey, 2019, pp. 1-4, doi: 10.1109/TIPTEKNO.2019.8895014

[39] H. I. Bulbul, N. Usta and M. Yildiz, "Classification of ECG Arrhythmia with Machine Learning Techniques," 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 2017, pp. 546-549, doi: 10.1109/ICMLA.2017.0-104.

[40] M. S. Refahi, J. A. Nasiri and S. M. Ahadi, "ECG Arrhythmia Classification Using Least Squares Twin Support Vector Machines," Electrical Engineering (ICEE), Iranian Conference on, Mashhad, Iran, 2018, pp. 1619-1623, doi: 10.1109/ICEE.2018.8472615.

[41] P. -Y. Hsu and C. -K. Cheng, Arrhythmia Classification using Deep Learning and Machine Learning with Features Extracted from Waveform-based Signal Processing, 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), 2020, pp. 292-295, doi: 10.1109/EMBC44109.2020.9176679.

[42] Ozal Yildirim, Muhammed Talo, Edward J. Ciaccio, Ru San Tan, U Rajendra Acharya, Accurate deep neural network model to detect cardiac arrhythmia on more than 10,000 individual subject ECG records,Computer Methods and Programs in Biomedicine,Volume 197,2020,105740,ISSN 0169-2607, https://doi.org/10.1016/j.cmpb.2020.105740.

[43] M. Shahin, E. Oo and B. Ahmed, Adversarial Multi-Task Learning for Robust End-to-End ECG-based Heartbeat Classification, 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), 2020, pp. 341-344, doi: 10.1109/EMBC44109.2020.9175640.

[44] Zheng, J., Chu, H., Struppa, D. et al. Optimal Multi-Stage Arrhythmia Classification Approach. Sci Rep 10, 2898 (2020). https://doi.org/10.1038/s41598-020-59821-7

[45] X. Xu and H. Liu, "ECG Heartbeat Classification Using Convolutional Neural Networks," in IEEE Access, vol. 8, pp. 8614-8619, 2020, doi: 10.1109/ACCESS.2020.2964749.

CHAPTER 7

IMPLEMENTATION CODE

Install Packeges

```
[ ] !pip install -q keras
```

```
[ ] import keras
```

```
[ ] import numpy as np
    import pandas as pd
    import seaborn as sns
    from keras.layers import Dropout
    import matplotlib.pyplot as plt
    from sklearn.metrics import classification_report
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import f1_score
    from sklearn.metrics import confusion_matrix
    from keras.utils.np_utils import to_categorical
    from sklearn.utils import class_weight
    import warnings
    warnings.filterwarnings('ignore')
```

Import Dataset and View the Arch

```
[ ] #dataset
    train_df=pd.read_csv('/content/drive/MyDrive/ECG Data F/mitbih_train.csv',header=None)
    test_df=pd.read_csv('/content/drive/MyDrive/ECG Data F/mitbih_test.csv',header=None)
```

```
[ ] train_df[187]=train_df[187].astype(int)
    equilibre=train_df[187].value_counts()
    print(equilibre)
    train_df.shape
```

```
[ ] plt.figure(figsize=(20,10))
    my_circle=plt.Circle( (0,0), 0.7, color='white')
    plt.pie(equilibre, labels=['normal beat','unknown Beats','Ventricular ectopic beats','Supraventricular ectopic beats','Fusion Beats'],
            colors=['red','green','blue','skyblue','orange'],autopct='%1.1f%%')
    p=plt.gcf()
    p.gca().add_artist(my_circle)
    plt.show()
```

Resample Technique

```
[ ] #resample
    from sklearn.utils import resample
    df_1=train_df[train_df[187]==1]
    df_2=train_df[train_df[187]==2]
    df_3=train_df[train_df[187]==3]
    df_4=train_df[train_df[187]==4]
    df_0=(train_df[train_df[187]==0]).sample(n=20000,random_state=42)

    df_1_upsample=resample(df_1,replace=True,n_samples=20000,random_state=123)
    df_2_upsample=resample(df_2,replace=True,n_samples=20000,random_state=124)
    df_3_upsample=resample(df_3,replace=True,n_samples=20000,random_state=125)
    df_4_upsample=resample(df_4,replace=True,n_samples=20000,random_state=126)

    train_df=pd.concat([df_0,df_1_upsample,df_2_upsample,df_3_upsample,df_4_upsample])
```

Showing Resample Curve

```
[ ] equilibre=train_df[187].value_counts()
    print(equilibre)
```

```
[ ] plt.figure(figsize=(20,10))
    my_circle=plt.Circle( (0,0), 0.7, color='white')
    plt.pie(equilibre, labels=['normal beat','unknown Beats','Ventricular ectopic beats','Supraventricular ectopic beats','Fusion Beats'],
            colors=['red','green','blue','skyblue','orange'],autopct='%1.1f%%')
    p=plt.gcf()
    p.gca().add_artist(my_circle)
    plt.show()
```

Showing the ECG Curve after Resample

```
[ ] c=train_df.groupby(187,group_keys=False).apply(lambda train_df : train_df.sample(1))
```

```
[ ] plt.subplot(1, 5, 1)  # 1 line, 2 rows, index nr 1 (first position in the subplot)
    plt.plot(c.iloc[0,:186])
    plt.subplot(1, 5, 2)  # 1 line, 2 rows, index nr 2 (second position in the subplot)
    plt.plot(c.iloc[1,:186])
    plt.subplot(1, 5, 3)  # 1 line, 2 rows, index nr 2 (second position in the subplot)
    plt.plot(c.iloc[2,:186])
    plt.subplot(1, 5, 4)  # 1 line, 2 rows, index nr 2 (second position in the subplot)
    plt.plot(c.iloc[3,:186])
    plt.subplot(1, 5, 5)  # 1 line, 2 rows, index nr 2 (second position in the subplot)
    plt.plot(c.iloc[4,:186])

    plt.show()
```

Plot the Classes

```
[ ]  def plot_hist(class_number,size,min_):
         img=train_df.loc[train_df[187]==class_number].values
         img=img[:,min_:size]
         img_flatten=img.flatten()

         final1=np.arange(min_,size)
         for i in range (img.shape[0]-1):
             tempo1=np.arange(min_,size)
             final1=np.concatenate((final1, tempo1), axis=None)
         print(len(final1))
         print(len(img_flatten))
         plt.hist2d(final1,img_flatten, bins=(80,80),cmap=plt.cm.jet)
         plt.show()
```

```
[ ]  plot_hist(0,70,5)
```

```
[ ]   plt.plot(c.iloc[1,:186],color='red')
```

```
[ ]  plot_hist(1,50,5)
```

```
[ ]  plt.plot(c.iloc[2,:186],color='yellow')
```

```
[ ]  plot_hist(2,60,30)
```

```
[ ]  plt.plot(c.iloc[3,:186],color='grey')
```

```
[ ]  plot_hist(3,60,25)
```

```
[ ]  plt.plot(c.iloc[4,:186],color='green')
```

```
[ ]  plot_hist(4,50,18)
```

Add Gaussian Noise

```
[ ] #Gaussian Method
    def add_gaussian_noise(signal):
        noise=np.random.normal(0,0.05,186)
        return (signal+noise)
```

```
[ ] tempo=c.iloc[0,:186]
    bruiter=add_gaussian_noise(tempo)

    plt.subplot(2,1,1)
    plt.plot(c.iloc[0,:186],color='green')

    plt.subplot(2,1,2)
    plt.plot(bruiter)

    plt.show()
```

Dividing Dataset into Train and Test set

```
[ ] target_train=train_df[187]
    target_test=test_df[187]
    y_train=to_categorical(target_train)
    y_test=to_categorical(target_test)
```

```
[ ] X_train=train_df.iloc[:,:186].values
    X_test=test_df.iloc[:,:186].values
    #for i in range(len(X_train)):
    #    X_train[i,:186]= add_gaussian_noise(X_train[i,:186])
    X_train = X_train.reshape(len(X_train), X_train.shape[1],1)
    X_test = X_test.reshape(len(X_test), X_test.shape[1],1)
```

```
[ ] X_train.shape
```

```
[ ] #train and test samples
    print(X_train.shape[0], 'training samples')
    print(X_test.shape[0], 'testing samples')
```

1D CNN Proposed Model 01

```
[ ] #model
    def network(X_train,y_train,X_test,y_test):

        """
        #Model 01
        im_shape=(X_train.shape[1],1)
        inputs_cnn=Input(shape=(im_shape), name='inputs_cnn')
        conv1_1=Convolution1D(64, (6), activation='relu', input_shape=im_shape)(inputs_cnn)
        conv1_1=BatchNormalization()(conv1_1)
        pool1=MaxPool1D(pool_size=(3), strides=(2), padding="same")(conv1_1)
        drop = Dropout(0.2)
        conv2_1=Convolution1D(64, (3), activation='relu', input_shape=im_shape)(pool1)
        conv2_1=BatchNormalization()(conv2_1)
        pool2=MaxPool1D(pool_size=(2), strides=(2), padding="same")(conv2_1)
        drop = Dropout(0.2)
        conv3_1=Convolution1D(64, (3), activation='relu', input_shape=im_shape)(pool2)
        conv3_1=BatchNormalization()(conv3_1)
        pool3=MaxPool1D(pool_size=(2), strides=(2), padding="same")(conv3_1)
        drop = Dropout(0.2)
        flatten=Flatten()(pool3)
        dense_end1 = Dense(64, activation='relu')(flatten)
        #drop = Dropout(0.2)
        dense_end2 = Dense(32, activation='relu')(dense_end1)
        main_output = Dense(5, activation='softmax', name='main_output')(dense_end2)

        model = Model(inputs= inputs_cnn, outputs=main_output)
        model.compile(optimizer='adam', loss='categorical_crossentropy',metrics = ['accuracy'])
        history=model.fit(X_train, y_train,epochs=200,batch_size=16,validation_data=(X_test,y_test))
        return(model,history)
       """
```

Proposed Model 02

```
#model 02
 im_shape=(X_train.shape[1],1)
 inputs_cnn=Input(shape=(im_shape), name='inputs_cnn')
 conv1_1=Convolution1D(64, (6), activation='relu', input_shape=im_shape)(inputs_cnn)
 conv1_1=BatchNormalization()(conv1_1)
 pool1=MaxPool1D(pool_size=(3), strides=(2), padding="same")(conv1_1)
 drop = Dropout(0.2)
 conv2_1=Convolution1D(64, (3), activation='relu', input_shape=im_shape)(pool1)
 conv2_1=BatchNormalization()(conv2_1)
 pool2=MaxPool1D(pool_size=(2), strides=(2), padding="same")(conv2_1)
 drop = Dropout(0.2)
 conv3_1=Convolution1D(64, (3), activation='relu', input_shape=im_shape)(pool2)
 conv3_1=BatchNormalization()(conv3_1)
 pool3=MaxPool1D(pool_size=(2), strides=(2), padding="same")(conv3_1)
 drop = Dropout(0.2)
 flatten=Flatten()(pool3)
 dense_end1 = Dense(64, activation='relu')(flatten)
 drop = Dropout(0.2)
 dense_end2 = Dense(32, activation='relu')(dense_end1)
 drop = Dropout(0.2)
 main_output = Dense(5, activation='softmax', name='main_output')(dense_end2)

 model = Model(inputs= inputs_cnn, outputs=main_output)
 model.compile(optimizer='adam', loss='categorical_crossentropy',metrics = ['accuracy'])
 history=model.fit(X_train, y_train,epochs=150,batch_size=16,validation_data=(X_test,y_test))
 return(model,history)
```

Proposed Model 03

```
#model 03
 im_shape=(X_train.shape[1],1)
 inputs_cnn=Input(shape=(im_shape), name='inputs_cnn')
 conv1_1=Convolution1D(64, (6), activation='relu', input_shape=im_shape)(inputs_cnn)
 conv1_1=BatchNormalization()(conv1_1)
 pool1=MaxPool1D(pool_size=(3), strides=(2), padding="same")(conv1_1)
 #drop = Dropout(0.2)
 conv2_1=Convolution1D(64, (3), activation='relu', input_shape=im_shape)(pool1)
 conv2_1=BatchNormalization()(conv2_1)
 pool2=MaxPool1D(pool_size=(2), strides=(2), padding="same")(conv2_1)
 drop = Dropout(0.2)
 conv3_1=Convolution1D(64, (3), activation='relu', input_shape=im_shape)(pool2)
 conv3_1=BatchNormalization()(conv3_1)
 pool3=MaxPool1D(pool_size=(2), strides=(2), padding="same")(conv3_1)
 drop = Dropout(0.2)
 flatten=Flatten()(pool3)
 dense_end1 = Dense(64, activation='relu')(flatten)
 drop = Dropout(0.2)
 dense_end2 = Dense(32, activation='relu')(dense_end1)
 drop = Dropout(0.2)
 main_output = Dense(5, activation='softmax', name='main_output')(dense_end2)

 model = Model(inputs= inputs_cnn, outputs=main_output)
 model.compile(optimizer='adam', loss='categorical_crossentropy',metrics = ['accuracy'])
 history=model.fit(X_train, y_train,epochs=150,batch_size=16,validation_data=(X_test,y_test))
 return(model,history)
```

Proposed Final Model

```
#Model 04
 im_shape=(X_train.shape[1],1)
 inputs_cnn=Input(shape=(im_shape), name='inputs_cnn')
 conv1_1=Convolution1D(64, (6), activation='relu', input_shape=im_shape)(inputs_cnn)
 conv1_1=BatchNormalization()(conv1_1)
 pool1=MaxPool1D(pool_size=(3), strides=(2), padding="same")(conv1_1)
 drop = Dropout(0.2)
 conv2_1=Convolution1D(64, (3), activation='relu', input_shape=im_shape)(pool1)
 conv2_1=BatchNormalization()(conv2_1)
 pool2=MaxPool1D(pool_size=(2), strides=(2), padding="same")(conv2_1)
 drop = Dropout(0.2)
 conv3_1=Convolution1D(64, (3), activation='relu', input_shape=im_shape)(pool2)
 conv3_1=BatchNormalization()(conv3_1)
 pool3=MaxPool1D(pool_size=(2), strides=(2), padding="same")(conv3_1)
 drop = Dropout(0.2)
 flatten=Flatten()(pool3)
 dense_end1 = Dense(64, activation='relu')(flatten)
 dense_end2 = Dense(32, activation='relu')(dense_end1)
 main_output = Dense(5, activation='softmax', name='main_output')(dense_end2)

 model = Model(inputs= inputs_cnn, outputs=main_output)
 model.compile(optimizer='adam', loss='categorical_crossentropy',metrics = ['accuracy'])
 history=model.fit(X_train, y_train,epochs=150,batch_size=16,validation_data=(X_test,y_test))
 return(model,history)
```

Evolution Functions

```
[ ] from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score, classification_report, confusion_matrix

    def evaluate_model(history,X_test,y_test,model):
        scores = model.evaluate((X_test),y_test, verbose=0)
        print("Accuracy: %.2f%%" % (scores[1]*100))
        print('Test Loss:', scores[0])
        print('Test Accuracy:', scores[1])
        #precision_score = precision_score((X_test),y_test)
        #print('Precision score: %f' % precison_score)
        #recall_score = recall_score((X_test),y_test)
        #print('Recall Score: %f' % recall_score)
        #f1_score = f1_score((X_test),y_test)
        #print('f1 score: %f' % f1_score)



        print(history)
        fig1, ax_acc = plt.subplots()
        plt.plot(history.history['accuracy'])
        plt.plot(history.history['val_accuracy'])
        plt.xlabel('Epoch')
        plt.ylabel('Accuracy')
        plt.title('Model - Accuracy')
        plt.legend(['Training', 'Validation'], loc='lower right')
        plt.show()
```

```
[ ]
        fig2, ax_loss = plt.subplots()
        plt.xlabel('Epoch')
        plt.ylabel('Loss')
        plt.title('Model- Loss')
        plt.legend(['Training', 'Validation'], loc='upper right')
        plt.plot(history.history['loss'])
        plt.plot(history.history['val_loss'])
        plt.show()
        target_names=['0','1','2','3','4']

        y_true=[]
        for element in y_test:
            y_true.append(np.argmax(element))
        prediction_proba=model.predict(X_test)
        prediction=np.argmax(prediction_proba,axis=1)
        cnf_matrix = confusion_matrix(y_true, prediction)
```

```
[ ] from keras.layers import Dense, Convolution1D, MaxPool1D, Flatten, Dropout
    from keras.layers import Input
    from keras.models import Model
    from keras.layers.normalization import BatchNormalization
    import keras
    #from keras.callbacks import EarlyStopping, ModelCheckpoint

    model,history=network(X_train,y_train,X_test,y_test)
```

```
[ ] model.summary()
```

Confusion Matrix and Accuracy

```
[ ] evaluate_model(history,X_test,y_test,model)
    y_pred=model.predict(X_test)


    print(y_pred)
    print("Result from real time data included in testing dataset:")
    Y_pred_classes = np.argmax(y_pred,axis = 1)
    print(Y_pred_classes[0])
```

```
[ ] #confusion matrix
    from sklearn.metrics import f1_score, precision_score, recall_score, classification_report, confusion_matrix
    from sklearn import metrics

    import itertools
    def plot_confusion_matrix(cm, classes,
                              normalize=False,
                              title='Confusion matrix',
                              cmap=plt.cm.Blues):
        """
        This function prints and plots the confusion matrix.
        Normalization can be applied by setting `normalize=True`.
        """
        if normalize:
            cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
            print("Normalized confusion matrix")
        else:
            print('Confusion matrix, without normalization')
```

```
[ ]
        plt.imshow(cm, interpolation='nearest', cmap=cmap)
        plt.title(title)
        plt.colorbar()
        tick_marks = np.arange(len(classes))
        plt.xticks(tick_marks, classes, rotation=45)
        plt.yticks(tick_marks, classes)

        fmt = '.2f' if normalize else 'd'
        thresh = cm.max() / 2.
        for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
            plt.text(j, i, format(cm[i, j], fmt),
                     horizontalalignment="center",
                     color="white" if cm[i, j] > thresh else "black")

        plt.tight_layout()
        plt.ylabel('True label')
        plt.xlabel('Predicted label')

    # Compute confusion matrix
    cnf_matrix = confusion_matrix(y_test.argmax(axis=1), y_pred.argmax(axis=1))
    np.set_printoptions(precision=2)

    #f1_score = cnf_matrix
    #print('f1 score' %f1_score)
    # Plot non-normalized confusion matrix
    plt.figure(figsize=(10, 10))
    plot_confusion_matrix(cnf_matrix, classes=['N', 'S', 'V', 'F', 'Q'],normalize=True,
                          title='Confusion matrix, with normalization')
    plt.show()
```