Detection of Arrhythmia

Vaishnavi Balaji

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

13th December 2021

Abstract

Electrocardiogram (ECG) is a health monitoring procedure that helps medical professionals to observe the cardiac activity of the human heart. It is a common procedure used for the quick detection of heart diseases and to monitor their activity. Detection of arrhythmia in its early stages enables ease of treatment, in turn, reduces the risk of the patient's health. In recent developments in telecommunication, ECG can be recorded and sent through the telephone, internet, or any other data communication links. These conventional ECG machines can be used in rural areas where healthcare is primary and also where professionally trained cardiologists are not available. Different machine learning and deep learning architectures and approaches are used in various ECG classification tasks such as arrhythmia detection and classification, annotation of different waves, procedures during the sleep stage, data denoising and so on. The combined architecture of a CNN and RNN with some ensemble techniques using expert features is observed to yield the best results. Interpretability, scalability, and efficiency while scaling the ECG data remain as some of the few concerns while building this classification model. Therefore this thesis focuses on studying the heartbeats of various arrhythmia conditions and devising a novel way to predict the heartbeat types using deep techniques using CNN, LSTM, and GRU. Evaluation and comparing the performance of the models using confusion matrix, precision, recall, and F1 score.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form.

Name: Vaishnavi Balaji Signature: *Vaishnavi Balaji*

Acknowledgments

I would like to thank Dr. Fani Deligianni for her continuous support and guidance throughout the dissertation period. I would also like to thank my friends and family for their support and encouragement during this challenging time.

Contents

1 Introduction

Electricity, an innate biotic electricity is what keeps the heart beating. Electrocardiogram (ECG) is merely a graphical representation of the heart's electrical activity and rhythm of its muscles. The electrical activity of the human heart produces current, which then passes from the surrounding tissue to the skin. When the electrodes are attached to the human body, they sense these currents passing through and transmit them to the ECG monitor. The signals are then converted into waveforms, which represent the depolarization and repolarization of the heart cycle. Depolarization is the crucial electrical event, where the cardiac cell loses its internal negativity. The flow of electricity produced by depolarization waves propagates from one cell to another across the entire heart. After depolarization, these cardiac lessors restore their polarity through repolarization, where the membrane pumps reverse the flow of ions. All of these are detected by the electrodes attached to the human skin, which then projects in the monitor.

According to the WHO's International Cardiovascular Disease Statistics, approximately 16.7 million deaths are from cardiovascular diseases (CVD) per year around the globe and around 31% of total deaths in the world are primarily caused by CVD (1). In developing countries like India and China CVD is more numerous than in economically developed countries put together. There are many socio-economic causes surrounding it, but an accurate algorithm that helps in the early detection of CVD is very important as it enables ease of treatment as well as saves millions of people's lives.

1.1 Interpreting ECG complex

The ECG complex exemplifies the electrical event occurring in one cardiac cycle of the human heart. Each segment or wave in the ECG corresponds to a certain activity of the cardiac electrical cycle. A single ECG complex consists of five waveforms labeled as P, Q, R, S, and T. The QRS peaks in the complex are the combination of three deflections seen on an ECG. The first deflection is known as a P wave, Q is the first negative deflection to the baseline, R is the highest positive deflection, S is the second negative deflection, followed by T wave. It is produced due to the depolarization of the left and the right ventricles of the heart, followed by the contraction of large ventricular muscles.

The P waves represent the contraction and the atrial depolarization. The first and the second part of the P waves record the right and left atrial activity respectively. There is a quick pause when the electricity reaches the AV node and the ECG falls at the PR segment. Following this Ventricular depolarization takes place which generates the QRS complex. Next, the T wave reports ventricular repolarization. Atrial repolarization is usually not seen in ECG.

The PR section records the start time of atrial depolarization till the start time of ventricular depolarization. Then the PR segment reports the end time of atrial depolarization to the start time of ventricular depolarization. Then the ST segment measures the time between the end of ventricular depolarization and ventricular repolarization. Likewise, the QT interval records the starting of ventricular depolarization

till the end of ventricular repolarization. The QRS peak unit in total measures the time of the ventricular depolarization.

The amplitude of normal QRS is usually around 5 to 30 mm and the duration is 0.06 to 0.12 seconds. The width, shape, and amplitude of the QRS complex are useful in diagnosing cardiac arrhythmias, ventricular hypertrophy, electrolyte derangements, conduction abnormalities, myocardial infarction, and other CVD-related states.

The shape of abnormal QRS varies from normal to wide beats and bizarre or slurred-looking beats. The Tall spiked QRS is the result of hypertrophy of ventricles or is usually caused either by an aberrantly conducted beat or by an abnormal pacemaker. On the other side, the low QRS complex is due to obesity, hyperthyroid or pleural effusion.

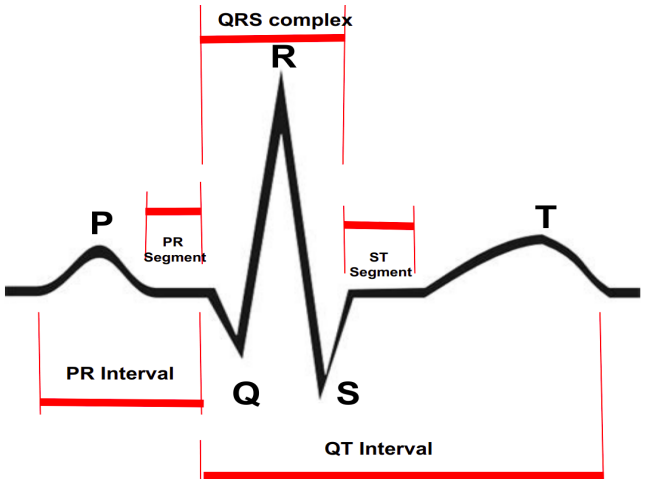

Figure 1.1 Interpretation of ECG wave

1.2 Arrhythmia

The Normally functioning human heart beats around 60 to 100 times per minute following the regular rhythm. The usual cardiac rhythm is called a normal sinus rhythm. Any other distortion in the rhythm is usually categorized as arrhythmia. The term arrhythmia refers to the abnormal heart rhythm caused by any distortion in the rate, regularity, site of origin, or conduction of the cardiac electrical activity. An arrhythmia could either be a single abnormal beat or a prolonged rhythm disturbance for the lifespan of the patient. But not every variant of arrhythmia is life-threatening. For instance, well-trained athletes usually have a low heartbeat around 35 to 40 per minute. However, some arrhythmia can be very dangerous and may even lead to sudden death. The instant diagnosis of arrhythmia is one of the most important uses of ECG.
Many arrhythmias are discreet and the symptoms are usually overlooked by the patients. These can be picked up either by continuous physical examination or by ECG. This makes arrhythmia detection much more challenging. However, the most important symptoms are palpitations. Patients may observe an unusual accelerated or decelerated heartbeat or a long pause between heartbeats. The decreased cardiac output is usually the one to look out for. The sudden occurrence of arrhythmia with the underlying cardiac vascular disease can even lead to congestive heart failure.

Arrhythmia is classified into many different types, some of the major types include Sinus arrhythmias, Premature Beats, Brady & Tachy, Atrial vs Junctional vs Ventricular, Heart Blocks.

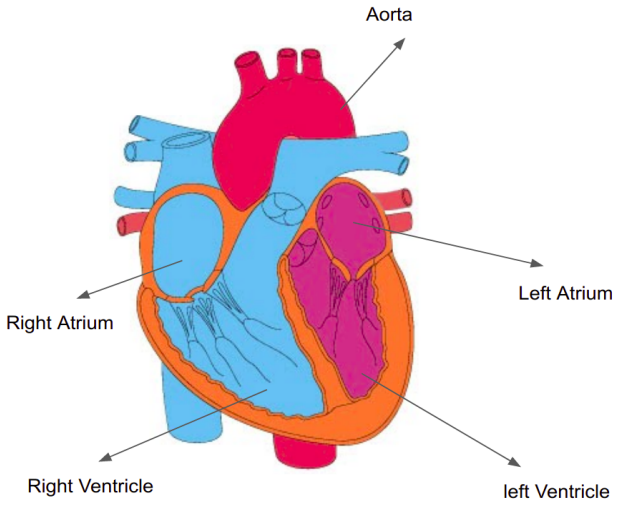

Figure 1.2 Heart Chambers (Clipart source: Internet)

1.3 ECG Lead configuration

A lead is simply a perspective of the human heart's electrical activity. 12 Lead ECG is a diagnostic test, performed on patients with arrhythmias, angina, and acute myocardial infarction to get an absolute perspective of the heart's electrical activity. This 12 Lead ECG is recorded by placing the electrodes in the patient's chest wall and extremities. The limb electrodes include three bipolar leads Lead I, Lead II, Lead III and three unipolar augmented leads (a V_R , a $V_{L,}$ and a V_F) . Unipolar chest Leads are represented as V1, V2, V3, V4, V5, V6. These 12 different leads provide 12 different views of the electrical activity.

In this study, we use a 2 Lead ECG configuration, where the signals are recorded by positioning the electrodes on the chest and limbs. The two leads are limb electrode lead II (MLII) and in the majority of cases the other lead is the chest electrodes V1 but rarely V2 or V5, and in one case it is V4. These decisions are made by the BIH arrhythmia Laboratory since QRS complexes are prominent in these upper signals. There are some modifications in leads due to the different medical conditions of the patients. For example for records 102 and 104, V5 lead is used for the upper signal because of the surgical dressings of the patients.

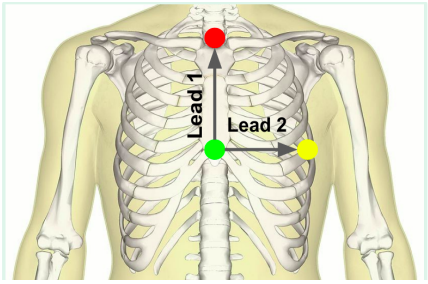

Figure 1.3 Two Lead ECG configuration (Clipart source: Internet)

2. Literature Survey

In this chapter, a summary of the previous work on detecting arrhythmia using the ECG signals is showcased. Since our study mainly focuses on using neural networks for the arrhythmia classification, the literature survey will focus more on the machine learning and deep learning techniques carried out till now.

Traditional approaches in ECG classification take a two-step approach to the raw time-series data. The first step is usually finding the expert features which usually requires domain experts to do the engineering. Then the next step is to deploy the machine learning methods for classification. These human-engineered expert features can be categorized into statistical features, frequency domain features, and time-domain features according to Shenda Hong(2). In the practical deep learning approaches, these expert features can be automatically detected. But these expert features are very limited and are highly subjected to data quality and human bias (3). In contrast, deep learning has achieved significant advancements in many fields including speech recognition, computer vision, and natural learning applications (4). One of the main reasons is that deep learning performs automatic feature extraction and also processes flexible architecture which makes it much more versatile to use in different fields.

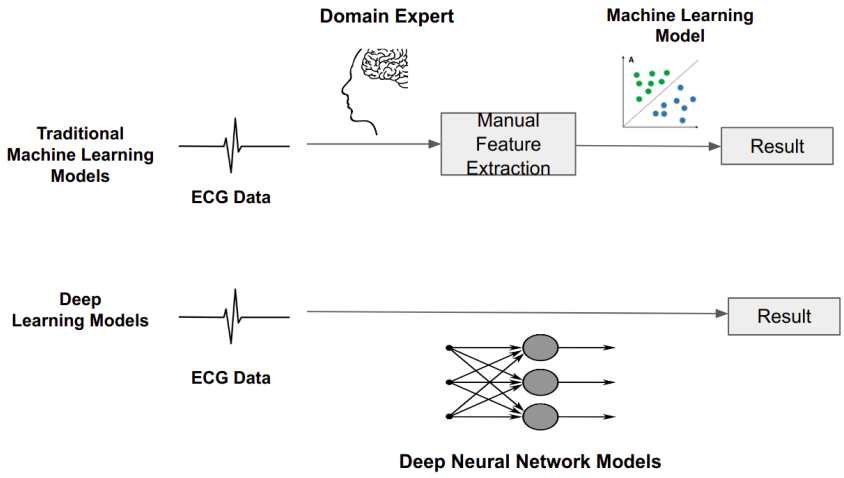

Figure 2.1 Traditional Machine Learning models vs Deep Learning models(Clipart source: Internet)

2.1 CNN

CNN's are widely used in image classification, signal processing, and natural language processing. A typical CNN framework is proposed with a few convolutional layers, with batch normalizations, dropout layer, nonlinear activation, pooling layer, and the final fully connected classification layers are discussed in (5). Typically CNN achieves superior performance because of their shared weight architectures and parallelizations. CNN can be applied to ECG data into two waves, one as a raw ECG data signal and the other as a 2D ECG signal which is used as an image and further proceeded as an image classification problem. In such cases like in (6) the ECG data are treated as an image and pre-trained models like

DenseNet, ResNet or Inception-Net can be trained on the ImageNet dataset and is further fine-tuned to fit the ECG image data. In this paper (7), the localization task to find the QRS detection is done by reducing the image height and maintaining it equal to the output length. One of the other approaches is data augmentation (8) and active learning(9) into the CNN which handles the imbalanced data problem in the MIT-BIH Arrhythmia dataset which in turn improves accuracy.

2.2 RNN

A Recurrent Neural network is a special type of neural network built to handle time-series data, sequential data, and natural language. The important feature of RNN is that it takes the previous state information along with the current input and updates them in the hidden state, this will end up remembering data in sequential order. CNN has a significant problem of vanishing and exploding gradient descent, which is usually handled well by RNN. In support of that, RNN is well suited for time-series data which may have different input lengths and can also capture temporal dependencies. Some of the commonly used RNN variants are GRU/LSTM and bidirectional-LSTM (BiLSTM), because of their unique ability to remember previous inputs. In this paper, the author approaches an autoencoder based on LSTM and also filters out the noise using the Butterworth function thus overcoming the gradient disappearance in the traditional CNN methods (10). In this paper(11), Two small LSTM networks are combined in the raw ECG features for doing the continuous real-time execution on devices. A bidirectional LSTM network is proposed to improve the performance and interoperability using the attention mechanism in (12).

2.3 CRNN

A CRNN is a combination of both CNN and RNN modules. CRNN is the much-suited model for ECG signals with multiple channel inputs and varying input lengths. Usually, features are extracted using the regular CNN and then RNN summarizes the local features with time to generate the global features. One of the popular research is DeepHeart(13) which adapts CRNN to analyze the cardiovascular disease risk prediction from the ECG data using the AE model to initialize weight and to improve performance. Another notable research in CRNN is MINA(14), this paper provides an interpretable diagnosis. They achieve this by incorporating a CRNN architecture which makes use of multi-level attention mechanism along with medical domain knowledge like beat level, rhythm-level, and frequency-level expert features.

2.4 GAN

A GAN is a special type of neural network, which consists of a generative model and a discriminative model. These two models work together iteratively to capture the latent features and the true data distribution. In this paper (8), the authors achieved significant improvement in performance by using GAN for data augmentation to handle the imbalanced dataset in the MIT-BIH arrhythmia dataset, followed by the CRNN framework. Another approach is taken in (15), where GAN is used for data denoising of the ECG data. To generate synthetic data representation this paper (16) proposed BiLSTM as a generator and CNN as a discriminator and trained their model for better performance.

3. Current State Analysis

The MIT-BIH database has been publicly available for commercial and academic purposes for more than 25 years now. This leads to numerous advancements in automatic real-time ECG classification. Existing systems take ECG data as input and use deep CNN and other heavy deep learning algorithms for training and classification, which usually require hardware with high computational power. This limitation makes it unsustainable for it to be implemented in the practical medical field. On the other hand, medical practitioners and Cardiologists analyze the ECG patterns manually and spot any deviation in the waves and annotate the distortion. Manual Analysis is usually prone to a high degree of misclassification and is a very time-consuming process. And also in rural areas, where they have no special access to expert cardiologists, it might cost someone their life. Therefore the development of a more accurate and low-cost model for arrhythmia classification is essential. Some of the things which concern the development of the automatic arrhythmia detection systems are manual feature extraction by domain expert which might have human bias, choosing the best algorithm for classification, and most important of all is how to make use of the imbalance classes in the database. Moreover, much research surrounding ECG classification focuses very little on the minor classes.

In contrast to the heavyweight algorithms, this study mainly focuses on two important aspects. Firstly, the data preprocessing including data denoising, Z score normalization, and sampling techniques are performed on the raw ECG data. Secondly, three neural network architectures based on CNN, LSTM, and GRU are proposed and their performance and classification are evaluated and compared.

4. System Requirements

Deep Learning libraries mainly rely on parallelism, therefore using a simple CPU might not be very efficient. But if the task at hand is a bit computational heavy, a GPU would make a better choice. We can either use a high-end laptop with an inbuilt GPU like Nvidia GTX 1080 (8 GB VRAM) or build our PC with a powerful GPU. For commercial purposes, deep learning models are usually handled by cloud services like Google Cloud, Azure, and AWS.

In our study, the dataset is around 495 MB, and three neural networks are constructed which makes use of GPU services provided by google cloud and kaggle to run quickly. The total time it takes for each model is around 20-30 minutes. However, it might vary slightly according to the type of GPUs provided by the providers.

4.1 Software requirements:

- Google Collab
- Sklearn
- Keras
- Numpy
- Pandas
- Matplotlib
- Seaborn
- Pywt
- Scipy

4.2 Hardware requirements:

- Compatible GPU for better performance or Google collab GPU service
- 16GB RAM recommended
- Minimum of Intel Core i7 processor
- Minimum of 4GB storage
- Windows or Linux Operating System

5. Data Pre-processing

In this chapter, a brief description of the preprocessing techniques that are performed to the raw ECG data before fitting it to the model is given.

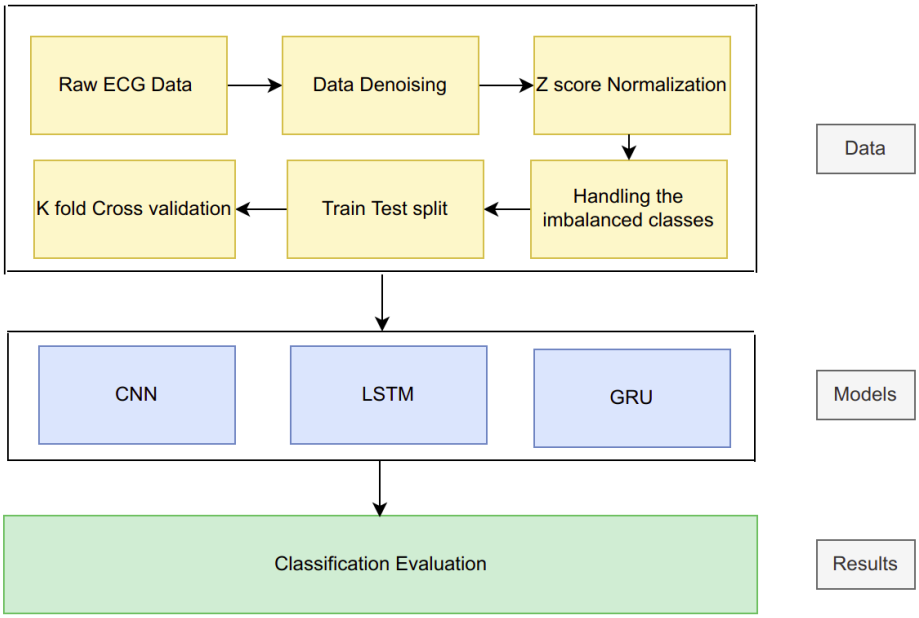

Figure 5.1 Overview of the study

5.1 Database Description

The MIT-BIH Arrhythmia Database was originally obtained by Beth Israel Hospital Arrhythmia Laboratory around the year 1975 to 1979. It consists of a set of over 4000 Holter recordings from both the inpatients and the outpatients with the 60:40 ratio. The patients were 25 men around 32 to 89 years old and 22 women around 23 to 89 years old. The Database records numbers from 100 to 124 (with some numbers missing) were chosen randomly from the above set. This group represents the different waveforms and artifacts that can be encountered in clinical use by the arrhythmia detector. Then the records from 200 to 234 (with some missing numbers) were selected to represent a variety of rare and clinically important samples of the Holter recordings. This set of records includes the complex arrhythmia morphologies, few rare and complex cases of ventricular, junctional, supraventricular arrhythmias.

In this study, we use a 2 Lead ECG configuration, where the signals are recorded by placing the electrodes on the chest and limbs. The two leads are limb electrode lead II (MLII) and in the majority of cases the other lead is the chest electrodes V1 but occasionally V2 or V5, and in one case it is V4. These decisions are made by the BIH arrhythmia Laboratory since QRS complexes are prominent in these upper signals. There are some modifications in leads due to the different medical conditions of the

patients. For example for records 102 and 104, V5 lead is used for the upper signal because of the surgical dressings of the patients.

All the recordings are over the range of 10 mv and are digitized at 360 samples per second per channel along with the 11-bit resolution. Initially, the annotations were made by two or more cardiologists independently, and the discrepancy was reviewed and resolved together. In total, approximately 110,000 annotations are made computer-readable and are constantly reviewed and updated in the database frequently in case of any error.

There are many different types of arrhythmia that can be detected. In this paper, the focus is on classifying the arrhythmia given in the MIT-BIH database, which was originally annotated by cardiologists into 15 different types and is then furthermore grouped into five major different classes followed by the recommendations of the Association for the Advancement of Medical Instrumentation (AAMI).

AAMI class	MIT-BIH original class	Type of beat
Normal (N)	N L R E j	Normal beat Left bundle branch block beat Right bundle branch block beat Atrial escape beat Nodal (junctional) escape beat
Supraventricular ectopic beat (S)	A a J S	Atrial premature beat Aberrated atrial premature beat Nodal (junctional) premature beat Supraventricular premature beat
Ventricular ectopic beat (V)	V E	Premature Ventricular contraction Ventricular escape beat
Fusion beat (F)	F	Fusion of ventricular and normal beat
Unknown beat (Q)	/ f Q	Paced beat Fusion of paced and normal beat Unclassifiable beat

Table 5.1: Heartbeat described in the MIT-BIH Arrhythmia database grouped by AAMI standard

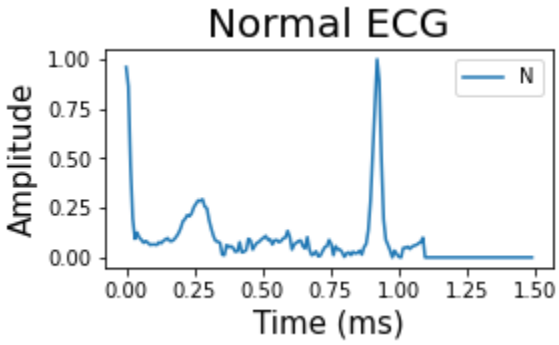

Figure 5.2 Normal Heartbeat (N)

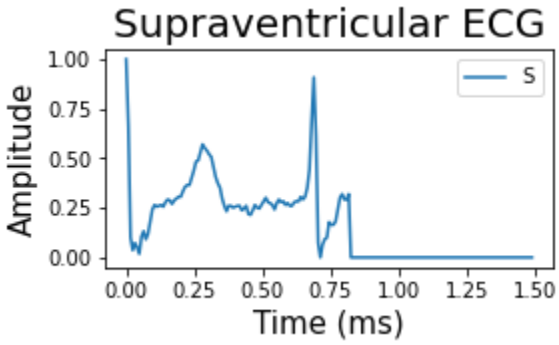

Figure 5.3 Supraventricular beat (S)

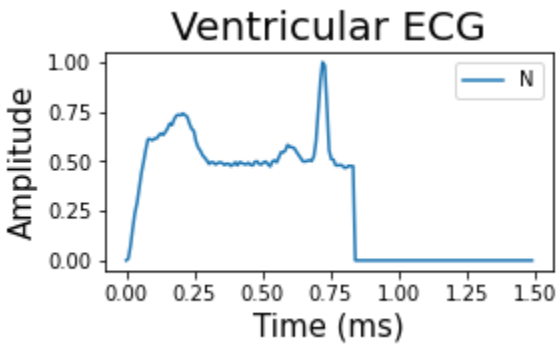

Figure 5.4 Ventricular beat (V)

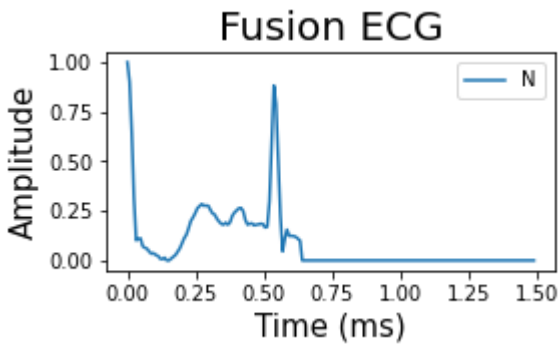

Figure 5.5 Fusion beat (F)

5.2 Data denoising

The ECG recording performed in the clinical environment is usually disrupted by power line interference, wandering baseline during respiration, and Electromyography which records the electrical activity of the skeletal muscles. Therefore all these noises have to be removed to make the ECG classification more robust and accurate. Different solutions have been proposed for denoising of ECG data and are being used in a number of applications. Few most common techniques used for denoising in the ECG data are bandpass filters, low-pass filters, and wavelet transform. In this study, the wavelet transforms methods with an adaptive threshold filtering algorithm are used which select Sym4 as the primary wavelet function.

Wavelets are wave-like oscillations that are localized in space or time. The two basic properties of wavelets are scale which tells how stretched or squished the wavelets are with their frequency and the location of where the wavelet is positioned in time. The basic idea behind the wavelet transform is it evaluates the convolution of signals and wavelets at varying scales. Fourier Transform, which is a decomposition using sine and cosine in itself observes the global frequency information that persists over the signal and makes obscure local information. This makes it irrelevant to use it in ECG signals since it has characteristic oscillation in short intervals. Therefore using wavelet transform which extracts the local spectral and temporal information and decomposes the nonstationary signals into a set of wavelets of different frequency bands is appropriate for ECG signals. An adaptive threshold filtering method is used where the same threshold is not used through the whole record, the thresholding changes depending on the data. The main algorithm used in this study is the Sym4 wavelet function. The 'sym4' wavelet

resembles the QRS complex, which is useful for QRS detection. It is more preferred for analysis. The length of the filter is denoted as 2n, where n is the no. of vanishing movement.

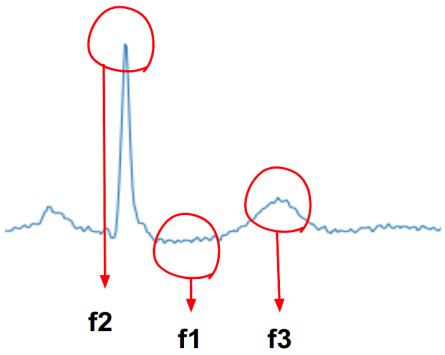

Figure 5.6 Types of Frequency

Here the f1 denoted the high-frequency noise and the f2 denoted the QRS frequency and the f3 represented the P or T wave which are the slowly varying components of the ECG. Out of these three different frequencies, detecting f2 is the main objective. The relationship between the frequency is F1 > F2 > F3. The objective is to preserve the R peaks frequency (f2) and eliminate the rest (f1, f3). So bandpass filtering is needed and can be achieved by eliminating some frequency bands. Eliminating the low and high frequencies and only keeping the middle frequency. So this can be done using wavelet transform, because it is more robust, and works well with all the shapes, while normal frequency selector fails in certain shapes. Because in some cases the QRS complexes are not sharp, sometimes they are wider and shorter and tend to fall under f3.

This Bandpass filtering can be achieved by eliminating wavelet coefficients of some lower scales (high frequency) and high scales (low frequency) of ECG signal. For this purpose undecimated wavelet transform is used to get wavelet coefficients. A 4-level decomposition of an ECG signal using sym4 is shown. Coefficient a4 is not considered because it is carrying all the low-frequency signals which we do not want, similar d1 and d2 will also be eliminated because they are carrying the very high-frequency signals.

In a normal wavelet transform, the signals are downsampled after every decomposition and it's every reduced after every decomposition label. But in an undecimated wavelet transform the signal length remains the same. So only d3 and d4 are considered for getting the BPF effect. After that, we take d3 and d4 and take an inverse wavelet transform to get the signal. Here the R peaks are well preserved and other P, T is completely removed and noises are also reduced. The Standard peak detection algorithm is used to locate these R peaks. With the help of the total number of R peaks within the time interval, we can calculate the heartbeat rate.

We are computing the maximum level of useful decomposition with the input data, the wavelet filter length as inputs and it returns the maximum level. In the coefficient, we are performing the multilevel 1D discrete wavelet transform of data. Here the data, the sym4 function, and the maxlev are taken as input, and then the threshold is given to the coefficient. Then the inverse discrete wavelet transformation is performed and returned in this function.

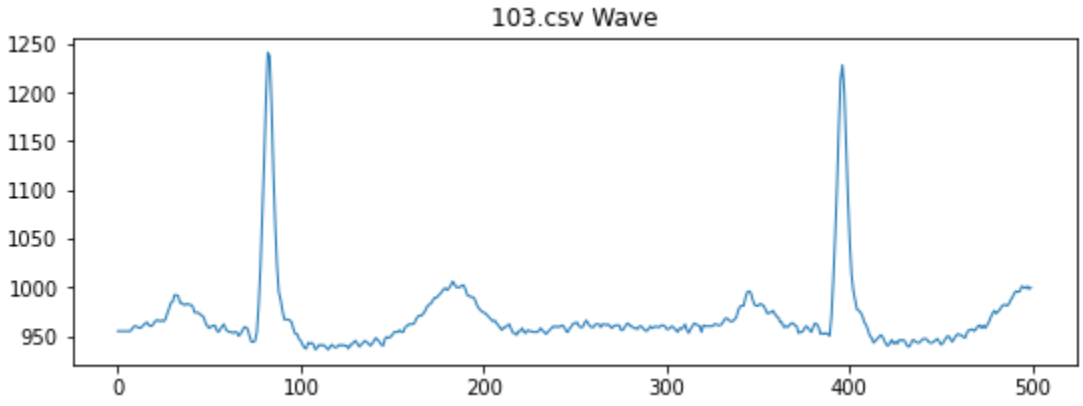

Figure 5.7 Waves before denoising

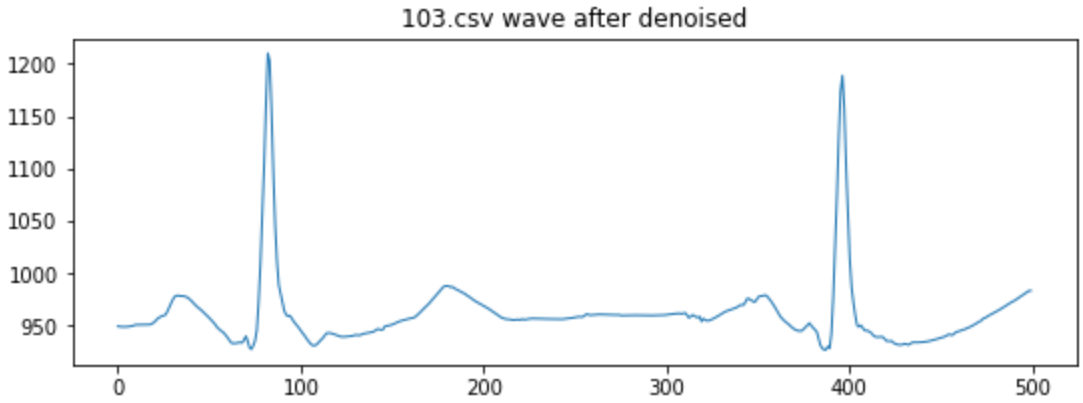

Figure 5.8 Waves after denoising

5.3 Data segmentation:

In the MIT-BIH arrhythmia dataset, each and every heartbeat is annotated with the disease. In this study, we classify them according to the five AAMI classes. The Data is segmented by using the famous algorithm called Pan-Tompkins algorithm to detect the R-peaks in every heartbeat. The process is done by taking a single lead in the dataset and then segmenting them using the Z score normalization method into 360 samples centered around R peaks.

$$z = \frac{x - mean(x)}{stdev(x)}$$

Z score Normalization

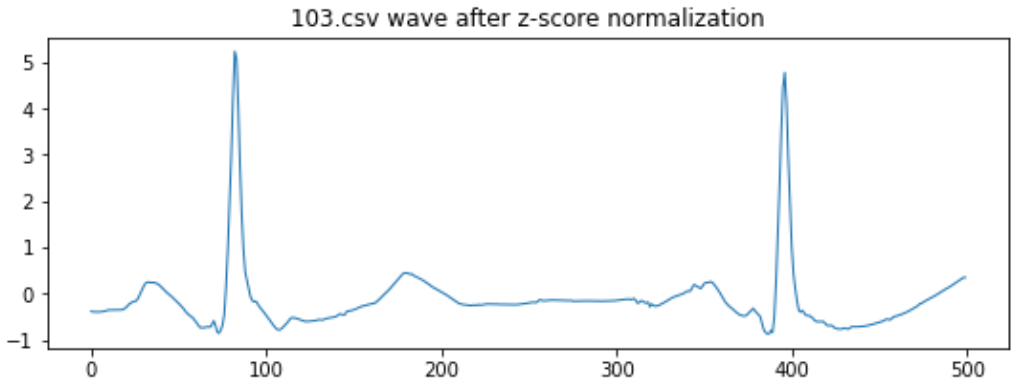

Figure 5.9 Waves after z score normalization

5.3.1 Pan-Tompkins algorithm for QRS detection:

The Pan-Tompkins algorithm was proposed by Jiapu pan and Willis J Tompkins in 1985, in the journal IEEE Transaction on Biomedical engineering (17). This algorithm is the mostly used to spot the QRS complexes in the ECG signals. The main spike in the ECG signal is the QRS complex which represents the ventricular depolarization of the heart. This is an important feature to measure the heartbeat rate, which is the primary assessment meant to evaluate the heart conditions. This algorithm removes the background noise by applying a series of bandpass filters and also highlight the frequency of rapid heart depolarization. Then amplifies the QRS complex by squaring the signals. Finally, an adaptive threshold is applied to detect the peaks in the filtered signals.

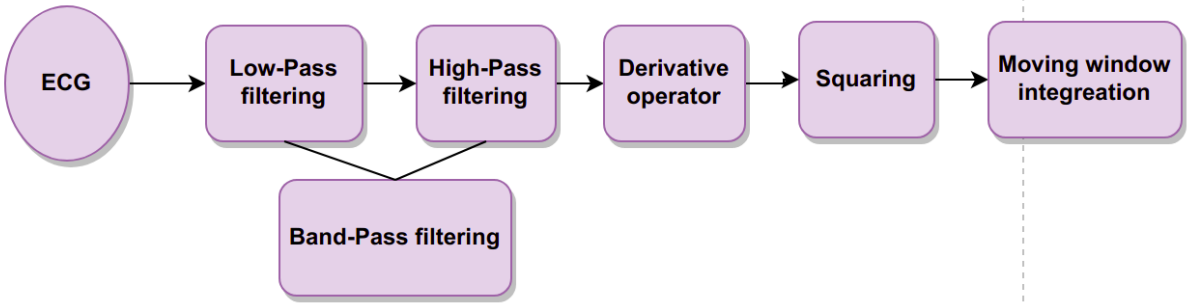

Figure 5.10 Pan-Tompkins algorithm

The threshold θ is the main parameter used since it filtered the signal that is higher than the threshold and marks it as QRS complex. Noises in the signals are filtered in the bandpass filters. Then the derivative filter evaluates the information of the slopes. The amplitude squaring is done and averaged and passed through the moving window integrator to determine the true R peaks. There the sampling is done at 360 signals/s.

(i) Bandpass filter

This filter removed the noise by cascading the low pass and the high pass filter. Removing the noise makes the classification more accurate and help in detecting R peaks

1. Low pass filter transfer function is given by

$$H(Z) = \frac{(1-z^{-6})^2}{(1-z^{-1})^2}$$

2. High pass filter transfer function is given by

$$H(Z) = z^{-16} - \frac{(1-z^{-32})}{(1-z^{-1})}$$

(ii) Derivative filter

After the bandpass filtering stage, the signals are passed through the derivative filter to calculate the slope. We have employed a four-point derivative.

(iii) Squaring filter

After the derivative filter, all the points are squared to make the values positive and is denoted by the following equation

$$y[nT] = (x[nT])^2$$

(iv) Integrator

Then the signals are integrated with the slope information and then thresholded to locate the true R peaks.

Thus the Pan-Tompkins algorithm based on differentiated ECG signals is computed and can be used in real-time analysis to detect the R peaks.

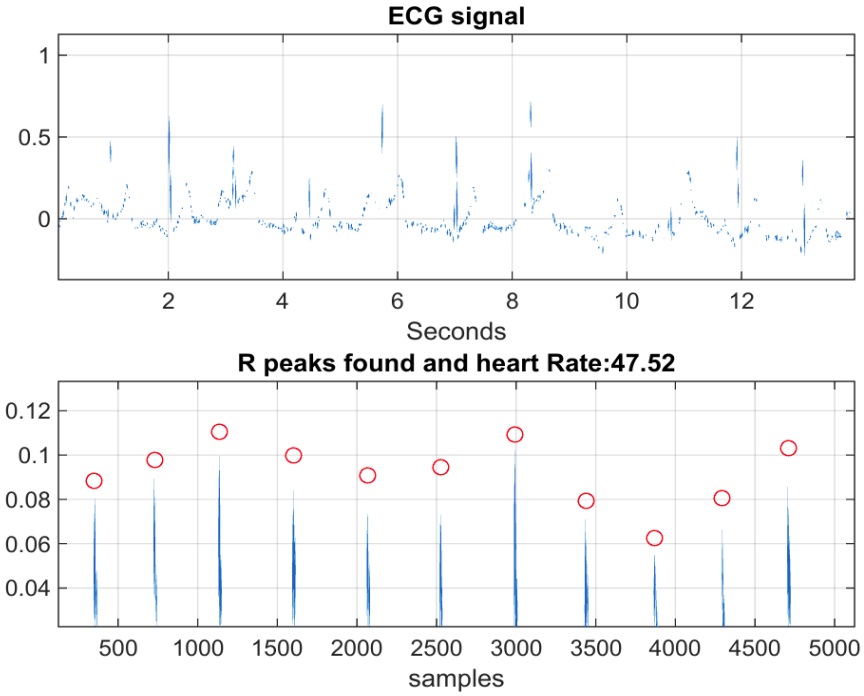

Figure 5.11 R peak detection and heartbeat rate

5.4 Class Distribution

Imbalanced classes in training data are one of the biggest challenges in real-world data since machine learning and deep learning models and algorithms are built around the assumption of a perfectly balanced class. So balancing the classes before the model implementation is the crucial step to optimize classification accuracy. In our study, the distribution of the classes is given below.

Here as shown in the figure 75% of the data below belongs to class N, and the next major distribution is in class L with 8.1% and class R contributes to 7.3%, class A to 7.1%, and finally class V has the smallest distribution of 2.5%. Our study focuses mainly on classifying the different types of arrhythmia, predicting the minor classes is much more important and sensitive compared to the classification of the normal heartbeat class. Severe imbalance in the training data makes it harder to predict the minority classes since only a few samples of them are given to the model which makes it harder to learn the characteristics features of the minority class. So this imbalance in the dataset should be handled before the model training.

In this study oversampling is performed on the minority classes. It duplicates the minority classes and randomly selected from the majority classes, which overall balances the training data set. So After the sampling techniques, the class distribution in our training dataset looks like the below figure.

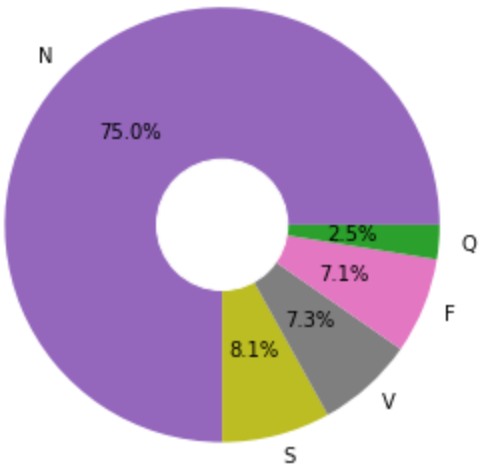

Figure 5.12 Class Distribution before sampling

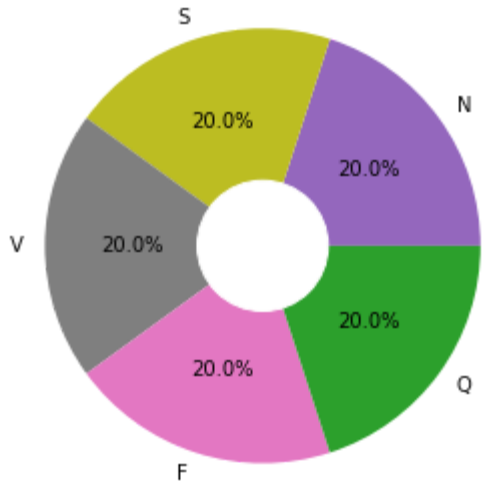

Figure 5.13 Class Distribution after sampling

5.5 Train - Validation - Test Split

The next important step after balancing the classes is splitting the train and test data. Here the traditional method of splitting the data into 80:20 is used. The training dataset contains 80% of the data and the testing data consists of 20% of the data. After this step, the data are reshaped in order to fit the CNN model. The final shape of the test and train data are given by

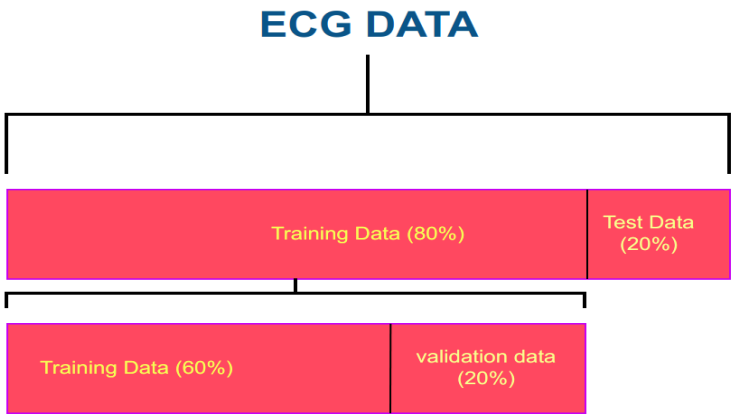

Figure 5.14 Splitting of data

5.5 KFold Cross-Validation

KFold cross-validation is a resampling technique usually used to evaluate the performance of the learning models on the unseen data. Initially, the dataset is shuffled randomly, then it is split into K sets. In this study, we use 5 fold cross-validation. Then a group is taken as a hold-out set which acts as validation data, while the remaining groups act as training data. Then we fit the model with these data and evaluate the results on the validation set. The score is stored in a list and the model is discarded. These procedures are repeated for each set and finally score stored in the list helps us to summarize the performance of the model and compare it with the other models. This helps the model to generalize more and gets rid of any biases in the data while splitting.

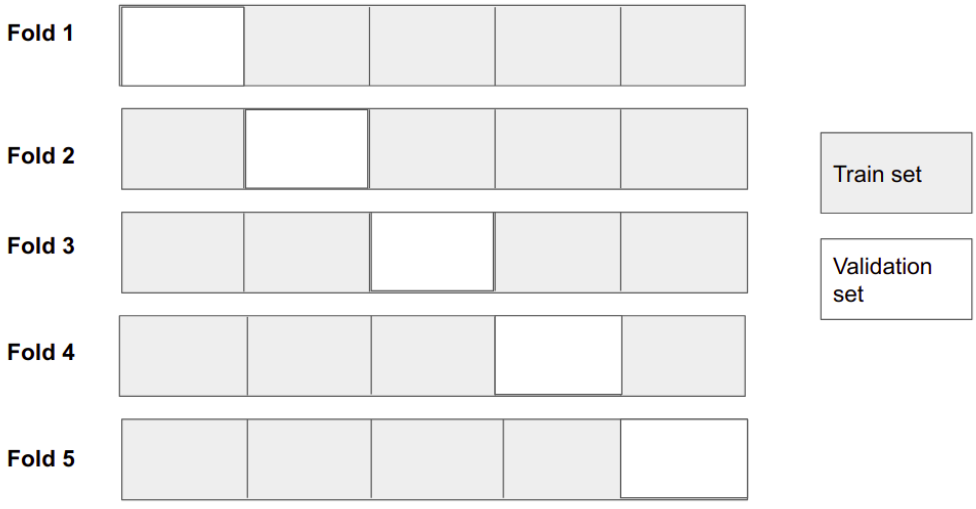

Figure 5.15 KFold cross-validation

6. Neural Network Models

In this chapter, a brief description of the CNN, LSTM, and GRU neural networks are given, along with a detailed summary of the proposed models.

6.1 CNN

Convolutional neural networks are a type of artificial neural networks, which are specialized in detecting the patterns in the data. This makes CNN much more useful than a traditional ANN or multilayer perceptron. The basics of CNN is their convolutional layer which takes the input and undergoes a convolution operation to the desired output and passes it to the next layer. These operations performed by the convolutional layers are called cross-correlations. Each convolutional layer has a specific number of filters, these filters determine the number of output channels and are primarily responsible for detecting the patterns. The deeper the convolutional neural network goes, the these filters become more and more sophisticated. The filters are also known as kernels that read the data bit by bit. This size is determined by the window size. These CNN filters have specific weights that are learned during the training process, therefore making it much more meaningful in finding the hidden pattern than a human-designed filter. Stacking these filters and layers makes the CNN undergo hierarchical feature learning, which is very similar to how our brain learns to identify the objects.

(i) Convolution Layer:
Convolutional Layers are the most significant layers in the convolutional neural networks. The convolution layer applies the inputs to the filters repeatedly which in turn creates a feature map that denotes the most important features in the input. Generally, the filters are smaller than the input size to find the overlapping part, then an element-wise dot product is performed between the inputs and filter which in turn gives a scalar value. After applying filters multiple times it results in a feature map, which are then passed to a nonlinearity function like ReLu. This systematic application of the filters results in translation invariance.

$$h = \sigma(\sum_{n}^{N} W^T x + b)$$

Here h is the output of the neurons in the convolutional layer. σ is the activation function and b is the offset of the neurons in the given layer l. x is the output of the neurons from layer $l-1$. And w is the weight matrix of the k^{th} kernel in l^{th} layer.

(ii) Average Pooling Layer
Pooling layers are usually followed by a convolutional layer. Pooling layers reduce the model complexity by reducing the dimensions of the output data. Thus handles the overfitting issue and increases the robustness of the network. There are two types of pooling layers that either average or maximize the output. In our study, we use the average pooling since they tend to preserve the overall features of the ECG data better than the max-pooling layers.

$$o = \sigma(\alpha\, pool(x) + b)$$

Here O is the output of the pooling layer, which σ is the activation function, with α as the sampling weight with offset b and $pool()$ represents the pooling function

(iii) Flatten

After stacking multiple convolutional and the pooling layer, the model would have obtained a pooled feature map. In this flatten layer we simply flatten the feature map into a single long vector of input data, which is then passed through the traditional fully connected layers of the Artificial Neural networks.

(iv) Dropout

The dropout layer usually comes before the fully connected layer. Since the fully connected layer has a large number of neurons, multiple neurons tend to learn the same features from the input data. This co-adaptation happens with neurons that have identical weights. This will lead to overfitting by duplicating those co-adapted features and also increase the computation power. To overcome this we drop out neurons that are below the threshold value. Some fraction of neurons r_d is zeroed at each training step. The remaining neuron is multiplied by $\frac{1}{1-r_d}$, so the total number of neurons remains the same. Thus hidden features are learned better by solving the co-adaption

(v) Fully connected Layer

After the dropout layer, the fully connected dense layer helps us in expanding the connectivity of all the features. This layer passes the weighted sum of the outputs to the softmax layer which then classifies the inputs into one of the five classes. Fully connected layers are the second most important and resource-consuming layer after the convolutional layer. The formula for the fully connected layer is given by

$$o = f(wx + b)$$

Here o is the output of the layer and the f is the activation function, where b and w are the offset and weight of the neurons while x is the output of the previous layer.

This study proposes a one-dimensional 12 layer convolutional neural network architecture that is used as a sequential model in which the inputs are taken one tensor after another and passed to the stack of layers to give the output classification. A sequential model won't be appropriate if our data has multiple inputs or multiple outputs, since that's not the case we are creating a list of layers in the sequential model. The weights in the sequential models are learned as the inputs are given sequentially from the training data. Initially, a convolution layer is created with the input shape 360 x 1 with 16 filters of kernel size 13 each, with zero padding and a relu activation layer. Then the Average pooling layer is stacked with pool sizes 3 and 2 strides. And then another layer of convolution is added with the kernel size of 15 with 32 filters. After that, another average pooling layer of size 3 is added to reduce the input size to 89 x 32. Layer 5 is a convolutional layer with 64 filters of kernel size q7. Another layer of average pooling layer of size 3 is added which reduces the output size of 44 x 64. This output is then convoluted with the 128 filters of kernel size 19 in layer 7. A final average pooling layer of size 3 is added to the architecture. The next layer is a dropout layer which drops almost 50% of its neurons by setting the dropout threshold to 0.5. The next two layers are the dense layer with 35 and 5 output neurons with L2 regularization, which in turn is sent to the final softmax layer with 5 neurons. These 5 neurons represent the 5 classes in our classification problems. The regularization in the dense layers helps us minimize the overfitting problem. The relu activation function is used in all pooling layers.

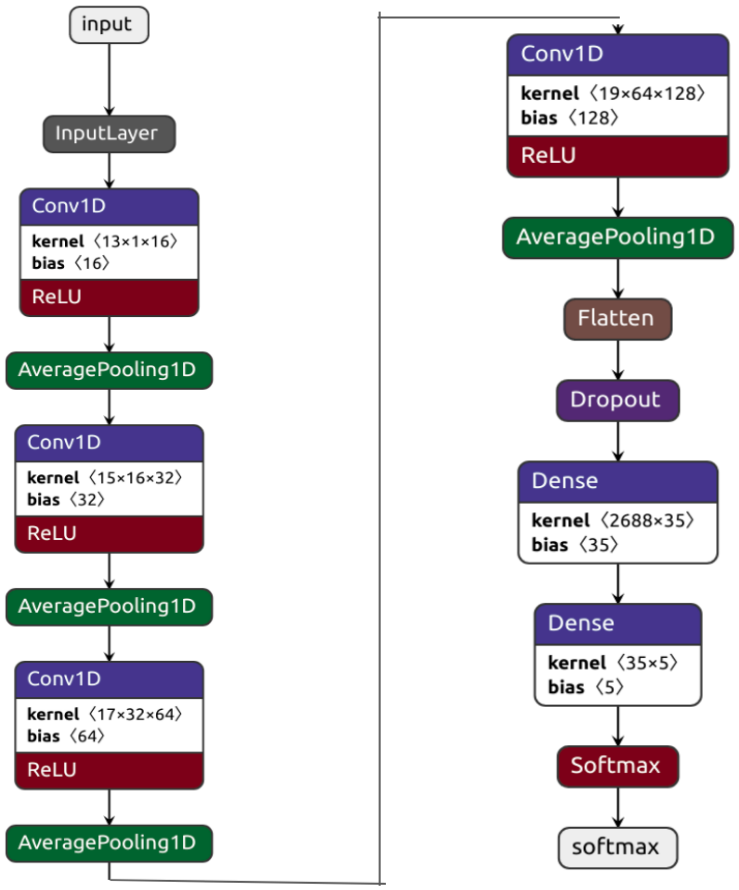

Figure 6.1 Proposed 12 Layer CNN Architecture

Layers	**Input Shape (N)**	**No. of Filters (n)**	**Filter Size (f)**	**Padding (p)**	**Strides (s)**	**Size of each feature map S= (N-f+2p)**	**No. of parameters (P)**
Convolution Layer C1	360 x 1	16	13 x1	True	1	360	224
Average Pooling Layer S2	360 x 16	16	3 x1	0	2	179	32
Convolution Layer C3	179 x 16	32	15 x 1	True	1	360	512
Average	179 x 32	32	3 x 1	0	2	89	64

Pooling Layer S4							
Convolution Layer C5	89 x 32	64	17 x 1	True	1	89	1152
Average pooling Layer S6	89 x 64	64	3 x 1	0	2	44	128
Convolution Layer C7	44 x 64	128	19 x 1	True	2	21	256
Average Pooling Layer S8	44 x 128	128	3 x 1	0	2	21	256

Flatten	128 * (21 * 1) = 2688 neurons
Dropout S9	0.5

Layer	**No. of Inputs**	**No. of outputs**	**No.of parameters**	**Kernel Regularizer**	**Bias Regularizer**	**Learning Rate**
Dense Layer S10	2688	35	94115	L2	L2	0.0001
Dense Layer S11	35	5	180	L2	L2	0.0001
Softmax S12	5	5	-	-	-	-

Table 6.1 Internal structure of the proposed CNN architecture

One of the few drawbacks of CNN is that it needs a lot of training data to efficiently learn the patterns in the data. This is a major drawback in this case since the data we collected is very limited. And the lack of ability to encode the position and the orientation of the ECG signal annotations also puts CNN at a major disadvantage. In this study, we try to overcome this disadvantage of CNN, by comparing its performance with Long Short Term Memory (LSTM).

In RNN if the data sequence is long enough, they will have a hard time carrying information from one step to another. So important information might be lost from the beginning

6.2 LSTM

Long Short Term Memory (LSTM) is specifically designed for modeling time-series data and also to overcome the problem of vanishing and exploding gradient descent in the traditional Recurrent Neural Network (RNN). In a typical LSTM, some information about the data is retained and some information is forgotten. In this study, Some information about the ECG signals like the R peaks, heartbeat rate, QRS complex, and other important annotations are remembered and other less important features are forgotten by the LSTM. LSTM has incorporated gate units and memory cells in their architecture; they are responsible for efficiently maintaining and transmitting the features through the LSTM for classification. The typical LSTM architecture is given below in fig.

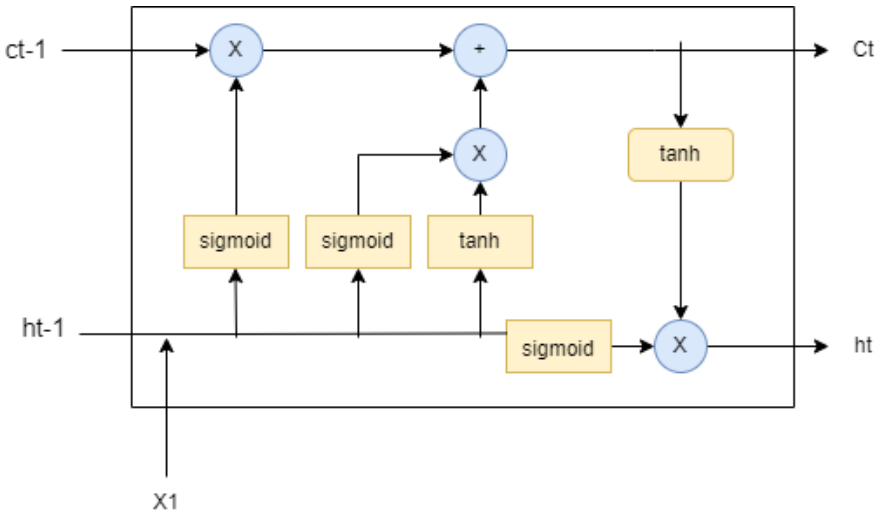

Figure 6.2 Structure of LSTM internal gates

The LSTM architecture consists of three major gates namely input gate, output gate, and forget gate. The output of the forget gate in the LSTM decides whether the information is kept in the memory of the forgotten. Here the h_{t-1} represents the information from the previous hidden state and x_t is the information of the current input. This information is then passed to the sigmoid function, which returns the values between 0 to 1 (f_t). If the output of the forget state is close to 1 then the information is remembered, on the other hand, if the output is closer to 0 it is forgotten. The equation of the forget gate is as follows.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_t)$$

Forget Gate Equation

The input gate in the LSTM architecture determines the information that enters the memory. The information in the cell state has to go through the sigmoid function [0,1] which adds to the memory. Because of the sigmoid activation function, the output from that cell state will always be between [0,1] therefore no forgetting information only retention. In contrast, the input gate has a tanh activation function which gives us the value between the range [-1,1] which allows us to either forget or remember the memory. The information h_{t-1} and are also passed in the input gate and then to tanh to squeeze the values between -1 to 1. Which helps us in regulating the network c_t.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

Input Gate Equation

$$c_t = tanh(W_c[h_{t-1}, x_t] + b_c)$$

Input Modulation Gate Equation

And for the output gate, the previously hidden state information h_{t-1} and the current information is sent to the sigmoid function and then the newly modified cell state is passed to the tanh function. These two outputs from both the activation function are then multiplied to get a new hidden state h_t.

$$h_t = O_t \bullet tanh\,(c_t)$$

Output Hidden State Equation

This is how the Long Short Term Memory (LSTM) architecture works and trains itself to decide which information is remembered and which information is forgotten.

However, the LSTM RNN relies on having sufficient training and testing data and also the assumption of the data coming from the same distribution. But the ECG data we have may have time-varying characteristics which may mislead the model classification. To overcome this problem we are doing transfer learning with LSTM.

In this study, LSTM architecture is combined with the fully connected layer for the model classification. In the first layer of the sequential model, LSTM is introduced with the 64 neurons with the input shape of (187,1) then a fully connected dense layer is introduced with 128 neurons with the relu activation function. Then a dropout layer with the threshold of 0.3 is introduced to improve the performance. Finally, the output layer is a fully connected dense layer with 5 neurons with the softmax activation function.

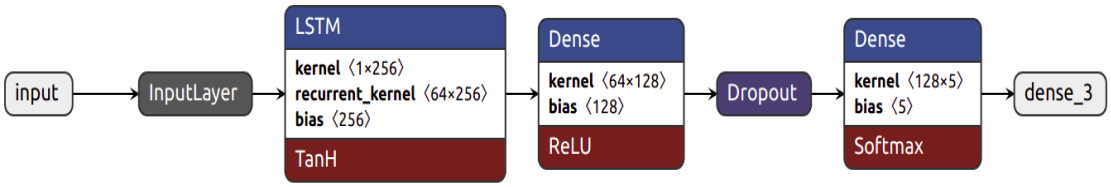

Figure 6.3 Proposed 4 Layer LSTM Architecture

Layer	No. of Inputs	No. of outputs	No.of parameters	Dropout threshold
LSTM layer L1	360 x 1	64	16896	-
Dense Layer S2	64	128	8320	-
Dropout Layer S3	-	-	-	0.3
Dense Layer S4	128	5	645	-

Table 6.2 Internal structure of the proposed LSTMarchitecture

6.3 GRU

The architecture of GRU is pretty similar to that of LSTM. Instead of cell state, hidden state transfers the information and has only two gates namely reset and update gate. In GRU the update gate does the same job as the forget gate and input gate in LSTM. It decides if the information is kept in the memory or thrown away. Then the reset gate decides how much previous state information to forget. Since GRU has fewer gates and fewer tensor operations, these are relatively faster to train than LSTM. Both are used appropriately for the requirements.

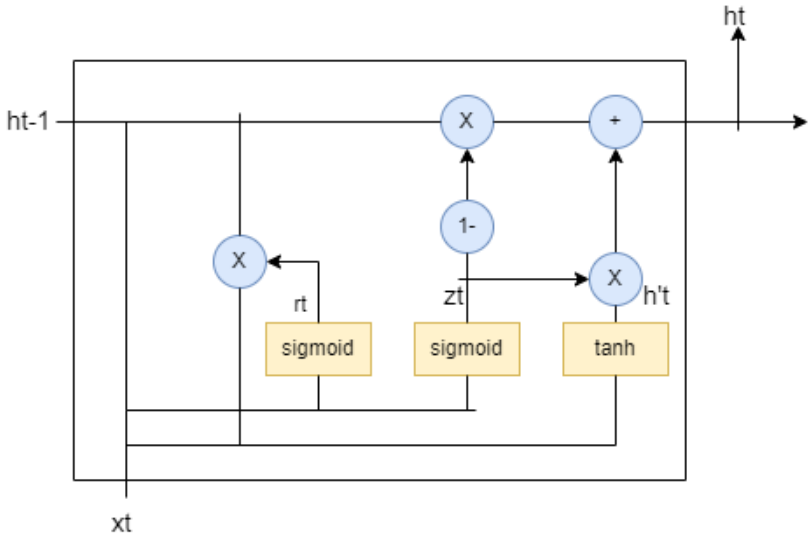

Figure 6.4 Structure of GRUinternal gates

The output gate equation is given by

$$z_t = \sigma(W^{(Z)}x_t + U^{(Z)}h_{t-1})$$

Input Gate Equation

Initially, x_t is multiplied with their own weights $W^{(Z)}$, and previous state information h_{t-1} is multiplied with weights $U^{(Z)}$ which is then added and passed into the sigmoid activation function which gives the values between 0 to 1. This gate copies all the data from the previous states and thus we successfully eliminate the vanishing gradient problem.

The similarity in the reset gate

$$r_t = \sigma(x_t * U_u + H_{t-1} * W_u)$$

Reset Gate Equation

Works similar to the output gate with weight U_u and W_u and are added and passed to the sigmoid function which in turn squeezes the values from 0 to 1.

In this study similar to the LSTM architecture, GRU is connected to the fully connected layer for model classification. First, the GRU layer is added with 64 neurons with the input shape (187,1) and then a dense layer of 128 neurons with the ReLu activation function is added. To reduce the model complexity the dropout layer is introduced with the threshold of 0.3. Then a final dense layer with 5 neurons is added with a softmax activation function.

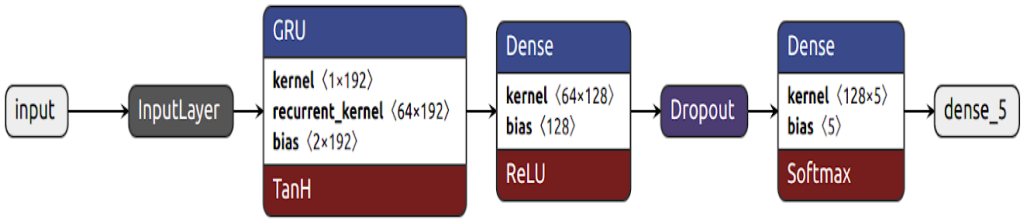

Figure 6.5 Proposed 4 Layer GRU Architecture

Layer	No. of Inputs	No. of outputs	No.of parameters	Dropout threshold
LSTM layer L1	360 x 1	64	12864	-
Dense Layer S2	64	128	8320	-
Dropout Layer S3	-	-	-	0.3
Dense Layer S4	128	5	645	-

Table 6.3 Internal structure of the proposed GRUarchitecture

7. Evaluation

As mentioned above in this study the original ECG data is categorized into five different classes before the neural network training and classification. In this chapter, a detailed discussion of the performance of different neural network models proposed and their performance comparison is done.

7.1 Accuracy

In Multi-class classification problems, Accuracy is defined as the total number of correct predictions predicted by the model to the total number of predictions for each class averaged together.

$$Accuracy = \frac{1}{N}[\sum_{K=1}^{G} \sum_{x:g(x)=K} I(g(x) = g'(\mathrm{x}))]$$

Here the Indicator function ***I*** return the value 1 if the prediction and the true value classes are the same, or return 0 if it's a mismatch.

The table below gives us the accuracy values for the baseline KNN and three different neural networks performed in our study.

Models	**Accuracy**
KNN (Baseline)	0.953
CNN	0.986
LSTM	0.750
GRU	0.963

Table 7.1 Accuracy of the proposed models

Classification accuracy is a good estimate of the model, but it gives us the false sense of belief because of its high accuracy like in CNN we have 99%, but it still doesn't mean our model is performing very good nor it is ready to be implemented in a real-world application. Here the main problem is the cost of misclassification of the minor arrhythmia classes. If the classification is wrong, it might lead to very dangerous consequences for the participant. So other factors should also be taken into account while evaluating the model.

7.2 Logarithmic Loss

Logarithmic loss penalizes the misclassification done by the model. During the training process, the proposed model predicts the probability of class for each data sample. The logarithmic loss is calculated as follows.

$$Logarithmic\ Loss = \frac{-1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} y_{ij} * log(p_{ij})$$

Here N is the number of samples, and y_{ij} records whether the sample i belongs to class j. Meanwhile, p_{ij} denotes the probability that the sample I is belonging to class j. The logarithmic loss falls between the range [0, ∞). The more the loss is closer to 0, the more accurate the model is. In general, Log Loss and accuracy are inversely proportional to each other. So for a good classification model, it should have high accuracy and minimum log loss. The log loss for the proposed model is given below.

Models	**Log Loss**
CNN	0.054
LSTM	0.720
GRU	0.140

Table 7.2 Logarithmic Loss of the proposed models

7.3 Confusion Matrix

Confusion Matrix is a matrix with the combination of actual and the predicted classes, which demonstrated the overall performance of the proposed models. It consists of 4 important terms :

- **True Positives**: When the model prediction is YES, the actual output is YES.
- **True Negatives**: When the model prediction is NO, the actual output is NO.
- **False Positives**: When the model prediction is YES, the actual output is NO.
- **False Negatives**: When the model prediction is NO, the actual output is YES.

The main diagonal in the confusion matrix shows all the true predictions by the proposed models, which forms the basis for determining other metrics. T
Accuracy can be calculated by averaging the values in the main diagonal and it forms the basis for the following types of metrics. The confusion matrix for the proposed models is given by

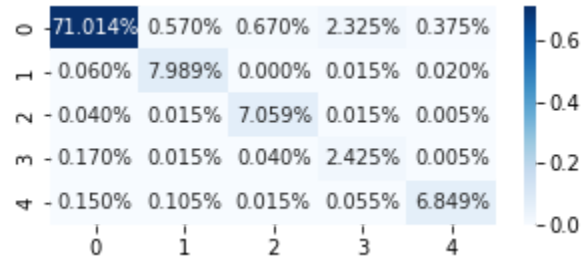

Figure 7.1 Confusion matrix for KNN

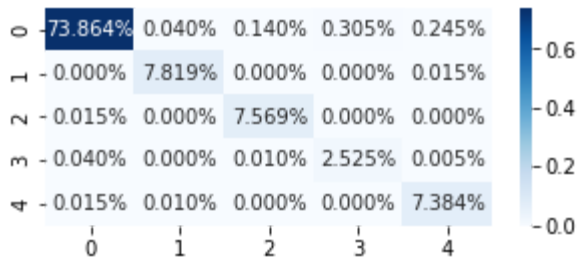

Figure 7.2 Confusion matrix for CNN

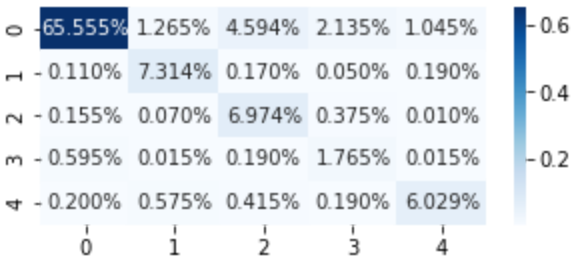

Figure 7.3 Confusion matrix for LSTM

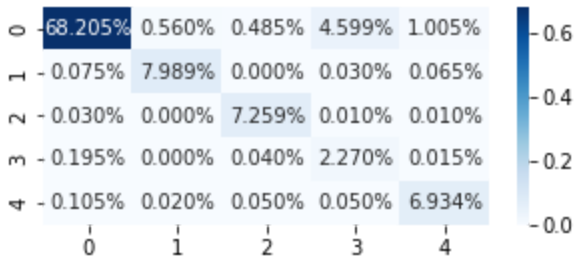

Figure 7.4 Confusion matrix for GRU

7.4 Precision, Recall, F1 score

The F1 score is calculated by taking the harmonic mean of the precision and recall, which ranges between [0,1]. The F1 score is a good indicatore to tell how precise and how robust our classifier is based on teh how many samples it classified correctly without mising significan number of samples. If the model shows high precision with low recall, then it indicates that it missed a large number of samples that are tricky to classify even though it shows high accuracy. The best-performing model usually has a higher F1 score.

Mathematically, F1 is expressed as :

$$F1 = 2 * \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}}$$

- Precision: It is the fraction of the number of true positives by the number of positive results predicted by the proposed model.

$$Precision = \frac{True\ positive}{True\ Positive + False\ Positives}$$

- Recall: It is the fraction of the number of true positive results divided by the total number of all relevant samples.

$$Recall = \frac{True\ positive}{True\ Positive + False\ Negatives}$$

Models	Weighted Average Precision	Weighted Average Recall	Weighted Average F1 Score
KNN (Baseline)	0.97	0.95	0.96
CNN	0.99	0.99	0.99
LSTM	0.87	0.75	0.78
GRU	0.97	0.96	0.97

Table 7.3 Precision, Recall, F1 score for the proposed models

7.5 Summary

In this chapter, we discussed the evaluation metrics and the performance of the proposed model. It was observed that the well-developed deep Convolutional neural network performs significantly better than a simple recurrent neural network like LSTM and GRU. This implies that the deeper the layer is, the more and more information is captured in the model, thus resulting in better classification.

8. Conclusion

An optimized classification model that can early detect arrhythmia will be able to reduce the number of deaths caused by cardiovascular disease every year. Most of the existing models require a machine with high computation power and hence making it unsustainable for medical practice. In this study, three neural networks were proposed and their performance was compared with each other and with the baseline machine learning model. The main observations are (i) Deep learning models outperform the traditional machine learning model since feature extraction is automatically done by the neurons instead of manual feature engineering (ii) a 12 layer deep CNN model performs much better than a 4 layer LSTM and a 4 layer GRU model. (iii) Even though LSTM is a powerful model, it underperformed when compared to a deep CNN model (iv) a simple GRU model performs slightly better than LSTM but no better than deep CNN. (v) Even Though CNN is better, it still struggles to classify the minority classes well.

In conclusion, the 12 layers proposed CNN model is concluded better with 99% accuracy, 99% precision, 99% recall compared with the machine learning baseline model, 4 layer LSTM, and 4 layer GRU.

8.1 Future work

For future work, it would be interesting to observe if a deep layer LSTM or transfer learning methods with the pre-trained models would outperform the proposed CNN. The limitation of this study is that we used K Fold cross-validation, instead of in the future we could try using StratifiedKFold and see if it produces any significant difference in the performance, since stratified preserves the weights of the classes during splitting. And also this study makes use of the 2 Lead ECG configuration, in the near future, we could try using the 12 Lead ECG configurations, which gives us a much deeper view of the heart's electrical activity.

Appendix A - Accuracy and loss graph

Accuracy and loss graph for proposed models

CNN:

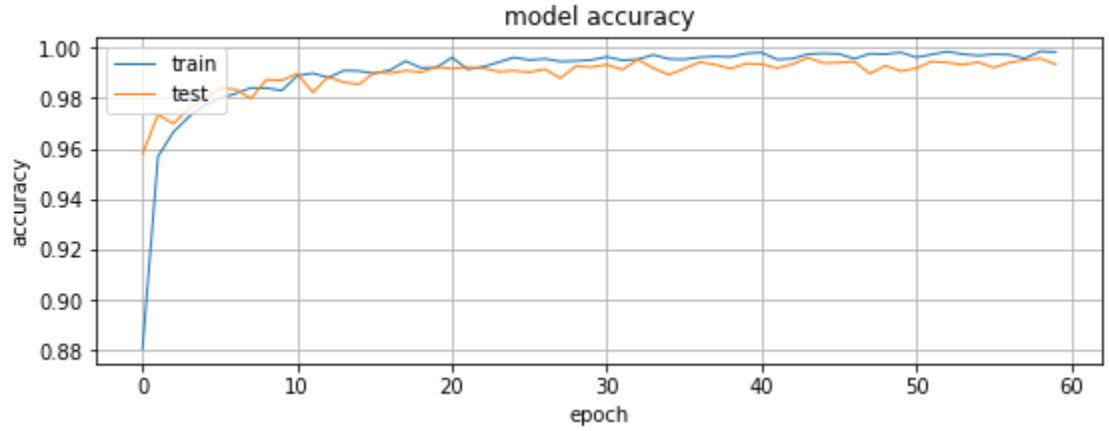

Figure A.1 Accuracy graph for CNN

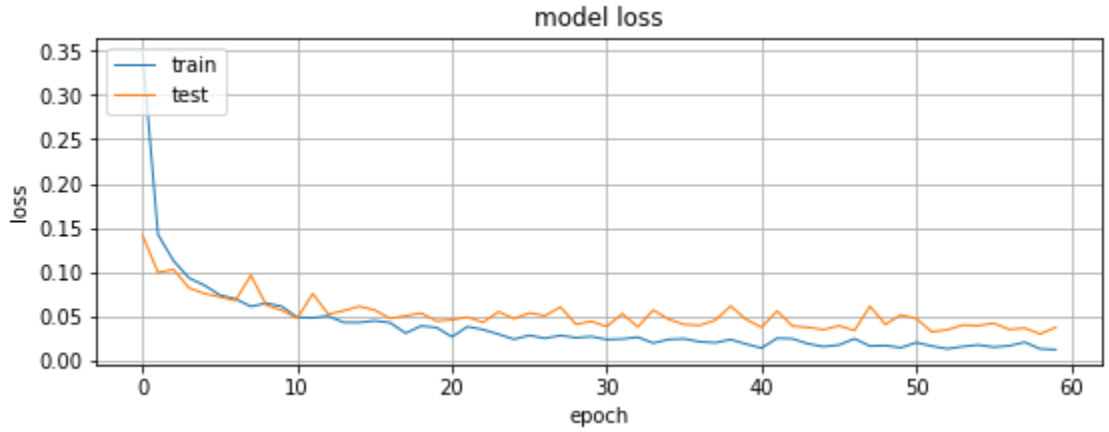

Figure A.2 Loss graph for CNN

LSTM:

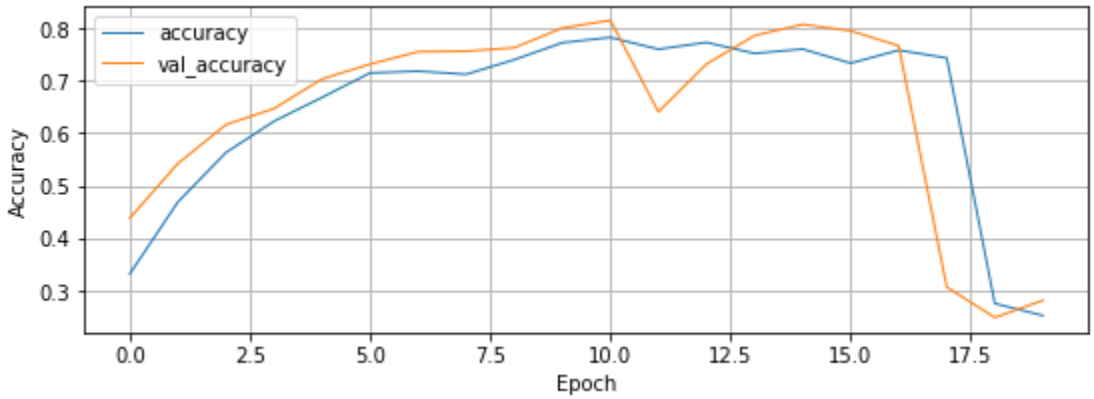

Figure A.3 Accuracy graph for LSTM

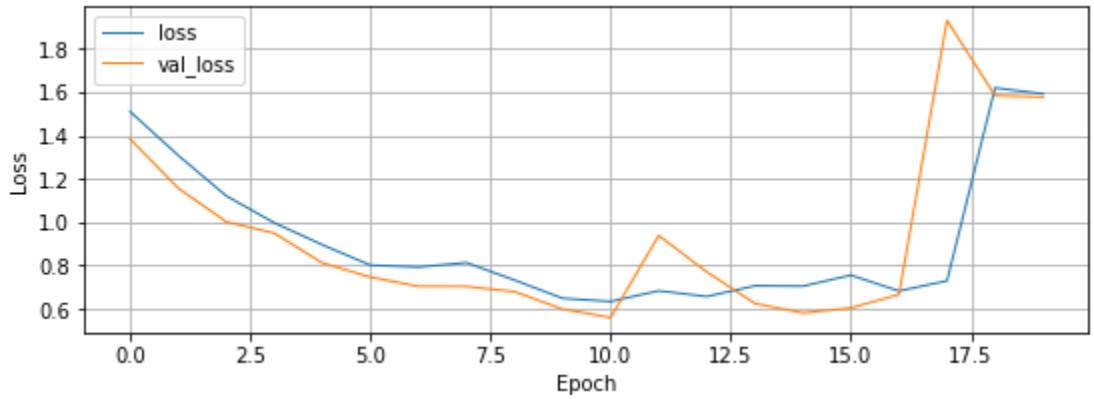

Figure A.4 Accuracy graph for LSTM

GRU:

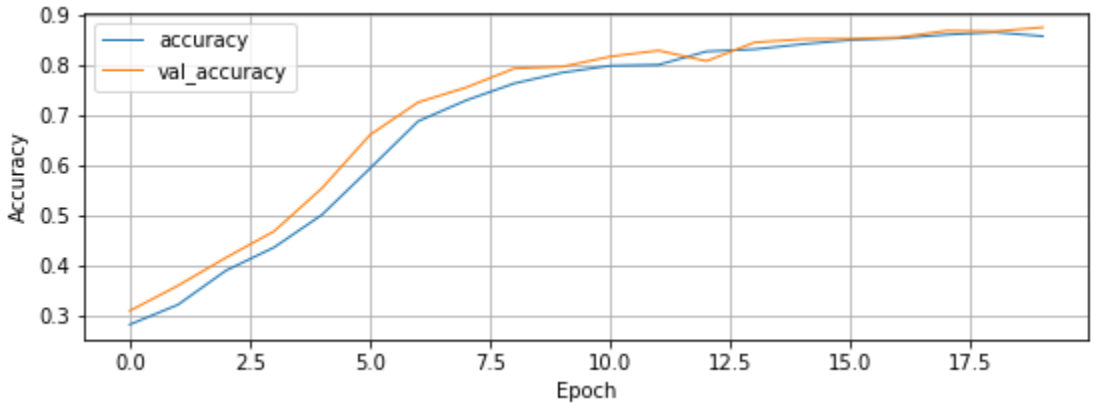

Figure A.5 Accuracy graph for GRU

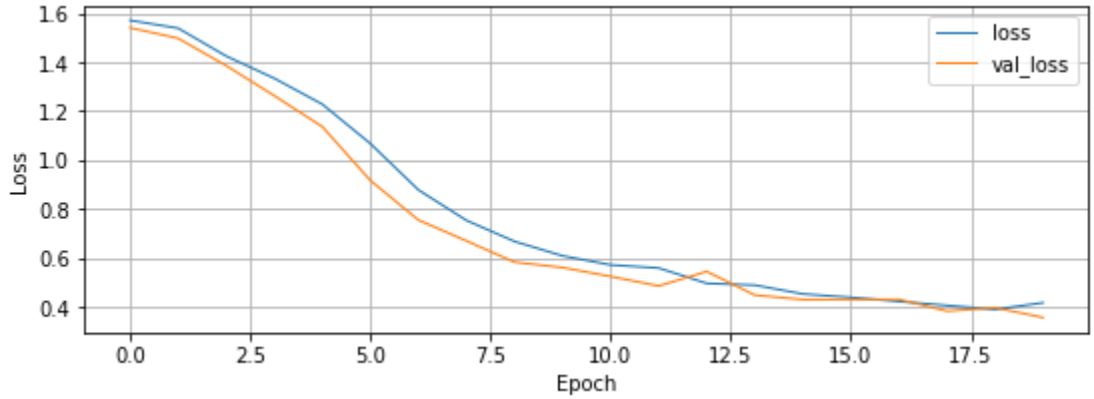

Figure A.6 Loss graph for GRU

Appendix B - Classification Report of the models

Classification report for the baseline model and three proposed deep learning models

KNN:

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.95	0.97	14993
1	0.92	0.99	0.95	1617
2	0.91	0.99	0.95	1427
3	0.50	0.91	0.65	531
4	0.94	0.95	0.95	1435
accuracy			0.95	20003
macro avg	0.85	0.96	0.89	20003
weighted avg	0.97	0.95	0.96	20003

Figure B.1 Classification report for KNN

CNN:

	precision	recall	f1-score	support
0	1.00	0.99	0.99	14921
1	0.99	1.00	1.00	1567
2	0.98	1.00	0.99	1517
3	0.89	0.98	0.93	516
4	0.97	1.00	0.98	1482
accuracy			0.99	20003
macro avg	0.97	0.99	0.98	20003
weighted avg	0.99	0.99	0.99	20003

Figure B.2 Classification report for CNN

LSTM:

	precision	recall	f1-score	support
0	0.98	0.88	0.93	14921
1	0.79	0.93	0.86	1567
2	0.57	0.92	0.70	1517
3	0.39	0.68	0.50	516
4	0.83	0.81	0.82	1482
accuracy			0.88	20003
macro avg	0.71	0.85	0.76	20003
weighted avg	0.91	0.88	0.89	20003

Figure B.3 Classification report for LSTM

GRU:

	precision	recall	f1-score	support
0	0.99	0.91	0.95	14973
1	0.93	0.98	0.96	1632
2	0.93	0.99	0.96	1462
3	0.33	0.90	0.48	504
4	0.86	0.97	0.91	1432
accuracy			0.93	20003
macro avg	0.81	0.95	0.85	20003
weighted avg	0.96	0.93	0.94	20003

Figure B.4 Classification report for GRU

Appendix C - Types of Arrhythmia and their QRS interpretation

Sinus arrhythmia originates in the SA node. Some of the major observed symptoms are increased or decreased Heartbeat. If we don't have that consistent regular heart rate then most likely it is sinus arrhythmia.

Premature Atrial Complex (PAC) originates in the SA node, essentially in the atria. Some beats come a little bit earlier than in the rhythm. These types of arrhythmia often occur due to some sort of irritation of the atria, which increases the automaticity. Could be observed by having narrow QRS, Upright P wave, and different morphology.

Premature Junctional Complex (PJC): These types of arrhythmias are very rare. Mainly caused due to the irritation of the AV junction. Usually observed by narrow QRS and Absent or Inverted P wave.

Premature Ventricular Complex (PVC): These types are very common, and caused due to the irritation of the ventricles. Observed by wide QRS and sometimes having no P wave and occurrence of T wave opposite to R wave.

Sinus Bradycardia (SB): This type of arrhythmia has lower HR < 60. Can be observed by narrow QRS, and usually upright P waves. Though SB is normal for Athletes, or slowed nervous system, can also be observed due to some medications.

Sinus Tachycardia (ST): HR is usually greater than 100. Has narrow QRS, and upright P wave. Could be due to increased sympathetic responses like pain, fever, increased O2 demand, etc.

SupraVentricular Tachycardia (SVT): HR is usually greater than 150. It has indistinguishable P waves and narrow QRS. Usually, there is an increased sympathetic response.

Atrial Fibrillation (Afib): Usually caused due to the damage to the structure of the heart, frequent in elderly patients. Has no defined structural pattern of morphology nor atrial kick. Very irregular and chaotic waves with narrow QRS and sometimes absent P waves.

Atrial Flutter (Aflutter): Usually has sawtooth P waves, and flutter waves greater than 250. Observed by narrow QRS.

Junctional: Originates from AV junction. HR 40-60. Observed by narrow QRS or absent or inverted P wave.

Accelerated Junctional usually has HR around 60-100 while **Junctional Tachycardia** has HR greater than 100. **Idioventricular(IVR):** Usually has wide QRS and Absent P waves with HR around 20-40 **Accelerated Idioventricular Rhythm (AIVR)** has an observed HR 40-100

Ventricular Tachycardia: These types of arrhythmias are usually monomorphic (one kind of morphology) and could be potentially lethal. Usually observed with wide QRS and no p wave, with HR greater than 100.

Heart Blocks: The heart blocks in arrhythmia are of three types namely first degree, second degree, and third-degree atrioventricular block. The third-degree AV block is referred to as a complete heart block. They usually have more P waves compared to theQRS complexes and no connection exists between them. HR is around 30-40.

Bibliography:

1. International Cardiovascular Disease Statistics 2019
2. Belkadi, M.A., Daamouche, A., Melgani, F., 2021. A deep neural network approach to QRS detection using autoencoders. Expert Syst. Appl. 184, 115528. https://doi.org/10.1016/j.eswa.2021.115528
3. DeepHeart: Semi-Supervised Sequence Learning for Cardiovascular Risk Prediction [WWW Document], 2018. . DeepAI. URL https://deepai.org/publication/deepheart-semi-supervised-sequence-learning-for-cardiovascular-risk-prediction (accessed 12.5.21).
4. Guglin, M.E., Thatai, D., 2006. Common errors in computer electrocardiogram interpretation. Int. J. Cardiol. 106, 232–237. https://doi.org/10.1016/j.ijcard.2005.02.007
5. Haase, N., n.d. International Cardiovascular Disease Statistics 14.
6. Hannun, A.Y., Rajpurkar, P., Haghpanahi, M., Tison, G.H., Bourn, C., Turakhia, M.P., Ng, A.Y., 2019. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. Nat. Med. 25, 65–69. https://doi.org/10.1038/s41591-018-0268-3
7. Hong, S., Zhou, Y., Shang, J., Xiao, C., Sun, J., 2020. Opportunities and challenges of deep learning methods for electrocardiogram data: A systematic review. Comput. Biol. Med. 122, 103801. https://doi.org/10.1016/j.compbiomed.2020.103801
8. Hong, S., Zhou, Y., Wu, M., Shang, J., Wang, Q., Li, H., Xie, J., 2019. Combining deep neural networks and engineered features for cardiac arrhythmia detection from ECG recordings. Physiol. Meas. 40, 054009. https://doi.org/10.1088/1361-6579/ab15a2cessed 12.5.21).
9. LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539
10. Li, R., Zhang, X., Dai, H., Zhou, B., Wang, Z., 2019. Interpretability Analysis of Heartbeat Classification Based on Heartbeat Activity's Global Sequence Features and BiLSTM-Attention Neural Network. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2933473
11. Liu, P., Sun, X., Han, Y., He, Z., Zhang, W., Wu, C., 2022. Arrhythmia classification of LSTM autoencoder based on time series anomaly detection. Biomed. Signal Process. Control 71, 103228. https://doi.org/10.1016/j.bspc.2021.103228
12. Saadatnejad, S., Oveisi, M., Hashemi, M., 2018. LSTM-Based ECG Classification for Continuous Monitoring on Personal Wearable Devices.
13. Wang, P., Hou, B., Shao, S., Yan, R., 2019. ECG Arrhythmias Detection Using Auxiliary Classifier Generative Adversarial Network and Residual Network. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2930882
14. Wołk, K., Wołk, A., 2019. Early and Remote Detection of Possible Heartbeat Problems With Convolutional Neural Networks and Multipart Interactive Training. IEEE Access PP. https://doi.org/10.1109/ACCESS.2019.2919485
15. Xia, Y., Xie, Y., 2019. A Novel Wearable Electrocardiogram Classification System Using Convolutional Neural Networks and Active Learning. IEEE Access 7, 7989–8001. https://doi.org/10.1109/ACCESS.2019.2890865
16. Zhu, F., Ye, F., Fu, Y., Liu, Q., Shen, B., 2019. Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network. Sci. Rep. https://doi.org/10.1038/s41598-019-42516-z