Anomalies in The Classification and Detection of Heart Rate Using Deep Learning

Sakshi Kishor

Dublin Business School

This dissertation is submitted for the degree of

MSc Data Analytics

October 2020

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Sakshi Kishor

October 2020

Acknowledgements

I would like to acknowledge my research guide, Dr. Amir Sajad Esmaeily, for his guidance and useful critiques throughout the research. My sincere thanks to all my lecturers for encouraging my interest in advanced technology growth.

Finally, I would like to thank my family and friends for their motivation and support.

List of Figures

Figure 1: ECG Waveform 8
Figure 2: Methodology Implementation 15
Figure 3: MIT BIH Data 19
Figure 4: Confusion Matrix 20
Figure 5: Design Flow 22
Figure 6: Process Flow 22
Figure 7: Wavelet Tree 23
Figure 8: Original Signal with coefficients 24
Figure 9: Filter 1- at node (1,0) 25
Figure 10: Filter 2- at node (2,0) 25
Figure 11: Filter 3- at node (3,0) 25
Figure 12: Filter 4- at node (4,0) 26
Figure 13: Filter 5- at node (5,0) 26
Figure 14: Denoised Signal 26
Figure 15: Categories of heart beats(Mathews, Kambhamettu and Barner, 2018) 27
Figure 16: ECG classified into 5 categories per beat 29
Figure 17: Applied CNN Model 29
Figure 18: Parameters and details of CNN 30
Figure 19: Confusion Matrix for CNN 32
Figure 20: Comparison of Precision 33
Figure 21: Comparison of Recall values 33
Figure 22:Comparison of Accuracy 34
Figure 23: Classification by Logistic Regression 35
Figure 24: Spectrum representation of the different categories of heartbeat 36
Figure 25: Number of categories in train and test data frames 36
Figure 26: Original and Smoothed signal 37
Figure 27: Processed waveform 37
Figure 28: Spectrum of the processed waveforms. 38
Figure 29: classification report before RBF 39
Figure 30: Count of each category in the data frame 39
Figure 31: Different categories of heartbeat in the dataset 40
Figure 32: Resampled data 40
Figure 33: Data frame after expanding dimension. 41
Figure 34: Data frame after one hot encoder 41
Figure 35: CNN model history 42
Figure 36:CNN ranking details 42
Figure 37: Comparative Analysis of the Models 43

List of Tables

Table 1: Comparison of models and their performance 12
Table 2:Logistic Regression-Evaluation Metrics Report 31
Table 3: Logistic Regression-Evaluation Metrics Report After Introducing Non-Linearity 31
Table 4: Evaluation Metrics Report for CNN 32

Table of Contents

1. Introduction 7
1.1 Classifications of work to describe the signal automatically 9
1.2 The CNN approach to classifying heartbeat characteristics 11
1.3 Requirements for further research in CNN 12
1.4 Comparative analysis of models based on previous researches 12
2. Methodology 14
2.1 CRISP-DM- 14
2.1.1 Business Understanding 15
2.1.2 Data Understanding 18
2.1.3 Signal Processing 19
2.1.4 Implementation of Models 19
2.1.5 Evaluation of Models 19
2.1.6 Deployment 21
2.2 Design Specification 21
3. Implementation 23
3.1 Signal Preprocessing 23
3.2 Logistic Regression 26
3.3 Convolution Neural Networks (CNN) 28
4. Evaluation 30
5. Results and Discussion 34
5.1 Results from Machine Learning Model- 34
5.2 Results from Deep Learning Model- 39
5.3 Tableau Results- 43
5.4 Discussion- 44
6. Conclusion and Future Work 44
References 44

ABSTRACT

Arrhythmias, which in non-medical terms is commonly known as the irregularity in a heartbeat, contributes to approximately 300,000 sudden deaths per year. This incidence rate is higher than the other lethal and dangerous disease like lung cancer, stroke, or a breast cancer. Cardiologists can help patients cope with many types of arrhythmias, but researchers and doctors are kept awake at night by sudden and lethal arrhythmias. This study is an effort to bring down the incidence rate due to heart rate arrhythmias by identifying the irregularities in a heartbeat at early stages and help the cardiologists with their patients. The 5 different categories of heartbeats are classified in this study from publicly available MIT-BIH database which consists of ECG data from various patients. The data has been preprocessed with the help of wavelet transformation tool in MATLAB. To help achieve the classifications of the heartbeats, logistic regression and convolution neural networks algorithms are used. These two models' evaluations are compared in the form of recall, precision, accuracy, and computational time. A comparative study between the two models is done with the help of tableau. Based on the evaluation, the deep learning model, CNN, gives out the accuracy of 99.09 percent.

Keywords: Arrhythmia, CNN, Logistic Regression, Wavelet Transform

1. Introduction

The cardiovascular system in a human being consists of arteries and veins which helps the heart to pump the blood throughout the body. For a healthy person, the heart beats 35 million times in a year. When a heart performs any deviation from this regularity or the usual sequence of the electrical impulses, then this anomaly is called as arrythmias. Because of this, the other major organs of a human body cannot perform as it is supposed to and can shut down which can lead to sudden death. Some of these anomalies are benign, but some are life threatening. Arrhythmias can be caused by a fever, emotional or physical stress, genes, or the consumption of drugs and other stimulants.

To understand this research work, it is important to understand how a human heart works. There are four chambers (two upper and two lower) in a normal heart of a human being. These chambers should beat properly for the heart to function in a systematic manner. When an electrical pulse from the sinoatrial node (natural pacemaker) passes through it, the heartbeat (contraction) starts. The usual electrical sequence must be followed to properly pump the heart. The heart pumps and beats at a regular pace if the electrical impulse is transmitted normally. An average heart beats 60 to 100 times a minute in an adult. A painless, non-invasive technique that tracks the electrical activity of the heart and may help diagnose arrhythmias is electrocardiography (ECG or EKG). Arrhythmias can be completely harmless, or it can be life threatening. Arrhythmias, such as a transient pause of the heartbeat or a premature beat, are so brief that they do not adversely effect the overall heart rate or its rhythm. But if these irregularities last longer then the heart will pump less efficiently. More information on arrhythmia available on website[1].

Electrocardiogram signals can differ gradually as the data related to the heart diseases increases which in turn can make the researcher's work harder. Figure 1 represents an ECG signal which is periodic in nature.

[1] https://www.heart.org/en/health-topics/arrhythmia/about-arrhythmia

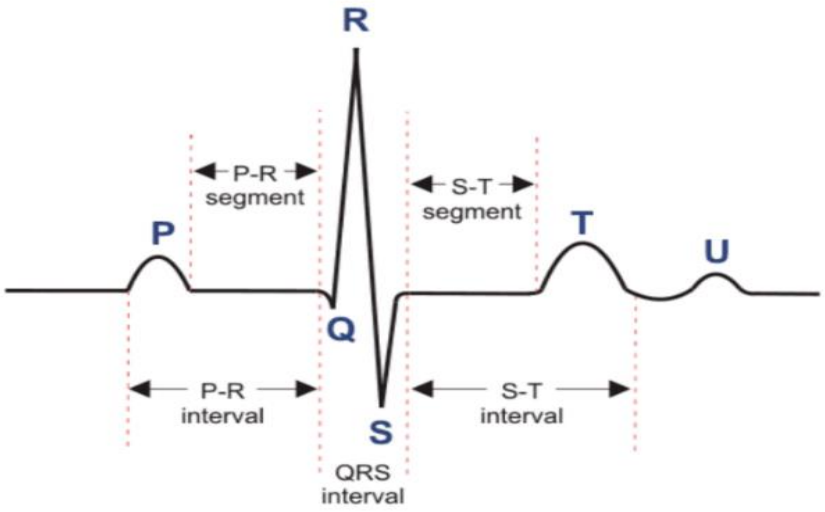

Figure 1: ECG Waveform

Since the increase in technologies has increased the human health risk factor, there are lots of research which are performed by using the algorithms of machine learning and with the help of data mining to identify these signals. Hear problems are increased drastically in recent decades due to the increasing pollution, stressful life and eating habits. So, an early identification of the heartbeat signals will help cardiologists to bring down the death rate and to also reduce the medical cost.

So, the below question arises which can help in detecting the heart rate arrhythmias for a better treatment at an early stage.

Can the arrhythmias in heartbeat signals be detected and categorized by deep learning and machine learning approaches to decrease efforts to analyze the signal for improved treatments?

There are various heartbeat categories: N, S, F, V, and Q, which is used in this study to identify and classify the heartbeats. With the help of models of machine learning and deep learning, it is less time consuming to identify the issues in a patient and to decrease a cardiologist's manual efforts or intervention. This study will also help a doctor's effort to analyze an ECG report.

One of the common research subjects in recent medical fields is heartbeat classification. Detecting these irregularities of heartbeat, which is called as arrhythmias, has caused difficulties to the doctors in day to day life as it is rising rapidly. It takes time to identify the irregular signals in conventional ways and needs some new methodologies and new methods to interpret the signals and detect whether there are any signal abnormalities. To identify the signals, there are several machine learning techniques such as SVM, KNN, GMM and so on that have been implemented on the ECG signals. In recent years, researchers have advanced the application of deep learning techniques to construct a more precise and effective model. The deep neural network is a component of machine learning, one of the emerging methods for analyzing and extracting huge dataset features. It is a subset of machine learning that extracts from the raw data high-level features in multiple layers. Researchers are more keen to work in this area as the deep neural network's popularity has increased drastically in biomedicine. This literature review will also present the previous work and research being done. The literature review has different algorithms for detecting the arrhythmias and the procedures for the results they have achieved are also described with the comparison table.

1.1 Classifications of work to describe the signal automatically

(Ye, Kumar and Coimbra, 2012) The study was conducted by categorizing the work into two intra patient paradigms, which are also known as class-oriented and inter patient-oriented as subject-oriented (Martis, Acharya, Prasad, Chua, & Lim, 2013) using intra patient paradigm, which is based only on the data mark. (de Chazal, O'Dwyer and Reilly, 2004) Inter patient paradigms were suggested to adapt to the more realistic circumstances, constructing the model in a very widespread manner with various ECG recordings taking the training and testing data type. This proposed paradigm is more practical and 83% and 88.1%, respectively, will achieve high accuracy and sensitivity. Hybrid paradigm, which is called patient specific, was proposed (de Chazal and Reilly, 2006), the model was developed by training global classifier and then the local classifier is (Shi *et al.*, 2019) taken to tuned global classifier. This showed 97.4 percent accuracy and 94.4 percent sensitivity. Surveys of (Luz *et al.*, 2016) on the research work performed using various algorithms to predict the anomalies and identify the signals. Using XGBoost, which is an effective machine learning technique that generalizes the data, brings regularity in controlling the complex model, the result is obtained in 3 steps of data preprocessing, extracting the features and hierarchical classification. (Shi *et al.*, 2019) have predicted the model. The extraction of features uses wrapper methods to train the model and delete features with the smallest scores. Feature extraction plays a major role in the classification outcome when classifying the signals. There are several other features that define the heartbeat, in addition to the extraction of RR-interval (Sannino and de Pietro, 2018) (Shi *et al.*, 2019). It includes morphological characteristics (de Chazal, O'Dwyer and Reilly, 2004) statistical characteristics (Mar *et al.*, 2011) interval characteristics, higher order statistical characteristics and vector cardiogram (Soria and Martínez, 2009). Moreover, linear prediction characteristics (Martis *et al.*, 2012) ,wavelet packet entropy (Liu *et al.*, 2016) are also proposed. By achieving an overall precision of 92.1 percent, the four classes N, S, V and F were segregated. In the past 2 decades, the principle of classification and prediction of heartbeat signals has been studied and evaluated by carrying out research and applying various algorithms. DNN (Deep Neural Network) technique was used (Sannino and de Pietro, 2018) with more than 3 layers for extracting and analyzing features and classifying the signals. The tensor flow structure is used, which is a Google library and has 5, 10, 30, 50, 30, 10, 5 neurons that make up seven layers. The signal is processed in four stages: 1. Denoising with a cut-off frequency of 35Hz using two median filters and 12-order low pass filters. 2. Peak detection is conducted using the toolbox of the waveform database available in MATLAB, which is very useful in detecting the QRS peaks in heart signals. 3. Segmentation of the signal in which the signals are segmented into one heartbeat. 4. Temporary function extraction that extracts the beats into the average interval of pre-RR, post RR interval, local and global RR. The results obtained by applying this method are then compared with other commonly used classifiers such as Naïve Bayes, Ada Boost, Random tree, SVM etc. for precision, specificity, and sensitivity. DNN resulted in an accuracy of 99 percent relative to all other algorithms. (Sannino and de Pietro, 2018) offers the enhancement that can be accomplished in a patient-specific paradigm, as each patient has a specific model, the number of models and the number of patients is equivalent. It produces redundancy and does not carry optimization and the generalization is poorly illustrated. Therefore, GRNN (Global Recurrent Neural Network)(Wang *et al.*, 2019) ,an improved version of RNN, was used to identify the beats as it uses a single model for all patients, generalizes the whole system and learns the distinctions between different groups. The process is based on 4 steps: System Framework, Model Classification, Framework Building

and System Implementation. They used the morphological vector and premature-or-Escape-Flag to automatically evaluate the characteristics. It uses three MIT-BIH, SVDM and INCARTDB data sets and applies an active learning setup that is a machine learning algorithm to retrain a model where the unlabeled data pool is rated and the labeling efforts are reduced. Training and testing data were collected from various databases and analyzed for GRNN0, GRNN1, GRNN2 and GRNN3, for a total of 12 experiments. The results obtained are 95.4 percent accurate and higher than other algorithms for machine learning. New deep learning techniques such as auto encoder have been used (Baloglu *et al.*, 2019) to decrease the size of arrhythmic beats and the CAE-LSTM (Long Term Sort Term Memory) method, which decreases the time to analyze the beats. Using CAE models, the input signals are compressed to 16 layers of CAE models by a two-part encoder and decoder. This will compress the input signal into small encoded features so that the analysis becomes simple. The function size is compressed to 32 * 1 from 260 * 1 sampled signals on the encoder part and the decoder uses the encoder part to recreate the original signals. The main objective of this system is to reduce the minimum loss size of the signals and develop the model on the LSTM network. Compression of signals and extraction of features requires full pooling, which is then normalized by the normalized layer of the batch. The experiment produced a 99.42 percent accuracy, which demonstrates that the model is highly efficient in classifying the signals. For greater precision, RBM (Restricted Boltzmann Machine) and DBN (Deep Belief Network) are used (Mathews, Kambhamettu and Barner, 2018) and two kinds of VEB (Ventricular ectopic beats) and supraventricular ectopic beats (SVEB) heartbeat anomalies are identified. There are 3 phases: Data pre-processing, segmentation, and extraction of features. It uses the very common MIT-BIH data base database. The pre-processing of data is carried out using a 12-tap low pass filter and a finite impulse response filter and RLS (Recursive Least Square) that is effective in eliminating the signals' motion artifacts. The signals were sampled down to 114Hz and used two feature extraction methods, one with 26 features and the other with 22 features. At a lower sampling rate, the system returned an accuracy of 96.94 (for VEB) and 93.78 (for SVEB) at a lower sampling rate. This proved to be better technique as compared to LDA (Linear Discriminant Analysis) , QDA (Quadratic Classifier), ANN (Artificial Neural Network) etc. A new approach to categorize the heartbeat is proposed by (Dong, Wang and Si, 2017) in which signals are enabled by a learning algorithm which are modelled and easily recognized. The method is carried out in two stages, first by modeling the ECG signals and secondly by classifying the rhythm. Instead of static features, it focuses more on dynamic features and uses beat dynamics, which is a special aspect in the classification of beats. For 5 beat classification forms, the system was able to generate 97.78 percent using 5 percent as the training set of results. Proposed dictionary learning algorithm to identify the signals by using k-medoids cluster optimized by k-means++. (Liu *et al.*, 2016) The algorithm is applied to the characteristics of vector quantization and the results are contrasted with the function of Fourier transformation, the function of discrete wavelet transformation, and the function of sampling point. (Dutta, Chatterjee and Munshi, 2011) uses ANN as a classifier with a correlation approach to identify the signals to construct a model. This makes use of LVQ (Learning Vector Quantization) to classify the signals into 3 forms of regular, PVC (Premature Ventricular Contraction) and other beats under supervised clustering. To evaluate the resemblance between the two signals, the cross-correlation technique is used. Using the MIT-BIH dataset, it achieved an overall precision of 95.24 percent.(Park and Kang, 2014) Demonstrate the characterization of heart signals using the Pan-Tompkins algorithm to accurately extract features such as QRS and P signals and to identify each rhythm, the decision tree applies. To have reliable performance, this algorithm applies the low pass and high pass filter differentiator, integrator, and cascades. The accuracy obtained using the MIT-BIH dataset is 97.06 percent.

1.2 The CNN approach to classifying heartbeat characteristics

(Sellami and Hwang, 2019) It uses the renowned MIT-BIH database to provide a highly effective and comprehensive method of implementing heartbeat signals taken from a single lead, which provides clarification for the classification of heartbeats and is checked twice by independent experts in R-peak locations. Therefore, the author says that preprocessing techniques are not required to eliminate noise in the signal and extract features. Heartbeats (N, V, S, F and Q) are divided into 5 groups. With each containing 64k kernels, the model has 9 convolutional layers. It takes longer to learn the 9-layer CNN structure, so batch normalization is used to increase the training time. By analyzing the raw data without any preprocessing techniques and generating reliable results, the result produces 99.79 percent-99.52 percent accuracy, which is a decent display and computationally inexpensive. Advances through the usage of deep learning techniques such as Convolutionary Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU) (G, K P, & R, 2018) The model has been analyzed by applying the Keras and TensorFlow method to the combination of algorithms such as CNN-RNN, CNNGRU, CNN-LSTM (Oh *et al.*, 2018) applied to the MIT-BIH database. Using back propagation through time (BPTT), the data is trained, and the only procedure carried out on the data is to remove the heartbeat from its sequence. An accuracy of 83.4 percent was achieved by the combination of different algorithms. (Hasan and Bhattacharjee, 2019a) Says that CNN offers high recognition for all lead signals, it uses CNN to detect myocardial infraction automatically, suggesting heart attack using a publicly accessible PTB (Physiobank) ECG database dataset. The method takes three steps: preprocessing of data, segmentation of data and extraction of features.

It is a 10-layer deep hierarchical structure with maximum pooling layers and is prevented by dropout technique over the required period of model training. On Ubuntu (16.04.4) the experiment is performed with the Keras deep learning library. It yielded a 99.78 percent accuracy. To decompose the signal, instead of using raw signal, CNN with EMD (Empirical mode decomposition) and IMF (Intrinsic Mode Functions) (Hasan and Bhattacharjee, 2019a) uses CNN. The PTB Diagnostic and MIT-BIH database combinations are used for the study. With the introduction of CNN technology, the features using EMD are trained in 1D (Yildirim *et al.*, 2019) as CNN provides more performance, non-complex, rapid, and easy to use. The author notes that it becomes hard to distinguish the beats due to noise present in P, Q, S, T peaks, but decomposing technique makes it easy to separate Q, R and S peaks. The experiment is performed between the layers by taking the 1st three MF 'and applying max pooling and batch normalization techniques. This resulted in an overall precision of 99.71 percent, showing that this model is better than other models. The relationship between a person's vocal sounds and his pulse rate via vocal response measurement was demonstrated (Ankışhan, 2019). The experiment involved the use of techniques from CNN, SVM and MLR. To divide them into parts, the signals are first normalized, windowed, and smoothed, and features are selected by applying various algorithms and the results are compared and checked. SVM 's accuracy is about 98.70%, CNN is 98.92% and MLV is about 96.88%. This indicates that, as opposed to other models, the CNN model has high precision.

1.3 Requirements for further research in CNN

CNN is one of the most common deep learning techniques for extracting features and obtaining high accuracy and sensitivity (Camacho *et al.*, 2018). This is one of the key reasons why, in recent times, deep learning is so common. However, CNN learning takes place with the layers that are right at the start to detect the basic characteristics and the layers that are deep within can detect the complex characteristics, but there are several drawbacks that need to be addressed. (Ha and Jeong, 2019) The shortcomings lie here because there is no spatial information in the model and lead to the loss of the most significant information since the next layer is transferred to only the active neurons. The capsule neural network model was proposed to solve this problem. The spatial information and the likelihood of the object present will be considered by this neural network. This refers to the routing method where the link between one layer and the other is vector to vector, while the link is scalar in CNN. Unlike the noise-sensitive CNN model, capsules perform well under noise conditions. In medical areas, capsule networks are used. To predict Alzheimer's disease, which is a brain-related problem of memory loss in elderly individuals using MRI data, the capsule neural network was used (Kruthika, Rajeswari and Maheshappa, 2019). Capsule networks have been introduced (Sajjad *et al.*, 2019) to automatically classify gastrointestinal disorders. The capsule network is used in the multi-grain brain tumor system classification (de Jesus *et al.*, 2018). Therefore, for the classification of the signals and prediction for the study, the project tends to implement the Capsule neural network model.

1.4 Comparative analysis of models based on previous researches

Table 1: Comparison of models and their performance

Article	Algorithms used	Dataset	Features	Methodology	Software used	Accuracy
(Shi *et al.*, 2019)	XGBoost	MIT-BIH	RR-interval, Morphological feature, statistical feature, Higher order statistics	KDD	R Program	92%
(Dutta, Chatterjee and Munshi, 2011)	ANN and LVQ	MIT-BIH	Premature Ventricular Contraction	CRISP DM	Matlab, R program Python	95.24%
(Park and Kang, 2014)	PanTompkins algorithm	MIT-BIH	QRS and P	CRISP-DM	R program, python	97.06%

(Ha and Jeong, 2019)	Genetic Algorithm Back Propagation Neural Network	MIT-BIH	RR-interval, QRS interval, P, Q and S	CRIPS-DM	R program, python	98.45%
(Ankışhan, 2019)	CNN – EMD, IMF (Intrinsic Mode Functions)	PTB Diagnostic and MITBIH	RR-interval, Morphological feature, statistical feature, Higher order statistics	KDD	Keras and TensorFlow framework	99.71%
(Baloglu *et al.*, 2019)	CNN	PTB (Physioban k)	Ventricular ectopic beats, supraventricular ectopic beats	CRISP-DM	Ubuntu (16.04.4) with Keras deep learning library, R program	99.78%
(Sannino and de Pietro, 2018)	CNN	MIT-BIH	RR-interval, QRS interval	KDD	MATLAB, R Studio	99.79% – 99.52%
(Hasan and Bhattacharjee, 2019b)	CNNRNN, CNN - LSTM, CNN- GRU	MIT-BIH	RR-interval, QRS interval, P, Q and S	KDD	Keras and TensorFlow framework	83.4%
(Sannino and de Pietro, 2018)	DNN	MIT-BIH	RR-interval, Post RR-interval, local and global RR-interval	CRISP-DM	Tensor Flow Framework, MATLAB, R program	99%
(Wang *et al.*, 2019)	GRNN	MIT-BIH, SVDM, INCART D B	Morphological vectors, Escape flag	KDD	R program, python	95.4%
(Yildirim *et*	CAELSTM	MIT-BIH	RR-interval, QRS interval	KDD	Matlab, version R2014b	99.4%

al., 2019)					(8.4.0.150 421)	
(Liu *et al.*, 2016)	dictionary learning algorithm	MIT-BIH	QRS and R	KDD	Ubuntu (16.04.4) with Keras deep learning library, R	98.89%
(Dong, Wang and Si, 2017)	Determinist ic learning algorithm	MIT-BIH	P, Q, S and T	SEMMA	Matlab, Rprogram	97.78%
(Mathews, Kambhamettu and Barner, 2018)	RBM and DBN	MIT-BIH	Ventricular ectopic beats, supraventricular ectopic beats	CRISP-DM	Matlab, Rprogram and Python	93.78%

The studies done indicates that there is a huge research done in the field of medical science and many algorithms have been suggested to identify the signals. Recent advances in the field of machine learning and the implementation of techniques for deep learning have enhanced the precision, sensitivity, and specificity of signal classification. Together with the combination of other models, the CNN implementation has shown a high advantage and an excellent response but addresses some problems. This problem needs to be tackled and some adjustments must be presented to the model. Therefore, another enhanced model of Convolutionary Neural Network, is required to address CNN's disadvantages.

2. Methodology

2.1 CRISP-DM-

To plan, organize, and implement this project, the cross industry standard process for data mining process is used.

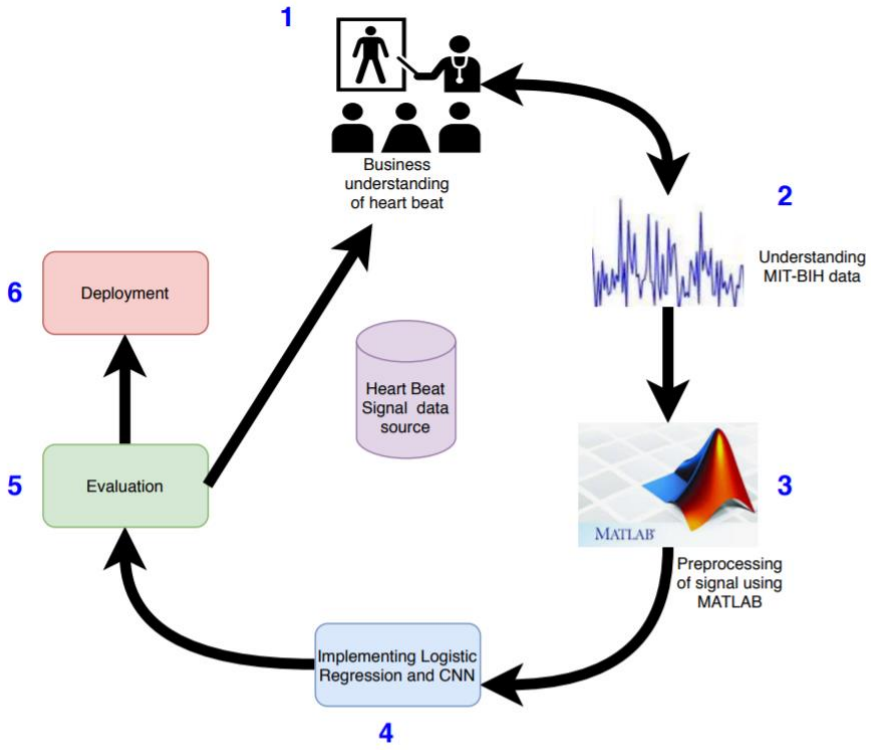

Figure 2: Methodology Implementation

2.1.1 Business Understanding-

The need of heartbeat classification has become suddenly important because of the increasing rate of heart diseases. The cardiologists will get help in providing an early analyzation of the patients to identify the criticality of the patient if the heartbeats are identified at an early stage.
Its better to understand the different forms of peak in a heartbeat to determine the locations of all the peaks of interest found in the signal from the ECG. The p-wave and t-wave locations are beneficial for the segmentation of the signal. Instead, the identification of r-peaks is important for the extraction of temporal features. Automated interpretation algorithms are used in most current ECG machines. Features such as the PR interval, QT interval, corrected QT (QTc) interval, PR axis, QRS axis, rhythm and more are measured in this analysis. Until checked and/or updated by expert interpretation, the findings from these automated algorithms are considered 'preliminary'.
The typical electrocardiogram (ECG) consists of many different waveforms that, in different parts of the heart, reflect electrical events during each cardiac cycle. Beginning with the P wave, ECG waves are numbered alphabetically, followed by the QRS and ST-TU complexes (ST segment, T wave, and U wave). The cross point between the beginning of the ST segment and end of the QRS complex is called as J point.

P Wave- Atrial depolarization reflects the P wave (refer to figure 1). The normal sinus P wave shows depolarization from the right to the left atrium and is a positive initial low amplitude deflection that precedes the QRS complex in most leads. In general, the length is <0.12 sec and the amplitude is <0.25

mv. The P wave is often notched in the limb leads and normally biphasic in lead V1 since the right atrial depolarization precedes that of the left atrium (as the sinus node is in the right atrium). The initial positive deflection in V1 is due to anteriorly guided right atrial depolarization, while the second negative deflection reflects posteriorly directed atrial depolarization. The series of atrial repolarization (atrial ST and T wave phases) begins a little before, during, and just after ventricular myocardial depolarization. The atrial "T wave" itself is normally concealed and not detected on the regular ECG by the QRS complex. Furthermore, the amplitude of the atrial T wave is typically too small for normal gain to be detected.
The PR interval is shortened when the heart rate is increased and there is an increased sympathetic tone; atrial repolarization (the atrial T wave) can often be observed at the very end of the QRS complex, altering the J point, resulting in J point depression with rapidly increasing ST segments, particularly during the first 80 msec after the QRS complex. This is a biological finding, but it can be confused with true ST depression, creating a false positive reading. Clinically, during acute pericarditis, atrial repolarization (the atrial ST phase) is most noticeable, in which one often sees PR segment elevation in lead aVr and PR segment depression in the infero-lateral leads, indicating an injury atrial current. In some cases of high degree AV block, especially when the atria are enlarged, the low amplitude atrial T wave may also be unmasked. Finally, with other pathologies, such as atrial fraction or atrial tumor invasion, alterations can occur in the atrial ST segment and T wave.

PR interval- In the PR interval, the P wave and the PR section are also included (refer to figure 1). The first component of the QRS complex is determined from the beginning of the P wave to (which may be a Q wave or R wave). This involves time and conduction through the AV node and the His-Purkinje mechanism for atrial depolarization (the P wave) (which constitutes the PR segment). With heart rate, the length of the PR interval varies but is usually 0.12 to 0.20 sec. Due to sympathetically mediated enhancement of atrioventricular (AV) nodal conduction, the PR interval is shorter at higher heart rates; it is longer when the rate is slowed as a result of slower AV nodal conduction resulting from sympathetic tone withdrawal or a rise in vagal input.

QRS complex- The QRS complex is the time for depolarization of the ventricles (refer to figure 1).

- It is called a Q wave if the initial deflection is negative. As a result of initial septal depolarization, small Q waves are often seen in I, aVL, and V4-V6 leads and are deemed natural.
- The QRS complex's first positive deflection is called the R wave. It represents a depolarization of the myocardium of the left ventricle. Right ventricular depolarization is blurred since the myocardial mass of the left ventricle is much higher than that of the right ventricle. Initial septal depolarization reflects the tiny R wave in lead V1.
- The S wave, which reflects terminal depolarization of the high lateral wall, is the negative deflection following the R wave.
- If a second positive deflection is present, it is known as a R '.
- For comparatively small amplitude waves of less than 0.5 mV, lower case letters (q, r, or s) are used (less than 5mm with standard calibration).
- A QS wave is when the QRS complex is negative completely.

Normally, the entire QRS cycle lasts for 0.06 to 0.10 seconds and is not affected by heart rate.

The R wave should advance in size over the V1-V6 precordial leads. A small R wave with a deep S wave is usually present in lead V1. Until V4-V6, the R wave amplitude can increase in size while the S wave becomes less profound. This is called the progression of the R wave through the precordium.

ST segment – The ST section appears at the end of ventricular depolarization and before the beginning of repolarization (refer to figure 1). It's a moment of silence in electrocardiography. The cross point between the beginning of the ST segment and end of the QRS complex is called as J point.

This segment is typically isoelectric, i.e., zero potential with a slight upward concavity, as defined by the T-P segment. However, depending on the associated disease states, it may have other configurations (e.g., acute myocardial infraction, or pericarditis). In these cases, with an upsloping, horizontal, or downsloping morphology, the ST segment may be flattened, depressed (below the isoelectric line) or raised in a concave or convex direction (above the isoelectric line).

The J point is depressed in certain normal cases (such as sinus tachycardia), and the ST segment rapidly upslopes, becoming isoelectric within 0.08 seconds after the end of the QRS complex.

T wave- The T wave is large, has a slow upstroke and a faster downslope to the isoelectric line following its peak, because the rate of polarization is slower than the depolarization. It represents the ventricular repolarization time (refer to figure 1). The T wave is asymmetric, therefore, and the amplitude is variable. The T wave, however, is usually smooth up and down. A superimposed P wave should be considered if there is some irregularity on the T wave (bump, notch, ripple, etc.). Since depolarization starts at the surface of the endocardium and extends to the epicardium, the direction of ventricular repolarization begins at the surface of the epicardium and spreads to the endocardium. The T wave vector on the ECG is thus usually in the same direction as the QRS's main deflection. A further explanation states this is that the wave axes of QRS and T are usually consistent. Various disease states can contribute to the discordance of the T wave.

QT interval- The QT interval is made up of the complex QRS, the section ST, and the wave T (refer to figure 1). The QT interval is, therefore, primarily a ventricular repolarization scale. A more reliable indicator of ventricular depolarization is the JT interval, which does not include the QRS complex, although the QT interval is used in most clinical circumstances. This will lead to an increase in the QT interval if the QRS complex length is increased but does not represent a shift in ventricular repolarization. Therefore, if a prolonged QT interval is being measured, a widened QRS must be considered.

The interval of QT (or JT) is based on the heart rate; when the rate is slower, it is shorter at higher heart rates and longer. Therefore, based on Bazetts's formula, a QT interval that is corrected for the heart rate (QTc) has been classically measured. The formula below is taken from website[2].

$$QTc = \frac{QT\ interval\ in\ seconds}{\sqrt{cardiac\ cycles\ in\ seconds}} = \frac{QT}{\sqrt{RR}}$$

Although this method is clear, at extremes of heart rate it is inaccurate and results in overcorrection at high rates and under correction at low rates. It should be noted that the QTc is given in units of seconds or milliseconds with the Bazzet formula, requiring the denominator to be expressed as a quantity without units.

[2] https://litfl.com/bazett-formula/

In men, the upper normal value for the QTc is typically given as about <= 440 ms and in women as about <= 450 to 460 ms. However, due to the intrasubject variability in QTc intervals throughout the course of the day and intraobserver variability in the visual or electronic detection of the end of the T wave, the issue is further complicated. In addition, even subjects with congenital long QT syndrome can have QT/QTc values that are intermittently less prolonged, or even intermittently normal. Including normal forms and uncommon individuals with congenital short QT syndrome, the lower limits of the QTc are less well described.

Health professionals do have to be mindful of the fact that the QT interval would also increase since the QRS widens in the setting of a bundle branch block. The greater QT interval doesn't always represent a cardiac repolarization deformity, as the increase is due to a depolarization abnormality. There were not many examples of how to calculate the QT interval in the QRS widening setting.

In this project, we are classifying heartbeat into five different categories using deep learning and machine learning techniques. The category N or class 0 is the normal beat, category S or class 1 beat is super ventricular beat, class 2 or category V is premature ventricular beat , class 3 or category F implies fusion of ventricular beats and class 4 or category Q is classified as unknown beat.

2.1.2 Data Understanding-

Since 1975, their own study into arrhythmia analysis and related topics has been funded by laboratories at Boston's Beth Israel Hospital (now the Beth Israel Deaconess Medical Center) and at MIT. The MIT-BIH Arrhythmia Database, which was completed and started to be distributed in 1980, was one of the first major products of that project. This database has been used by different five hundred websites for various kinds of research. The database was originally distributed on 9-track half-inch 800 and 1600 bpi digital tape, and on quarter-inch FM analog tape in IRIG format. A CD-ROM version of the data was created in August 1989.

The MIT-BIH Arrhythmia Database comprises 48 half-hour extracts from 47 subjects recorded by the BIH Arrhythmia Laboratory between 1975 and 1979, from two-channel ambulatory ECG recordings. Twenty-three recordings were randomly selected from the dataset of outpatient electrocardiogram readings. The clinically relevant irregularities of heartbeat are included from the remaining twenty-five recordings from the same set because these recording might not be well represented in small samples.

Every record was separately annotated by two or more cardiologists; differences were resolved to obtain the computer-readable reference annotations included with the database for each beat (approximately 10 million annotations in all).

The dataset consists of heartbeat signals extracted from the database of MIT-BIH arrhythmia and the number of samples in the set is appropriate for a deep neural network model to be trained.

This dataset has been used to use deep neural network architectures to explore heartbeat classification and to observe some of the transfer learning capabilities on it. In the usual case, the signals refer to the electrocardiogram (ECG) forms of the pulse and the cases caused by multiple arrhythmias and myocardial infarction. These signals, with each segment corresponding to a heartbeat, are preprocessed and segmented. The data consists of 87554 rows which represents the total samples and there are 187 columns which represents the different categories of heartbeats i.e., N, S, V, F and Q.

Type of Beats →

Number of Samples ↓

	A	B	C	D	E	F
1	0.977941155433654	0.926470577716827	0.681372523307800	0.245098039507865	0.154411762952804	0.1911764
2	0.960113942623138	0.863247871398925	0.461538463830947	0.196581199765205	0.094017095863819	0.1253561
3	1.000000000000000	0.659459471702575	0.186486482620239	0.070270270109177	0.070270270109177	0.0594594
4	0.925414383411407	0.665745854377746	0.541436493396759	0.276243090629577	0.196132600307464	0.0773480
5	0.967136144638061	1.000000000000000	0.830985903739929	0.586854457855224	0.356807500123977	0.2488262
6	0.927461147308349	1.000000000000000	0.626942992210388	0.193436965346336	0.094991363584995	0.0725388
7	0.423611104488372	0.791666686534881	1.000000000000000	0.256944447755813	0.000000000000000	0.2777777
8	0.716814160346984	0.539822995662689	0.283185839653015	0.129793509840965	0.064896754920483	0.0766961
9	0.874213814735412	0.849056601524353	0.480083853006362	0.058700211346149	0.090146750211716	0.3102725
10	1.000000000000000	0.996086120605468	0.694716215133666	0.336594909429550	0.238747552037239	0.2681017

Figure 3: MIT BIH Data

2.1.3 Signal Processing-

To eliminate the powerline and base-line interference, the data from MIT-BIH is extracted and then with the help of wavelet transformation in MATLAB, the data is preprocessed. In order to analyze and synthesize signals and images, Wavelet Toolbox provides functions and apps. The toolbox includes wavelet continuous analysis algorithms, wavelet coherence, synchrosqueezing, and time-frequency data-adaptive analysis. The toolbox also includes apps and functions, including wavelet packets and dual tree transforms, for decimated and nondecimated discrete wavelet analysis of signals and images. We can analyze signals and images at various resolutions using discrete wavelet analysis to identify shift points, discontinuities, and other events not readily apparent in raw data. On multiple scales, we can compare signal statistics, and perform fractal data analysis to discover hidden patterns. We can also obtain a sparse representation of data with the Wavelet Toolbox, useful for denoising or compressing the data while retaining important characteristics.
In this project, by applying the DWT technique, the signal is de-noised at 5 speeds. The final output is received which is filtered with the help high pass signals using DWT.

2.1.4 Implementation of Models-

The two models i.e., machine learning and deep learning models are fed with the preprocessed and clean data. These models are then used to predict the different types of irregularities present in the signal by classifying the signal into different categories. The data which is received after the preprocessing of the signal will be split into training and testing data in 70:30 ratio while implementing the model further. For machine learning model, algorithms of the logistic regression are used and for deep learning model, algorithms of CNN are used.

2.1.5 Evaluation of Models-

The evaluation of the applied models is carried out to assess the model 's efficiency. The parameters like accuracy, recall, F1-measure, precision, and computational time are used to evaluate the performance of

both the models. Figure 4 represents the confusion matrix which is used to derive all the evaluation parameters of the model.

A confusion table (also known as a confusion matrix) is a table with two rows and two columns in predictive analytics that records the number of false positives, false negatives, true positives, and real negatives. This facilitates a more detailed review than a simple percentage of accurate classifications (accuracy). If the data set is unbalanced, accuracy can produce misleading results; that is, when the numbers of observations vary greatly in different groups.

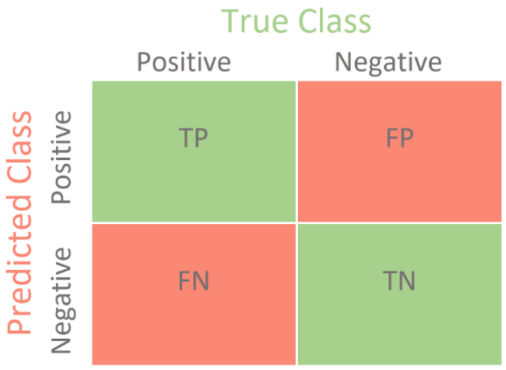

Figure 4: Confusion Matrix

Here,

- TP or true positive refers to the number of predictions where the positive class is correctly predicted as positive by the classifier.
- TN or true negative relates to the number of predictions where the negative class is correctly predicted as negative by the classifier.
- FP or false positive represents the number of predictions in which the negative class is wrongly predicted by the classifier as positive.
- FN or false negative corresponds to the quantity of predictions where the positive class is wrongly predicted as negative by the classifier.

There is no denying in the fact that for a machine learning model, it is often easier to use the confusion matrix as the assessment criterion as this provides with a very basic, but successful performance measurement for the model. Here are some of the most prevalent success metrics from the uncertainty matrix that can be used.

Accuracy: This gives the model's overall accuracy, i.e., the percentage of the total samples that the classifier correctly identified. Below is the formula to measure accuracy:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Misclassification Rate: It represents the fraction of the predictions which are wrong. It is also known as Classification Error and it can be measured by the formula:

$$Misclassification\ Rate = \frac{(FP + FN)}{(TP + TN + FP + FN)}$$

Precision: It informs about the proportion of the positive class predictions which is actually positive. The following formula is used to measure precision:

$$Precision = \frac{TP}{(TP + FP)}$$

Recall: It tells about what fraction of all positive samples the classifier has correctly predicted as positive. It is also called as True Positive Rate (TPR), Sensitivity, Probability of Detection. Using the following method to measure Recall:

$$Recall = \frac{TP}{(TP + FN)}$$

Specificity: It refers to what fraction of all negative samples the classifier properly predicts as negative. The True Negative Rate is also known as (TNR). Using the following formula to determine the specificity:

$$Specificity = \frac{TN}{(TN + FP)}$$

F1-score: In a single measure, it blends precision and recall. It is the harmonic mean of precision and recall mathematically. It can be measured by the following formula:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

In this project, we have used confusion matrix for multiclass classification as the data consists of different categories of heartbeats. Accuracy, precision, F1-score and recall are used as evaluation criteria for both the models. The formulae for the metrics from the uncertainty matrix are taken from the website[3].

2.1.6 Deployment-

A user interface using online integration will be deployed into the cloud platform as part of the deployment of the implemented models. For the future perspective work, web applications and mobile applications can be developed for classifying and detecting the irregularities of the heartbeat.

2.2 Design Specification

This study is based on three tier architecture. Figure 5 represents the architecture being used. The data interpretation layer and business logic layer are used for the access with the client at any point of time.

[3] https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-

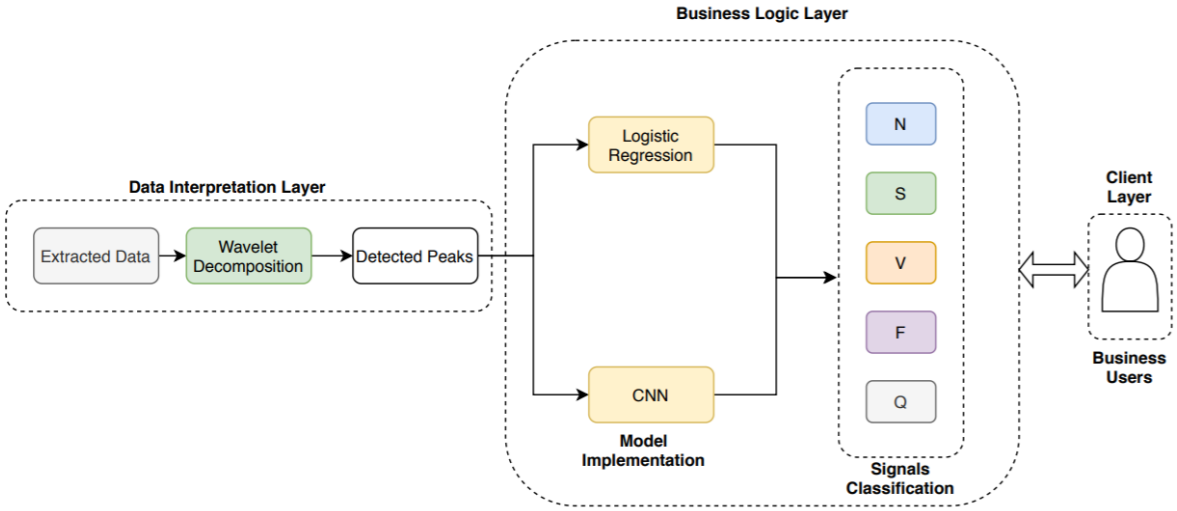

Figure 5: Design Flow

The local machine is used to store data which is extracted in the data interpretation layer. With the help MATLAB, while data preprocessing, all the unwanted noise, base line interference, power line interference are removed. The nine applications in business logic and technical implementation of the model are present in the business logic layer. This layer is used to implement the models used in this study i.e., CNN and logistic regression models. These models are evaluated on the basis of recall, accuracy, computational time, and precision and are assessed. The manual interference by the cardiologists, which are the stakeholders of this project, will be reduced by using this study. This study will help the stakeholders as well as the patients to give and get an early treatment.

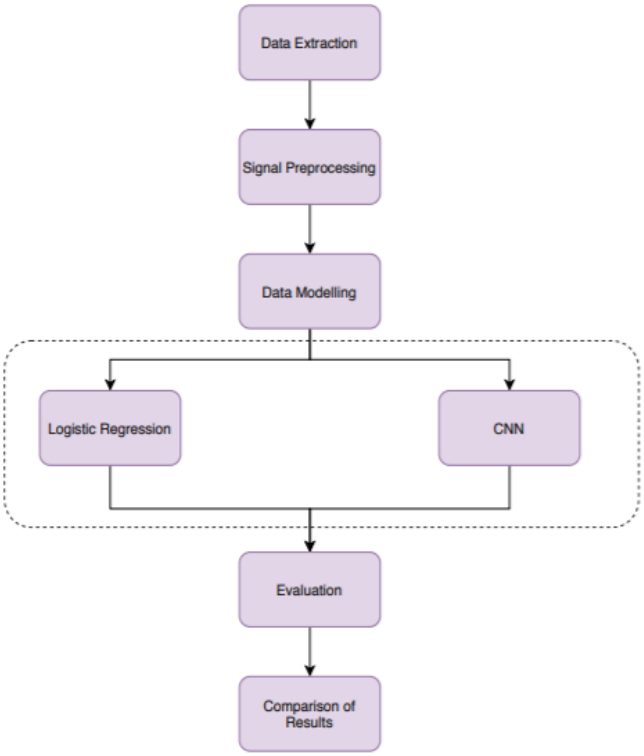

Figure 6: Process Flow

Figure 6 describes the flow of this study. The first process box represents that the data is extracted from the MIT-BIH database. After that the extracted data is fed to the signal preprocessing block. In this block, denoising of the signal is done with the help of MATLAB's discrete wavelet transformation tool. The out of this block is fed to data modeling, which consists of the two models being used n the study i.ie. CNN and logistic regression. At last, the evaluation of the models are done and a comparative analysis is done between both the models.

3. Implementation

This research focuses primarily on the classification of heartbeat signals for different categories. The algorithms like Logistic regression and CNN were implemented during this research. Also signal preprocessing is done using MATLAB.

3.1 Signal Preprocessing

MATAB version R2020b is used in preprocessing of MIT-BIH data. This is done using the Discrete wavelet transformation (DWT) tool in MATLAB. Since the distortions are caused in the ECG signals because of the patient movements and respiration, it is important to remove these noises from the original signals before proceeding with the algorithms. DWT was chosen because the signal displays sluggish and non-localized oscillations with respect to time and frequency. DWT is thus the most powerful way to handle these sudden shifts. Compressed wavelets is used to capture the signal's abrupt changes and extended wavelets (as its finite) is used to capture the original signal. Figure 7 displays half of the samples which are obtained from the tree structure using the Nyquist criteria which differentiates the original signal into high pass and low pass filters.

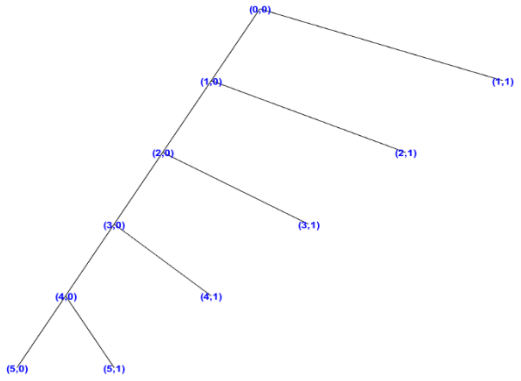

Figure 7: Wavelet Tree

Since the filters have the ability to ignore the sub-band signals and then again rebuild it, a narrower band signal will be iterated by the next sub-band stage by the same technique. The figures below explain the QRS complex and R peaks obtained in the filters after denoising the signals.

In Figure 8, **s** shows the original ECG signal from MIT-BIH database. As it is prominent from the figure below, there are many frequencies in the signal but only the frequencies related to R-peaks must be preserved. For this purpose, a band pass action is needed. So, the wavelet transform is used to separate

signals into different frequency bands. This is achieved by eliminating wavelet coefficients of high frequencies and low frequencies of ECG signal. Therefore, to get these wavelet coefficients, undecimated wavelet transform is used. We have used a 5-level decomposition of an ECG signal by using sym4 option in MATLAB. In figure 8, d1, d2, d3, d4, d5 are the detailed coefficients at different levels respectively and a5 is the approximation coefficient.

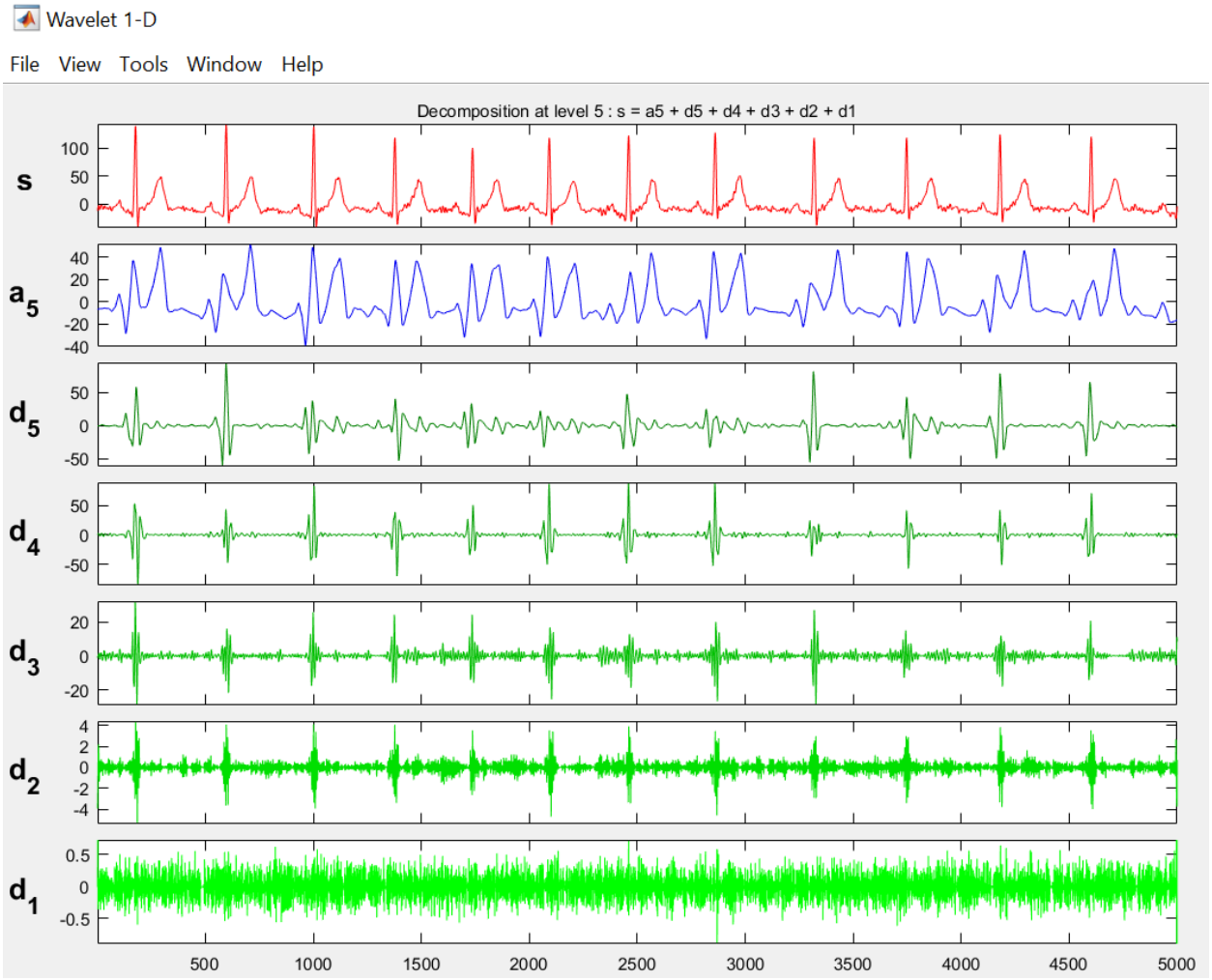

Figure 8: Original Signal with coefficients

A signal obtained after the first filtering step, which, as shown in Figure 9, eliminates the first stage of high-frequency signals.

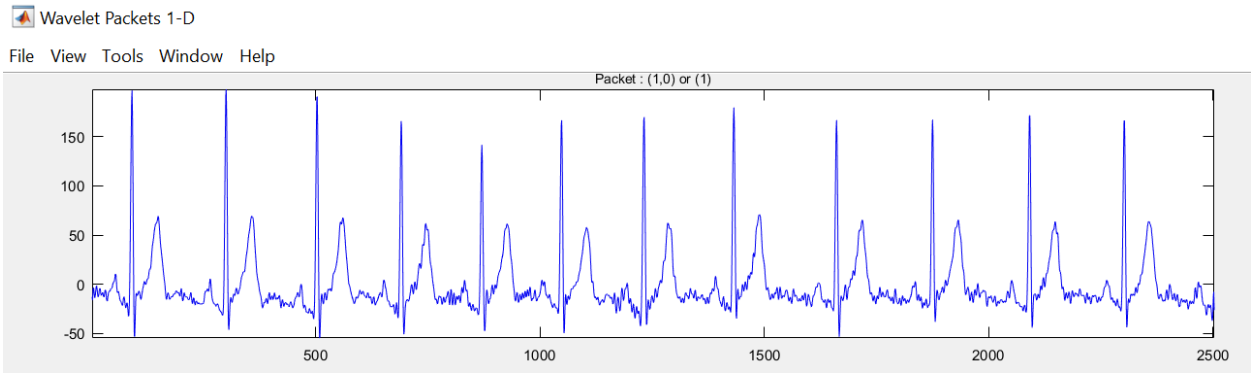

Figure 9: Filter 1- at node (1,0)

The signal obtained after the first filtering stage is shown in Figure 10. This eliminates the second stage of high frequency signals.

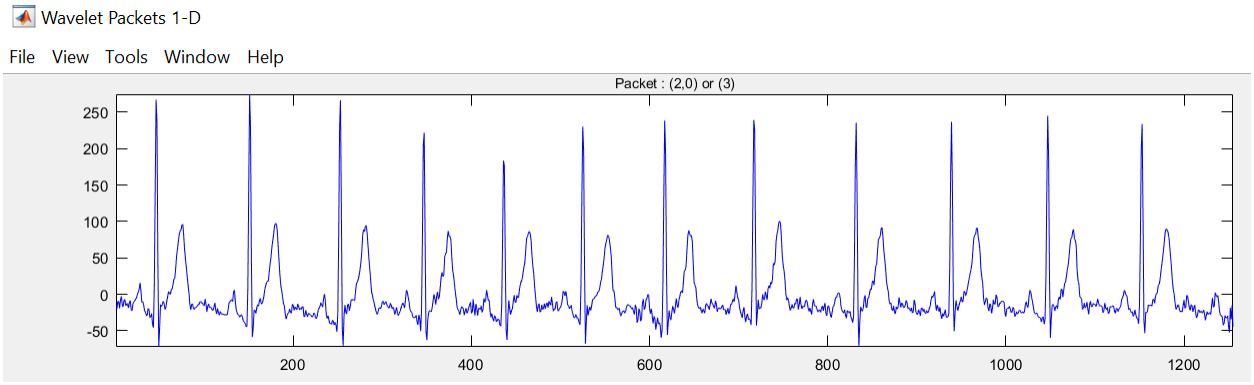

Figure 10: Filter 2- at node (2,0)

Below ,in figure 11, is the signal which is achieved by the third level of filetering at node(2,0). This removes fourth stage of high frequencies signals.

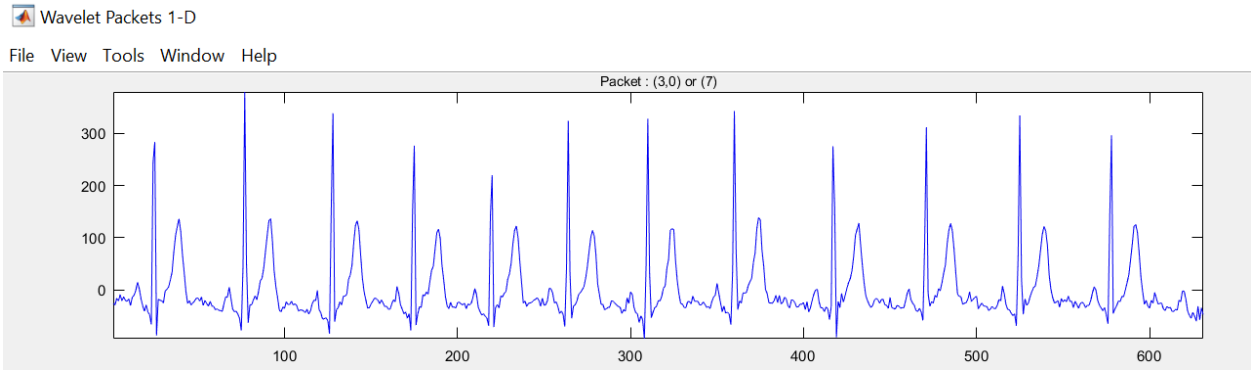

Figure 11: Filter 3- at node (3,0)

Figure 12 shows the signal in which fourth level of high frequencies are removed.

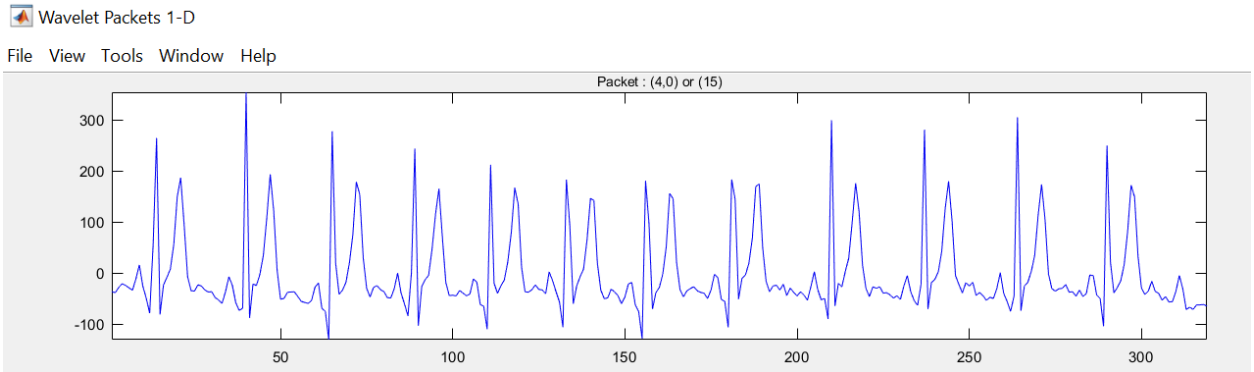

Figure 12: Filter 4- at node (4,0)

The final signal is obtained after removing the fifth level of high frequencies at node (5,0) as shown in figure 13.

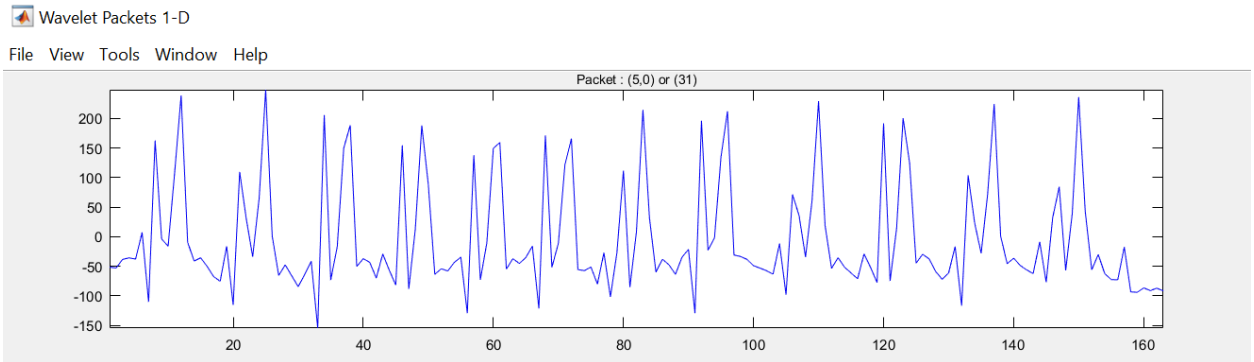

Figure 13: Filter 5- at node (5,0)

Figure 14 represents the obtained denoised signal which is similar to the QRS peak of the ECG waveform.

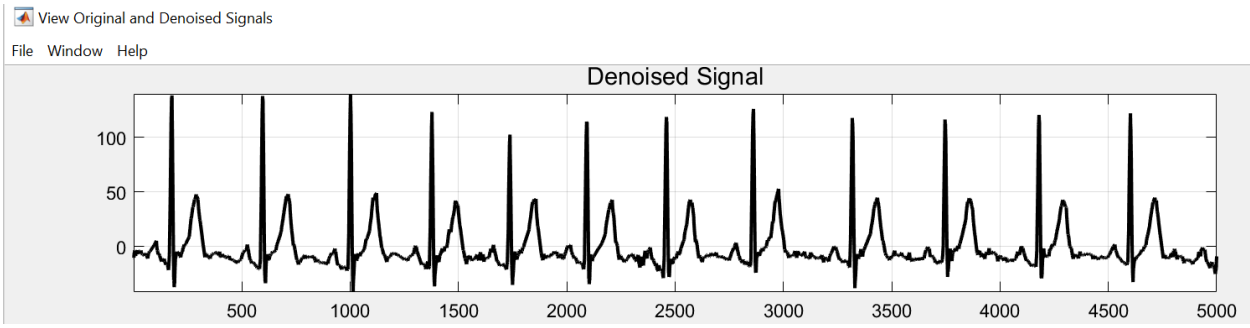

Figure 14: Denoised Signal

3.2 Logistic Regression

Within General Linear Model (GLM), logistic regression is one of the most popular and simple classification models. The logistic regression model is used to give the relationship between the continuous variables

and discrete variables. The hypothesis of the model is confined to 0 and 1 since it uses sigmoid function. The output[4] of the model is represented below

$$Y = \frac{e^{(b_0+b_1x)}}{1+e^{(b_0+b_1x)}}$$

Y -> is predicted output, b_0 -> intercept term, b_1-> coefficient of single input value

The data extracted from MIT-BIH contains eighty thousand samples per beat and are divided into train and test data. To fit to a synchronized time frame, every sample per beat is normalizes to 0 and 1 and padded with zeros. The range is set to 5. Plots per each class are set to 10 and then iterated over. According to this process, the categorization of five categories is achieved and represented in a graph. For various heartbeat types, refer to figure 15.

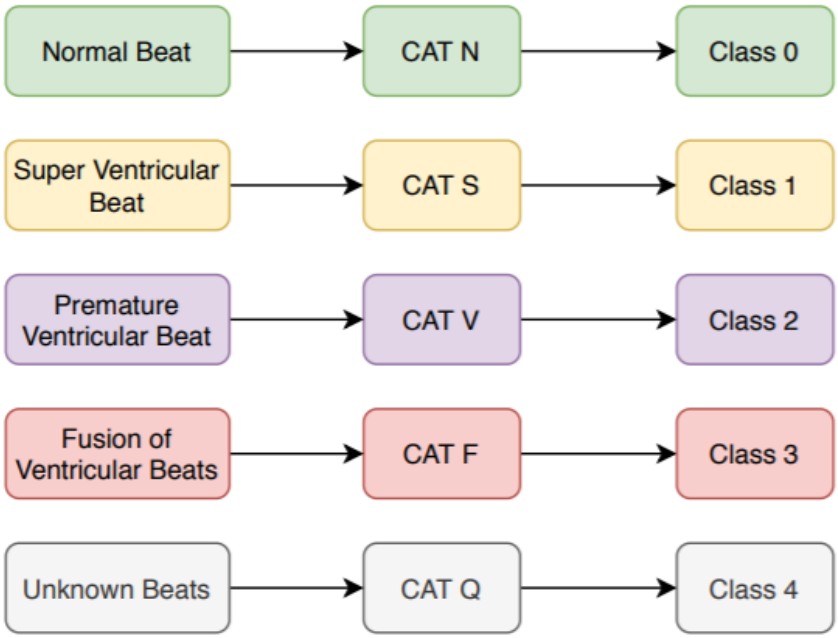

Figure 15: Categories of heart beats(Mathews, Kambhamettu and Barner, 2018)

Gaussian is introduced in this model to smoothen the signals to reduce the randomness of it and expose the signal to understand it better. In classifying the model, Logistic Regression was applied to the extracted MIT-BIH results. The performances of the models are assessed using the precision, recall, and accuracy. By classifying the signals in one of the groups, logistic regression gives the model classification. The estimation of extended limits of different categories becomes difficult as complexity is increased. To overcome this problem, non-linearity is introduced in the model to improve the accuracy, recall, and precision of the model. Fourier transform is used for this purpose. Also, RBF sampler is introduced in the model from the kernel for this purpose. Hence, the computational time of the model is improved along with the access to unlimited dimensions of the function space.

[4] Logistic Regression for Machine Learning (machinelearningmastery.com)

As the complexity increases, the extended limits of multi-class cases are difficult to estimate and, thus, the model needs to be applied with non-linearity to improve the consistency and efficiency of the model.

Logistic regression's output is given by-

$$P(y = |x) = \frac{1}{1 + \exp(-w^T x)}$$

The output formula for logistic regression model as represented by the kernel (Since (x) is mapped to kernel space is given by-

$$P(y = |x) = \frac{1}{1 + \exp(-\sum_{i=1}^{n} \propto {}_i \gamma_i k(x_i, x))}$$

The output of the RBF sampler is given by-

$$K(x, x') = \exp(-\frac{\left||x - x'|\right|^2}{2\sigma^2})$$

Where ||x-x'|| implies the square of Euclidean distance between two vectors x, x'.

For the purpose of clustering, the Gaussian mixture is implemented in the model. This categorizes the models into K numbers of groups in which K represents the individual machine state. The formulas referred to above are taken from website [5].

3.3 Convolution Neural Networks (CNN)

Input, output, and several hidden layers are found in the CNN artificial neural network. The data vector for the input of a single dimensional vector is given by the following formula:

$$x = (x_1, x_2, x_3, x_4, \ldots\ldots\ldots x_n)$$

Where x ϵ R.

The major libraries like Pandas, numpy, Keras, tensorflow and mathplot are installed. Convolution 1D and maxpooling 1D libraries are installed from keras.

Data from both the csv files are loaded and concatenated into one data frame into the CNN model. The count of all the unique values are taken and then displayed. This data is then distributed into five different categories i.e., C0, C1, C2, C3 and C4. These groups are then represented in a graph of 20,12 scale after flattening into row wise. As displayed in figure 16, these 5 groups are represented as the 5 different categories of the heartbeat (N, S, V, F, and Q). Figure 16 represents the different type of irregularities in a heartbeat.

[5] Radial basis function kernel - Wikipedia

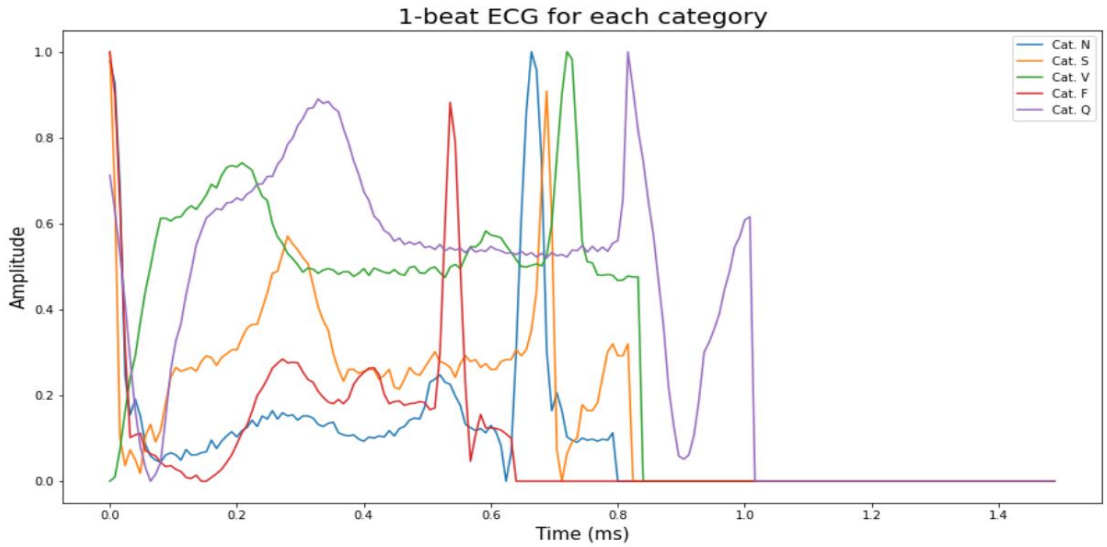

Figure 16: ECG classified into 5 categories per beat

The batch size is set to 500 to address the 3 convolution layers and after implementing conv1D is in the model. Each layer has 32 filters, and with a kernel size of 5 into 5, and moves of 1 have been selected to hop along the array axis from one object to another. The model is trained for 20 epochs and to differentiate the five categories of the heartbeat, activation layer is added into the model (since the classes are not linearly separable). Maxpooling is used to downsample the input. To reduce the dimensionality of the input, maxpooling is used. Rectified linear units (ReLU) is used as the activation function in this neural network. Figure 17 displays the end-to-end flow of the model.

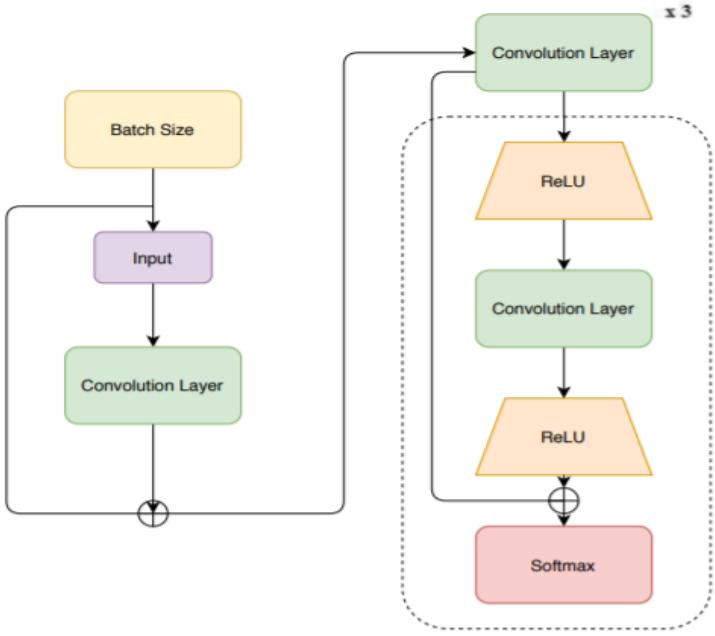

Figure 17: Applied CNN Model

```
Model: "model"
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
input_1 (InputLayer)         [(None, 187, 1)]          0
_________________________________________________________________________________________________
conv1d (Conv1D)              (None, 183, 32)           192         input_1[0][0]
_________________________________________________________________________________________________
conv1d_1 (Conv1D)            (None, 183, 32)           5152        conv1d[0][0]
_________________________________________________________________________________________________
activation (Activation)      (None, 183, 32)           0           conv1d_1[0][0]
_________________________________________________________________________________________________
conv1d_2 (Conv1D)            (None, 183, 32)           5152        activation[0][0]
_________________________________________________________________________________________________
add (Add)                    (None, 183, 32)           0           conv1d_2[0][0]
                                                                   conv1d[0][0]
_________________________________________________________________________________________________
activation_3 (Activation)    (None, 183, 32)           0           add[0][0]
_________________________________________________________________________________________________
max_pooling1d_1 (MaxPooling1D) (None, 90, 32)          0           activation_3[0][0]
_________________________________________________________________________________________________
conv1d_5 (Conv1D)            (None, 90, 32)            5152        max_pooling1d_1[0][0]
_________________________________________________________________________________________________
activation_4 (Activation)    (None, 90, 32)            0           conv1d_5[0][0]
_________________________________________________________________________________________________
conv1d_6 (Conv1D)            (None, 90, 32)            5152        activation_4[0][0]
_________________________________________________________________________________________________
add_2 (Add)                  (None, 90, 32)            0           conv1d_6[0][0]
                                                                   max_pooling1d_1[0][0]
_________________________________________________________________________________________________
activation_5 (Activation)    (None, 90, 32)            0           add_2[0][0]
_________________________________________________________________________________________________
max_pooling1d_2 (MaxPooling1D) (None, 43, 32)          0           activation_5[0][0]
_________________________________________________________________________________________________
flatten (Flatten)            (None, 1376)              0           max_pooling1d_2[0][0]
_________________________________________________________________________________________________
dense (Dense)                (None, 32)                44064       flatten[0][0]
_________________________________________________________________________________________________
activation_6 (Activation)    (None, 32)                0           dense[0][0]
_________________________________________________________________________________________________
dense_1 (Dense)              (None, 32)                1056        activation_6[0][0]
_________________________________________________________________________________________________
dense_2 (Dense)              (None, 5)                 165         dense_1[0][0]
_________________________________________________________________________________________________
softmax (Softmax)            (None, 5)                 0           dense_2[0][0]
=================================================================================================
Total params: 66,085
Trainable params: 66,085
Non-trainable params: 0
```

Figure 18: Parameters and details of CNN

Since the machine learning algorithms cannot convert categorical data into vector format, one hot encoder is used for this purpose. The compilation of the model is done with adam optimizer because of its efficiency and its capability perform well with large data files. At last, the model is trained by keras.fit() function[6] for 20 epochs. Figure 18 above is the output of the convolution layers parameters.

4. Evaluation

The findings of a systematic analysis of both the models are highly reliable and indicate strong efficiency in the classification of heartbeat signals. For each model, the final matrix was determined, such as precision, accuracy, and recall.

The formula for accuracy and recall below are:

[6] Keras: the Python deep learning API

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Samples}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

The logistic regression model achieved a decent enough accuracy of 91.51 percent, with a much lower testing time of 33.04 seconds. Table 2 represents the confusion matrix of logistic regression model.

Table 2:Logistic Regression-Evaluation Metrics Report

	precision	recall	f1-score	support
0.0	0.98	0.89	0.93	18118
1.0	0.31	0.65	0.42	556
2.0	0.69	0.83	0.75	1448
3.0	0.15	0.78	0.25	162
4.0	0.89	0.94	0.91	1608
accuracy			0.88	21892
macro avg	0.60	0.82	0.65	21892
weighted avg	0.93	0.88	0.90	21892

As it is seen from the table, the accuracy of the model is increased after introducing the random fourier non linearity in the model by 3.22 percent which made the accuracy to 91.51 percent.

Table 3: Logistic Regression-Evaluation Metrics Report After Introducing Non-Linearity

	precision	recall	f1-score	support
0.0	0.98	0.92	0.95	18118
1.0	0.40	0.69	0.51	556
2.0	0.73	0.87	0.79	1448
3.0	0.23	0.87	0.36	162
4.0	0.93	0.94	0.94	1608
accuracy			0.91	21892
macro avg	0.65	0.86	0.71	21892
weighted avg	0.94	0.91	0.92	21892

The CNN model yields the accuracy of 99.09 percent. This states that this model is highly relevant for this study. The computational time of the model is 5467 seconds. Figure 19 and table 3 represents the assessment and confusion matrix of the model.

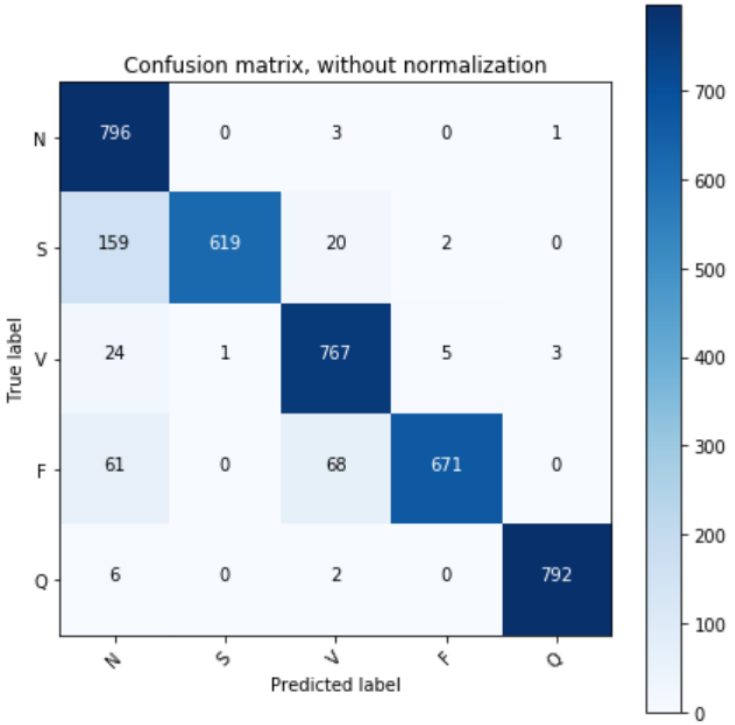

Figure 19: Confusion Matrix for CNN

Table 4: Evaluation Metrics Report for CNN

	precision	recall	f1-score	support
0	0.76	0.99	0.86	800
1	1.00	0.77	0.87	800
2	0.89	0.96	0.92	800
3	0.99	0.84	0.91	800
4	0.99	0.99	0.99	800
accuracy			0.91	4000
macro avg	0.93	0.91	0.91	4000
weighted avg	0.93	0.91	0.91	4000

For both models, the evaluation matrix is compared, and graphs have been plotted using tableau.

As it is clear from the precision graph in figure 20, the precision of CNN model is higher than that of logistic regression model.

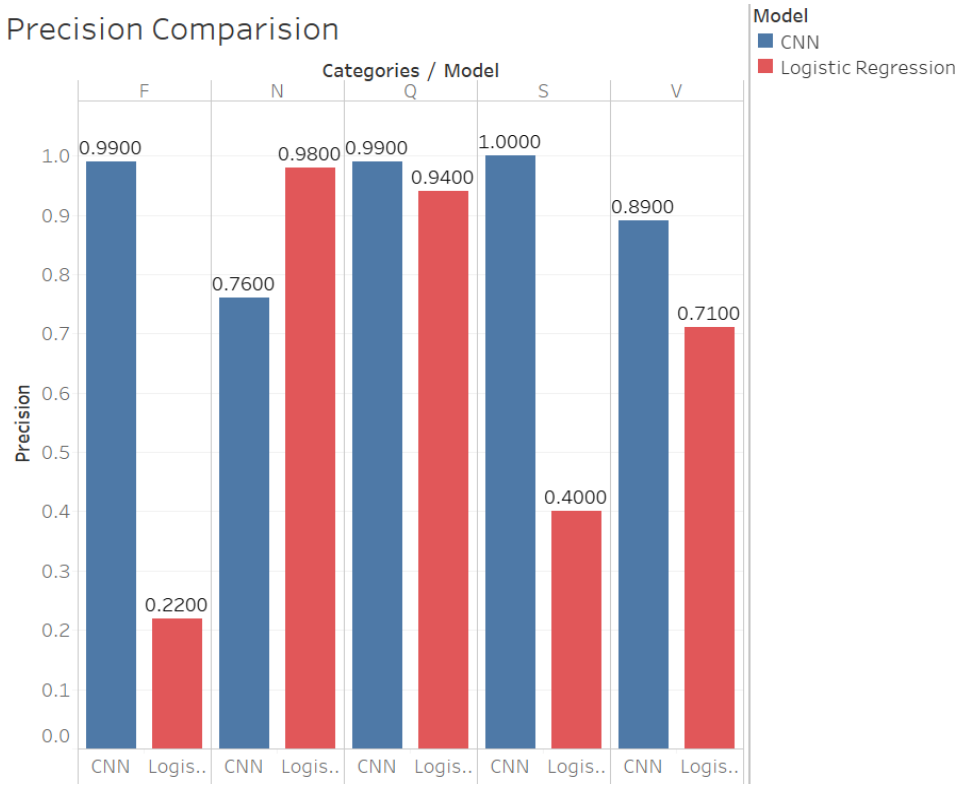

Figure 20: Comparison of Precision

From figure 21, which represents the recall comparative analysis of the logistic regression model and CNN model, it is clear that the recall value of CNN is higher than logistic regression model.

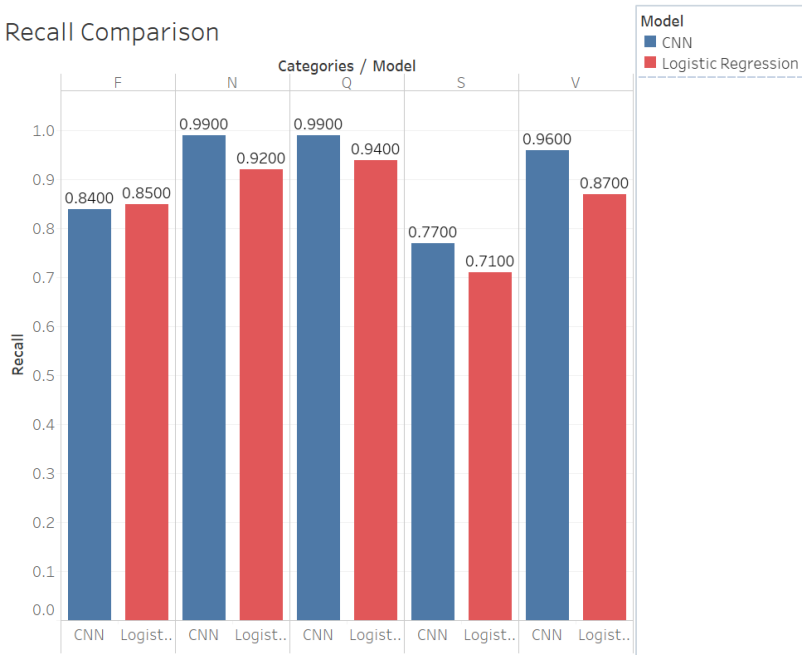

Figure 21: Comparison of Recall values

In the below figure 22, the accuracies of CNN and Logistic Regression models are compared. It is clear from the graph that CNN model has higher accuracy.

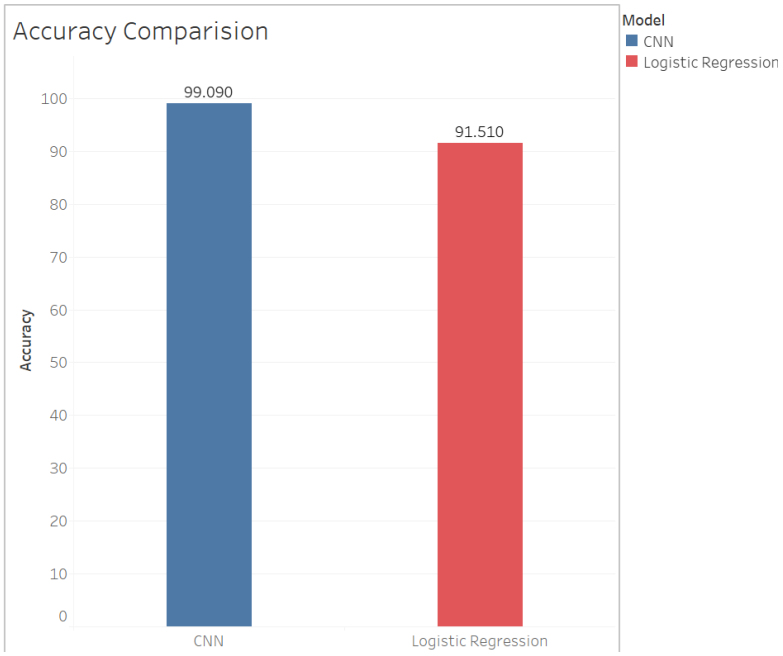

Figure 22:Comparison of Accuracy

The results obtained after the implementation of the two models indicate that, relative to logistic regression, the accuracy obtained from the CNN model is very high. But the CNN model takes longer to compute. Since the CNN model trains the data through the implementation of several hidden layers with high precision and performance, it can be inferred that the CNN model suits the data well compared to logistic regression.

5. Results and Discussion

5.1 Results from Machine Learning Model-

In the classification process, data containing records from preprocessed MATLAB files became the basis for the classification of N, S, V F, and Q. Using sampling, 70 percent in the training phase and 30 percent in the test phase were divided into the classification process. The shape of the training data came out to be 87554 rows × 187 columns and the shape of the test data was 21892 rows × 187 columns. After dividing the data, into training and testing, plots per class graph was plotted, by setting the range to 5, for each class i.e. C0, C1, C2, C3 and C4 and the aesthetics were set. Here each class corresponds to categories of heartbeat, where C0 corresponds to Normal Beat (N), C1 corresponds to super ventricular beat (S) , C2 corresponds to premature ventricular bear (v) , C3 corresponds to fusion beat (F) and C4 corresponds to unknown beats (Q). the output of the graph is represented in figure 23.

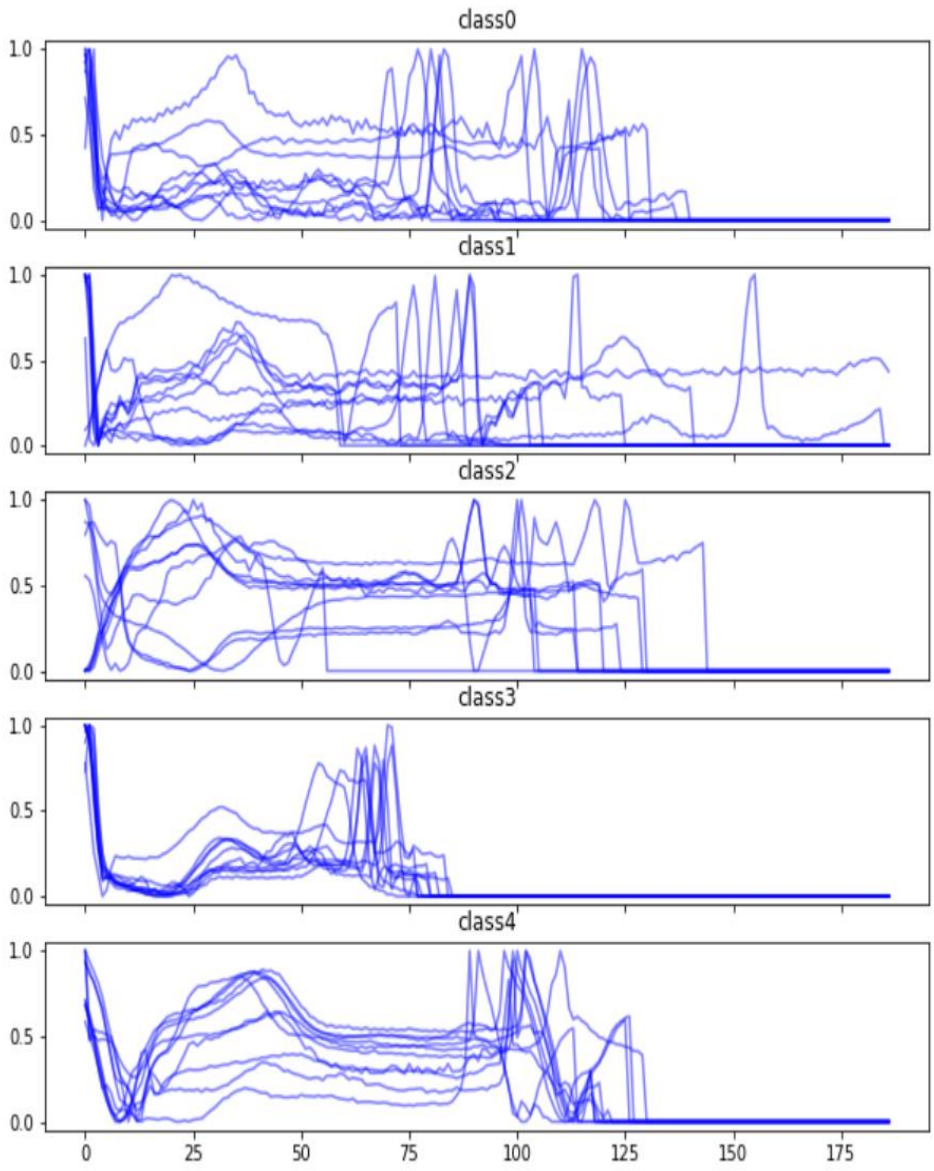

Figure 23: Classification by Logistic Regression

To get a clearer picture of the different waveforms of different heartbeat which is classified in figure 23, we then plotted the waveform as an image with the help of imshow() function of matplotlib library. This process was repeated for all the categories of the heartbeat we had obtained in figure 23. The output of the imshow() is shown in figure 24.

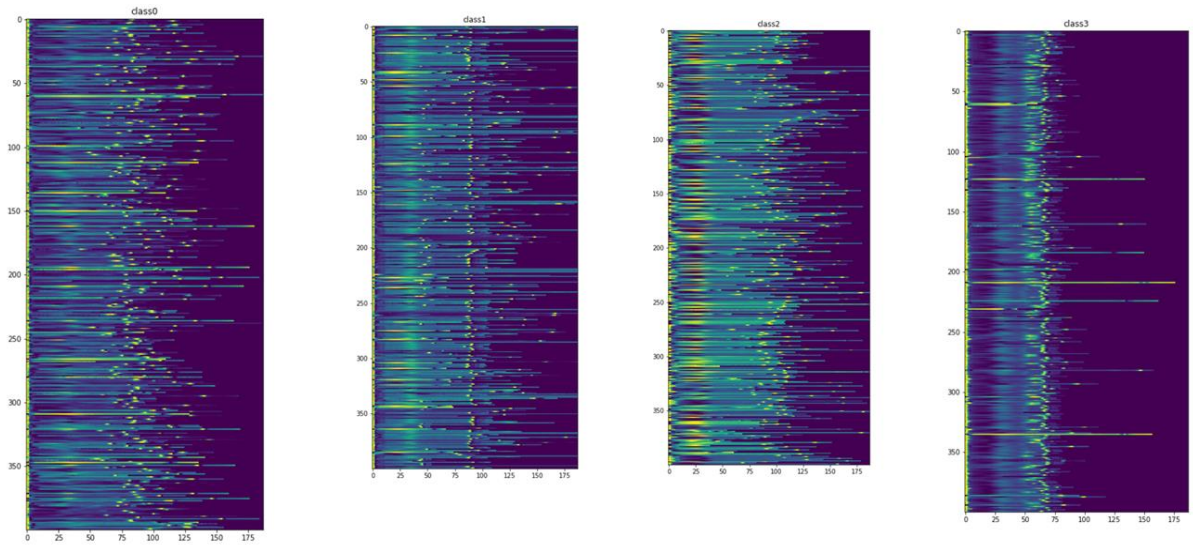

Figure 24: Spectrum representation of the different categories of heartbeat

To get the count of categories in the training and test data frame, graphs for each data frame was calculated. This is represented in figure 25 below.

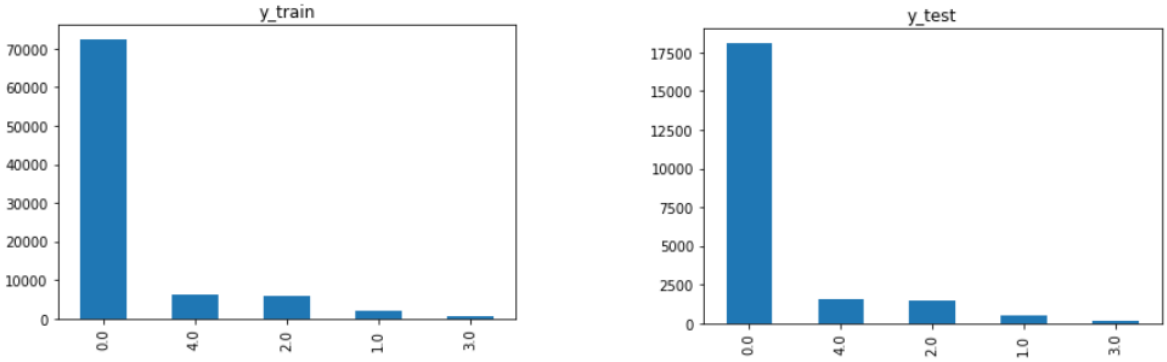

Figure 25: Number of categories in train and test data frames

For optimization and signal processing and to smoothen out the signals, scipy library is used. After importing gaussian from the scipy library, a comparative graph was plotted between the original signal and the signal output after applying the gaussian. This is displayed in figure 26.

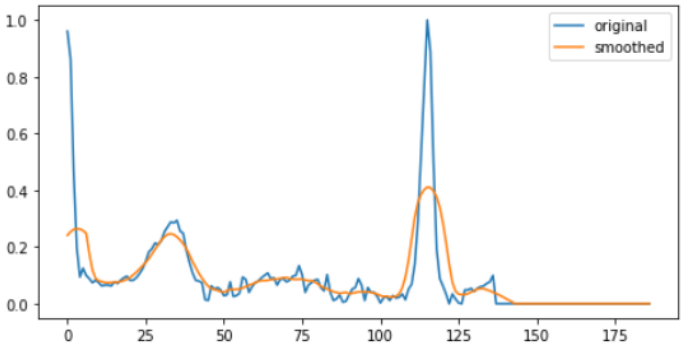

Figure 26: Original and Smoothed signal

To preprocess and normalize the data, two functions were created in which training and testing data were processed one by one and the output is plotted into graphs shown in figure 27 and 28.

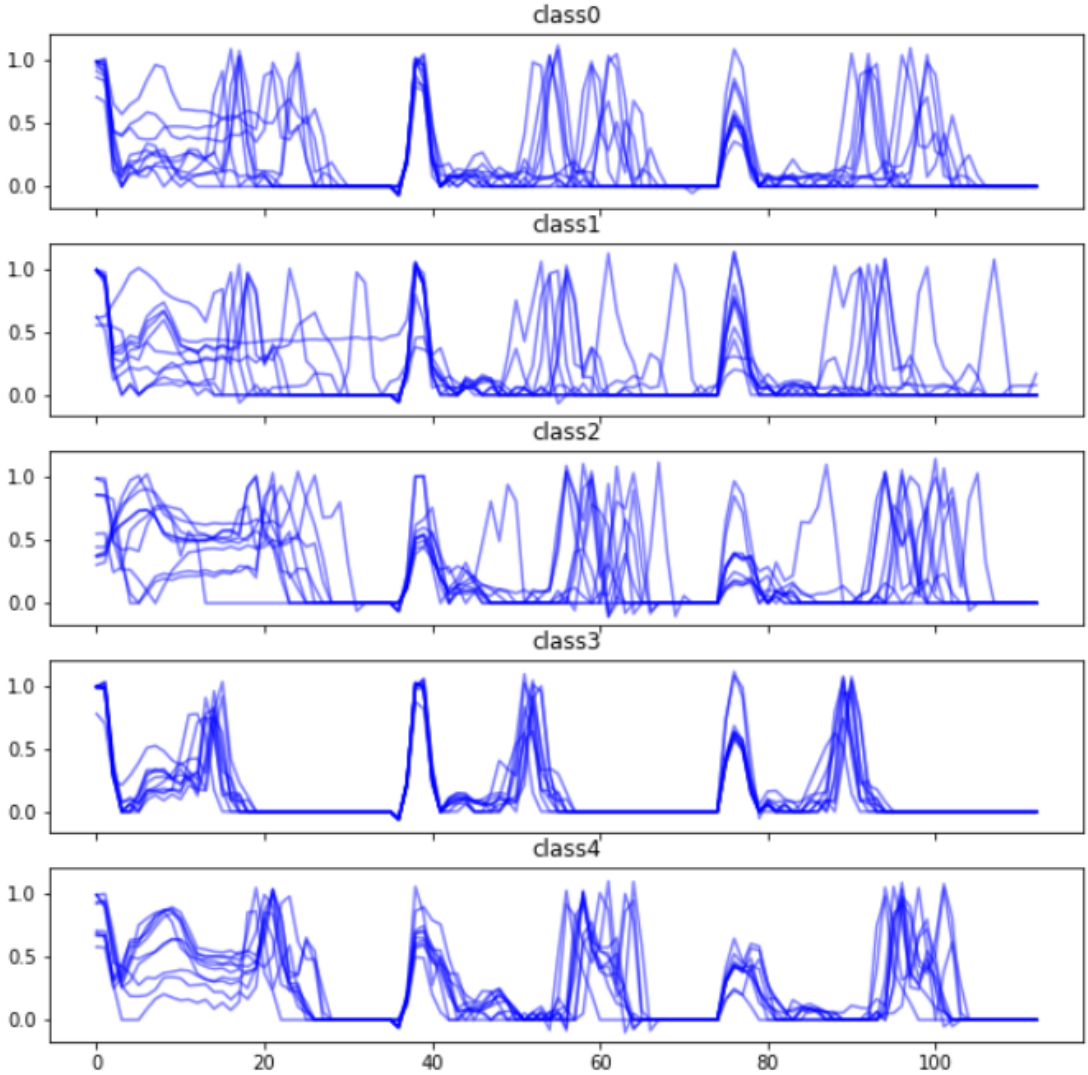

Figure 27: Processed waveform

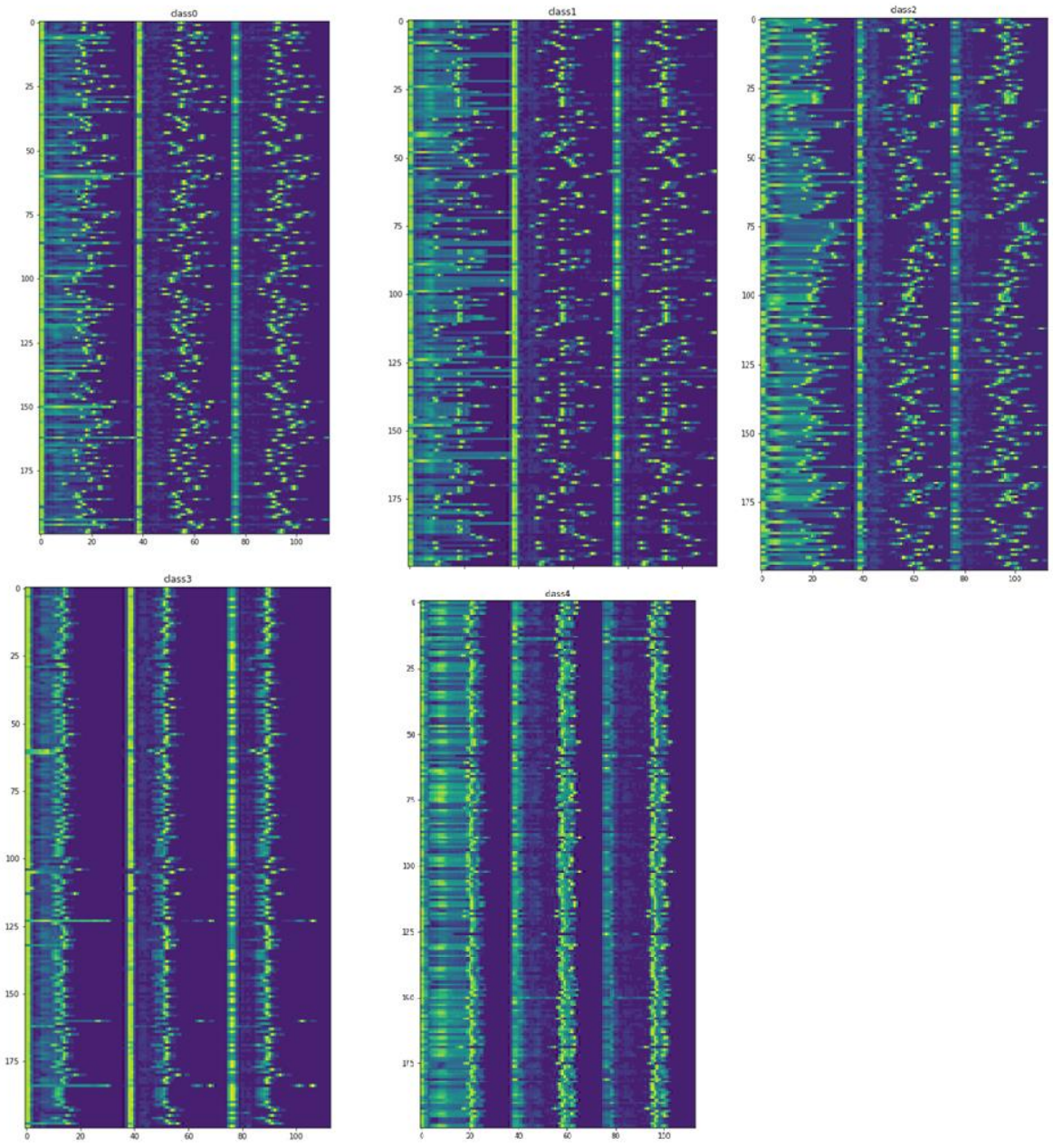

Figure 28: Spectrum of the processed waveforms.

After preprocessing the training and test data, Logistic regression was imported from sklearn library. And to calculate the accuracy and classification of the model, libraries form sklearn was also imported. The classification report was calculated after calculating the confusion matrix. The output of the report is presented in the figure 29.

	precision	recall	f1-score	support
0.0	0.98	0.89	0.93	18118
1.0	0.31	0.65	0.42	556
2.0	0.69	0.83	0.75	1448
3.0	0.15	0.78	0.25	162
4.0	0.89	0.94	0.91	1608
accuracy			0.88	21892
macro avg	0.60	0.82	0.65	21892
weighted avg	0.93	0.88	0.90	21892

Figure 29: classification report before RBF

As the complexity of the data increase, non- linearity increases. To overcome this issue, RBF sampler was used in the model. After introducing the RBF sampler, logistic regression model was trained again keeping newton-cg as the solver and the classification report was calculated again after calculation of the confusion matrix. Table 3 displays the improved accuracy of the model to 91%.

5.2 Results from Deep Learning Model-

In this study, we have used Convolution Neural Network to train our model. Using sampling, 70 percent in the training phase and 30 percent in the test phase were divided into the classification process. The libraries like math, random, pickle, numpy, itertools, pandas, sklearn, scipy, matplotlib were imported. The preprocessed csv files were loaded in two separate data frames and then concatenated to a final data frame. To check whether the data is loaded properly, the count of the data frame was performed. The result is shown on figure 30.

```
0.0    90589
4.0     8039
2.0     7236
1.0     2779
3.0      803
Name: 187, dtype: int64
```

Figure 30: Count of each category in the data frame

The data frame was then cast as int using the astype() function. With the help of numpy's argwhere function, the five different categories of the signals where separated out from the data frame. These different categories of the signal were then flatten with the help of flatten() function pf numpy. After when we had successfully categorized the data frame into C0, C1, C2, C3 and C4, a graph was plotted including these different categories with the help of plot() function of matplotlib. Figure 30 displays the graph amplitude per time of 1-beat ECG for every category .

OneHotEncoder from sklearn was also imported to convert categorical data into vector format. For neural neural networks, keras library was also used. We have used three convolution layers to model CNN. The data was concatenated form both the training and test csv files which were preprocessed by MATLAB. The ECG waveforms were plotted for all the categories of the heartbeat present in the data frame. The out put is represented in a graphical representation in figure 30 below.

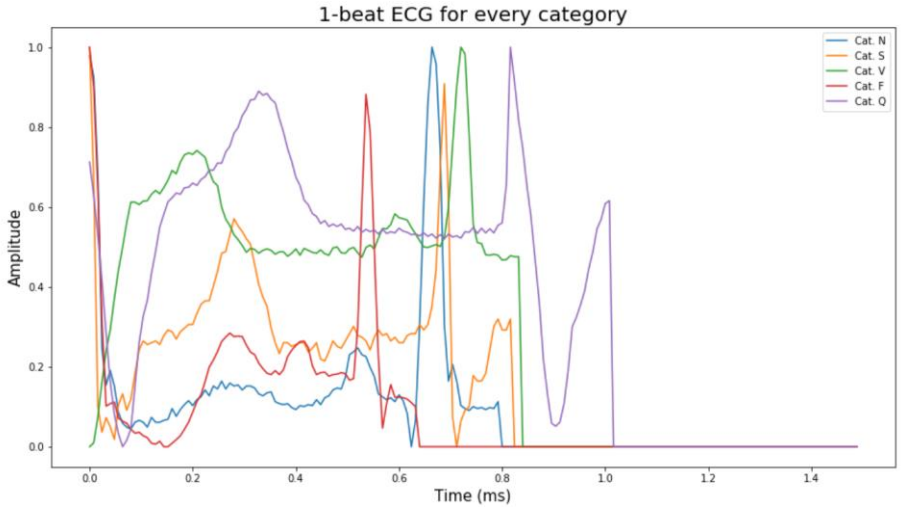

Figure 31: Different categories of heartbeat in the dataset

To resample the data, the all the columns were amplified, stretched, and reshaped. To achieve these three functions were created with the help of random() operator and the data frame which consists of all the columns were passed in these functions. The output is shown figure 32.

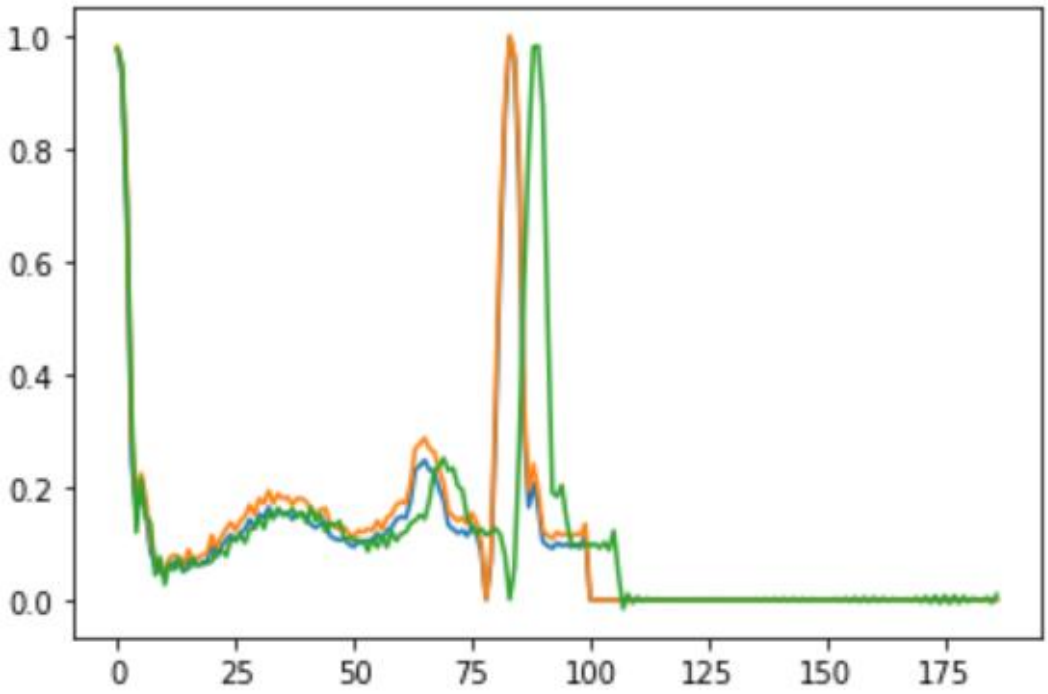

Figure 32: Resampled data

Stacking of the arrays were achieved by using the hstack and vstack function of numpy library. Now to include a new axis which can display an expanded array shape, expand_dims() function of numpy was used and test and train data were passed through it to achieve the shape of the data as shown in figure 33.

```
X_train (109150, 187, 1)
y_train (109150,)
X_test (4000, 187, 1)
y_test (4000,)
```

Figure 33: Data frame after expanding dimension.

As we can see from figure 33, there is a requirement to reshape the data. For this purpose, the one hot encoder was introduced to convert categorical data to vector format. After passing the data from one hot encoder, following results were achieved as displayed in figure 34.

```
X_train (109150, 187, 1)
y_train (109150, 5)
X_test (4000, 187, 1)
y_test (4000, 5)
```

Figure 34: Data frame after one hot encoder

After achieving the data in the required format, the Conv1D was implemented, which was imported from keras.layers. This layer produces a convolution kernel which is transformed to generate a tensor of outputs with the layer input over a single spatial (or temporal) dimension. We have used three convolution layers to train the model. The filters were set to 32 and kernel size as 5. Activation layer reLU was also introduced . Maxpooling, set to pool size as 5, was used by taking the maximum value over the window specified by pool size, the input representation is downsampled. Dense layer of the leras library is used as it helps in non-linearity property of the data. The model is trained where the output is set to softmax function of the dense function. The summary of the model is represented in figure 18.

The model was compiled where the loss was set as categorical_crossentropy and optimizer as adam. The batch size was fixed to 500 and the filters were 32. The model was fitted with adam optimizer for 20 epochs. The history of the model is displayed in figure 35.

```
Train on 109150 samples, validate on 4000 samples
Epoch 1/20
 - 140s - loss: 0.3429 - accuracy: 0.9030 - val_loss: 0.8093 - val_accuracy: 0.7533
Epoch 2/20
 - 144s - loss: 0.1296 - accuracy: 0.9640 - val_loss: 0.5448 - val_accuracy: 0.8413
Epoch 3/20
 - 164s - loss: 0.0980 - accuracy: 0.9720 - val_loss: 0.4567 - val_accuracy: 0.8600
Epoch 4/20
 - 166s - loss: 0.0829 - accuracy: 0.9765 - val_loss: 0.4369 - val_accuracy: 0.8702
Epoch 5/20
 - 192s - loss: 0.0718 - accuracy: 0.9791 - val_loss: 0.3347 - val_accuracy: 0.8965
Epoch 6/20
 - 204s - loss: 0.0654 - accuracy: 0.9805 - val_loss: 0.3798 - val_accuracy: 0.8925
Epoch 7/20
 - 206s - loss: 0.0596 - accuracy: 0.9821 - val_loss: 0.3938 - val_accuracy: 0.8923
Epoch 8/20
 - 185s - loss: 0.0531 - accuracy: 0.9839 - val_loss: 0.3359 - val_accuracy: 0.9082
Epoch 9/20
 - 164s - loss: 0.0494 - accuracy: 0.9848 - val_loss: 0.3051 - val_accuracy: 0.9055
Epoch 10/20
 - 156s - loss: 0.0469 - accuracy: 0.9857 - val_loss: 0.3312 - val_accuracy: 0.9137
Epoch 11/20
 - 155s - loss: 0.0443 - accuracy: 0.9861 - val_loss: 0.3369 - val_accuracy: 0.9047
Epoch 12/20
 - 163s - loss: 0.0407 - accuracy: 0.9873 - val_loss: 0.3011 - val_accuracy: 0.9195
Epoch 13/20
 - 160s - loss: 0.0385 - accuracy: 0.9880 - val_loss: 0.4079 - val_accuracy: 0.8967
Epoch 14/20
 - 162s - loss: 0.0363 - accuracy: 0.9888 - val_loss: 0.3110 - val_accuracy: 0.9140
Epoch 15/20
 - 165s - loss: 0.0337 - accuracy: 0.9892 - val_loss: 0.2315 - val_accuracy: 0.9410
Epoch 16/20
 - 158s - loss: 0.0321 - accuracy: 0.9898 - val_loss: 0.2873 - val_accuracy: 0.9153
Epoch 17/20
 - 165s - loss: 0.0297 - accuracy: 0.9904 - val_loss: 0.1853 - val_accuracy: 0.9440
Epoch 18/20
 - 162s - loss: 0.0294 - accuracy: 0.9907 - val_loss: 0.2588 - val_accuracy: 0.9327
Epoch 19/20
 - 157s - loss: 0.0286 - accuracy: 0.9905 - val_loss: 0.2861 - val_accuracy: 0.9178
Epoch 20/20
 - 157s - loss: 0.0271 - accuracy: 0.9908 - val_loss: 0.3382 - val_accuracy: 0.9112
```

Figure 35: CNN model history

Ranking based average precision, ranking loss, and coverage error was calculated. The result is represented in figure 36.

```
ranking-based average precision : 0.953
Ranking loss : 0.027
Coverage_error : 1.108
```

Figure 36:CNN ranking details

After calculating the confusion matrix, classification report was prepared as shown in table 4. The normalized confusion matrix was also plotted as can be seen in figure 19. The accuracy of the CNN model resulted as 99.09% which is much higher than the Logistic Regression model.

5.3 Tableau Results-

The output from both the models, Logistic Regression and Convolution Neural Network model were exported in a excel file. This excel file was then made as the data source of the tableau workbook for the comparative analysis between the two models.

For precision comparison, categories of the heartbeat and model were taken as the column and sum of the precisions were taken as row (refer figure 20). Similarly, for recall comparison (figure 21) of the models, categories of the heartbeat and model name were taken as column and sum of all the recalls were taken as column. For comparison of the accuracy (figure 22), model name was set as row and accuracy was taken as an attribute. All the three worksheets were combined together to build a dashboard presented in figure 37.

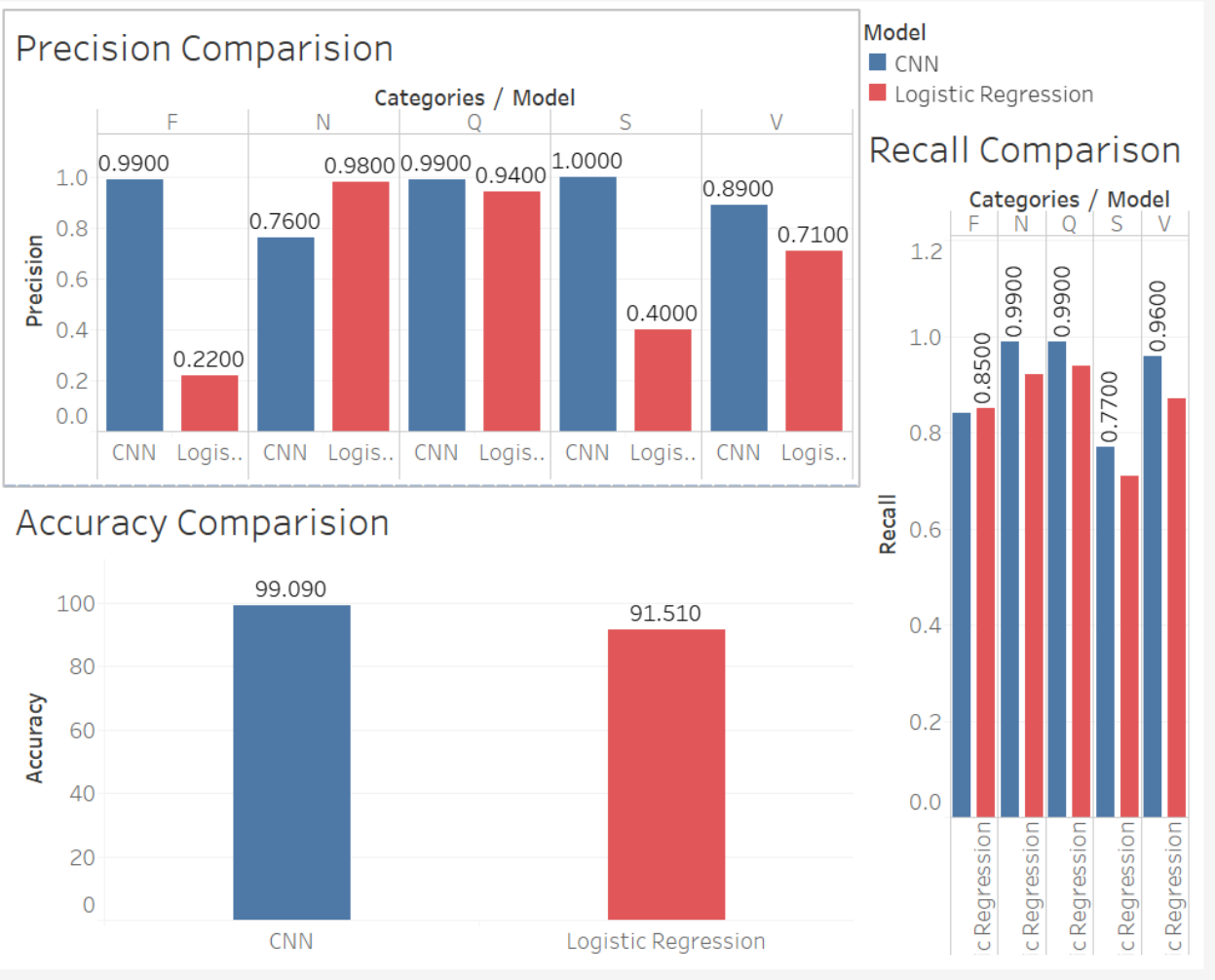

Figure 37: Comparative Analysis of the Models

5.4 Discussion-

The classification of heart signal rates was the main objective of the research project. As the tracking of heartbeat signals differed, for the same patient, the derived data had distinct findings. The classification of heart signal rates was the main objective of the research project. As the tracking of heartbeat signals differed, for the same patient, the derived data had distinct findings. Using the DWT approach wavelet transformation system, pre-processing was carried out by de-noising the signals. A further difficulty in developing the model was that when executed for 70 epochs, the Convolution Neural Network model algorithm did not respond properly. Therefore, for good performance and better precision, the model had to be set at 20 epochs. During the construction of a model rather than a complex architecture, computational costs were taken into consideration. After overcoming the difficulties, the findings obtained were better when predicting the heart signal rate and also the model had a higher accuracy.

6. Conclusion and Future Work

In the research project, early detection of arrhythmia would assist in solving the problem. Cardiac arrhythmia in the ECG cycle is a kind of irregularity that leads to severe heart problems and causes sudden death. Patient heartbeat signals were pre-processed for noise reduction using MATLAB. A comparative study for both the models, machine learning and deep learning, were done by classifying the heartbeat in five different categories. Efficiency i.e., precision, recall, accuracy, and computational time, of each model was calculated. With the help of both Jupyter notebook and tableau tools, data was presented effectively where the deep learning model, Convolution Neural Network, performed better than the machine learning model. CNN model turned out to be more cost effective than the logistic regression model.

To further illustrate the performance and accuracy of the other models, more research can be done by collecting CCU data for more patients and integrating further machine learning and deep learning algorithms. By deploying the results on the cloud platform of the applied models, future study work can be carried out.

References

Ankışhan, H (2019) “Estimation of heartbeat rate from speech recording with hybrid feature vector (HFV),” *Biomedical Signal Processing and Control*, 49, pp. 483–492. doi: https://doi.org/10.1016/j.bspc.2019.01.015.

Baloglu, U. B. *et al.* (2019) “Classification of myocardial infarction with multi-lead ECG signals and deep CNN,” *Pattern Recognition Letters*. Elsevier B.V., 122, pp. 23–30. doi: 10.1016/j.patrec.2019.02.016.

Camacho, D. M. *et al.* (2018) “Next-Generation Machine Learning for Biological Networks,” *Cell*. Elsevier Inc., 173(7), pp. 1581–1592. doi: 10.1016/j.cell.2018.05.015.

de Chazal, P., O’Dwyer, M. and Reilly, R. B. (2004) “Automatic classification of heartbeats using ECG morphology and heartbeat interval features,” *IEEE Transactions on Biomedical Engineering*, 51(7), pp. 1196–1206. doi: 10.1109/TBME.2004.827359.

de Chazal, P. and Reilly, R. B. (2006) “A patient-adapting heartbeat classifier using ECG morphology and heartbeat interval features,” *IEEE Transactions on Biomedical Engineering*, 53(12), pp. 2535–2543. doi: 10.1109/TBME.2006.883802.

Dong, X., Wang, C. and Si, W. (2017) “ECG beat classification via deterministic learning,” *Neurocomputing*. Elsevier B.V., 240, pp. 1–12. doi: 10.1016/j.neucom.2017.02.056.

Dutta, S., Chatterjee, A. and Munshi, S. (2011) “Identification of ECG beats from cross-spectrum information aided learning vector quantization,” *Measurement: Journal of the International Measurement Confederation*. Elsevier Ltd, 44(10), pp. 2020–2027. doi: 10.1016/j.measurement.2011.08.014.

Ha, K. W. and Jeong, J. W. (2019) “Motor imagery EEG classification using capsule networks,” *Sensors (Switzerland)*, 19(13). doi: 10.3390/s19132854.

Hasan, N. I. and Bhattacharjee, A. (2019a) “Deep Learning Approach to Cardiovascular Disease Classification Employing Modified ECG Signal from Empirical Mode Decomposition,” *Biomedical Signal Processing and Control*. Elsevier Ltd, 52, pp. 128–140. doi: 10.1016/j.bspc.2019.04.005.

Hasan, N. I. and Bhattacharjee, A. (2019b) “Deep Learning Approach to Cardiovascular Disease Classification Employing Modified ECG Signal from Empirical Mode Decomposition,” *Biomedical Signal Processing and Control*, 52, pp. 128–140. doi: https://doi.org/10.1016/j.bspc.2019.04.005.

de Jesus, D. R. *et al.* (2018) “Capsule Networks for Protein Structure Classification and Prediction.” Available at: http://arxiv.org/abs/1808.07475.

Kruthika, K. R., Rajeswari and Maheshappa, H. D. (2019) “CBIR system using Capsule Networks and 3D CNN for Alzheimer’s disease diagnosis,” *Informatics in Medicine Unlocked*, 14, pp. 59–68. doi: https://doi.org/10.1016/j.imu.2018.12.001.

Liu, T. *et al.* (2016) “Dictionary learning for VQ feature extraction in ECG beats classification,” *Expert Systems with Applications*, 53, pp. 129–137. doi: https://doi.org/10.1016/j.eswa.2016.01.031.

Luz, E. J. da S. *et al.* (2016) “ECG-based heartbeat classification for arrhythmia detection: A survey,” *Computer Methods and Programs in Biomedicine*. Elsevier Ireland Ltd, 127, pp. 144–164. doi: 10.1016/j.cmpb.2015.12.008.

Mar, T. *et al.* (2011) “Optimization of ECG Classification by Means of Feature Selection,” *IEEE Transactions on Biomedical Engineering*, 58(8), pp. 2168–2177. doi: 10.1109/TBME.2011.2113395.

Martis, R. J. *et al.* (2012) “Application of principal component analysis to ECG signals for automated diagnosis of cardiac health,” *Expert Systems with Applications*, pp. 11792–11800. doi: https://doi.org/10.1016/j.eswa.2012.04.072.

Mathews, S. M., Kambhamettu, C. and Barner, K. E. (2018) “A novel application of deep learning for single-lead ECG classification,” *Computers in Biology and Medicine*. Elsevier Ltd, 99(May), pp. 53–62. doi: 10.1016/j.compbiomed.2018.05.013.

Oh, S. L. *et al.* (2018) “Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats,” *Computers in Biology and Medicine*. Elsevier Ltd, 102(April), pp. 278–287. doi: 10.1016/j.compbiomed.2018.06.002.

Park, J. and Kang, K. (2014) “PcHD: Personalized classification of heartbeat types using a decision tree,” *Computers in Biology and Medicine*, 54(1), pp. 79–88. doi: 10.1016/j.compbiomed.2014.08.013.

Sajjad, M. *et al.* (2019) “Multi-grade brain tumor classification using deep CNN with extensive data augmentation,” *Journal of Computational Science*, 30, pp. 174–182. doi: https://doi.org/10.1016/j.jocs.2018.12.003.

Sannino, G. and de Pietro, G. (2018) “A deep learning approach for ECG-based heartbeat classification for arrhythmia detection,” *Future Generation Computer Systems*. Elsevier B.V., 86, pp. 446–455. doi: 10.1016/j.future.2018.03.057.

Sellami, A. and Hwang, H. (2019) “A robust deep convolutional neural network with batch-weighted loss for heartbeat classification,” *Expert Systems with Applications*. Elsevier Ltd, 122, pp. 75–84. doi: 10.1016/j.eswa.2018.12.037.

Shi, H. *et al.* (2019) “A hierarchical method based on weighted extreme gradient boosting in ECG heartbeat classification,” *Computer Methods and Programs in Biomedicine*. Elsevier B.V., 171, pp. 1–10. doi: 10.1016/j.cmpb.2019.02.005.

Soria, M. L. and Martínez, J. (2009) “Analysis of multidomain features for ECG classification,” in *2009 36th Annual Computers in Cardiology Conference (CinC)*, pp. 561–564.

Wang, G. *et al.* (2019) “A global and updatable ECG beat classification system based on recurrent neural networks and active learning,” *Information Sciences*. Elsevier Inc., 501, pp. 523–542. doi: 10.1016/j.ins.2018.06.062.

Ye, C., Kumar, B. V. K. V. and Coimbra, M. T. (2012) “Heartbeat classification using morphological and dynamic features of ECG signals.,” *IEEE transactions on bio-medical engineering*. United States, 59(10), pp. 2930–2941. doi: 10.1109/TBME.2012.2213253.

Yildirim, O. *et al.* (2019) “A new approach for arrhythmia classification using deep coded features and LSTM networks,” *Computer Methods and Programs in Biomedicine*. Elsevier B.V., 176, pp. 121–133. doi: 10.1016/j.cmpb.2019.05.004.