MSc Applied AI and Data Science

Classification of 12-lead ECG to Identify Different Types of Arrhythmias Using Machine Learning and Deep Learning

ANON

SOLENT UNIVERSITY SOUTHAMPTON

MSc Applied AI and Data Science

Classification of 12-lead ECG to Identify Different Types of Arrhythmias Using Machine Learning and Deep Learning

Acknowledgement

Abstract

This study aimed to develop automatic classification of 12-lead ECG to predict different types of cardiac arrhythmias by using traditional machine learning algorithms and deep learning model, CNN and Bi-LSTM(hybrid model) for quick and effective diagnosis which aids in early treatment. This work utilises a large scale 12-lead ECG dataset of Physionet data containing 10,646 records which was initially shared at figshare as data source and classifies four rhythm classes namely AFIB, SB, SR, ST. Random undersampling approach is used to balance the dataset. Feature extraction is done by using a simple peak detection algorithm to detect R peaks, calculated heart rate using RR intervals and statistical calculations of RR intervals and heart rate are done for all the 12 leads thus extracting 72 features. The extracted features are concatenated with 11 existing features from the diagnostics file to obtain total of 83 features. Feature selection is done using SelectKBest and f_classif and 31 features were selected based on the k highest score. The features reduced are used for training the model to make predictions in classification task. The deep learning model takes reduced features as input and 1D CNN extracts features from the input data and Bi-LSTM learn the temporal relationship between the extracted features and performs classification and predicts arrhythmias. The experimental results indicate that the proposed hybrid model outperforms the traditional machine learning models achieving promising results with accuracy and F1 score of 93% followed by Random Forest classifier with accuracy and F1 score of 92.5%.

Contents

List of Figures

List of Tables

Acronyms

AI	Artificial Intelligence
AF	Atrial Flutter
AFIB	Atrial Fibrillation
AT	Atrial Tachycardia
AVNRT	Atrioventricular Node Reentrant Tachycardia
AVRT	Atrioventricular Reentrant Tachycardia
Bi-LSTM	Bi-directional Long Short-Term Memory
CAE	Convolutional Auto-encoder
CD	Cardiac Disorders
CNN	Convolutional Neural Network
CR	Compression Ratio
CV	Cross Validation
DCNN	Deep Convolutional Neural Network
DT	Decision Tree
ECG	Electrocardiogram
EKG	Electrocardiogram
ENN	Edited Nearest Neighbour
FN	False Negative
FP	False Positive
HCM	Hypertrophic Cardiomyopathy
HR	Heart Rate
HRV	Heart Rate Variability
ICA	Independent Component Analysis
LSTM	Long Short-Term Memory
Max HR	Highest Heartrate
Mean RR	Mean of RR-interval
Mean HR	Mean of Heartrate
Min HR	Lowest Heartrate

KDF-WKNN	Kernel Difference Weighted K-Nearest Neighbour
KNN	K-Nearest Neighbour
LOESS	Local Polynomial Regression Smoother
PCA	Principal Component Analysis
PRD	Percentage Root Mean Square Difference
PTA	Pan Tompkins Algorithm
RELU	Rectified Linear Unit
RMS	Root Means Squared
SAAWR	Sinus Atrium to Atrial Wandering Rhythm
SB	Sinus Bradycardia
SDNN	Standard Deviation of the RR-intervals
STD HR	Standard Deviation of the Heartrate
SI	Sinus Irregularity
SMOTE	Synthetic Minority Oversampling Technique
SR	Sinus Rhythm
ST	Sinus Tachycardia
SVM	Support Vector Machine
SVT	Supraventricular Tachycardia
TMSE	Tan Mean Square Error
TN	True Negative
TP	True Positive
TPR	True Positive Rate

1. INTRODUCTION

This Chapter summarizes a brief understanding into the research topic, research question, aims and objectives of this study, proposed artefact and societal impact.

Arrhythmia also known as dysrhythmia is a heart disorder that reflects an irregularity or abnormality of the heartbeat or heart rhythm. If there is abnormality in the functionality of electrical impulses that regulates heartbeat, it leads to arrhythmia and shows changes in the heart rate. These abnormalities may be minor disturbances or sometimes fatal. (National Health Service, 2021)

Cardiovascular diseases are major cause of death of 32% of people worldwide every year. (World Health Organization, 2021). Of all the types of arrhythmias, atrial fibrillation (AFIB) is common and inimical with considerable increase in the cardiac impairment and stroke. According to American Heart Association reports, in 2015, AFIB is mentioned as cause of death in 1,48,672 death certificates from US. IN 2010, evaluated incidence of AFIB in US vary between 2.1 and 6.1 million. And by 2030, it is anticipated to increase to 12.1 million.US is not unique to this terrifying condition. Even in Europe, the incidence of AFIB is evaluated to be 8.8 million in adults with age group 55 and above and by 2060 it was expected to increase to 17.9 million. In Chinese population in adults older than 35 years the incidence of AFIB was 0.71%. (Zheng et al., 2020) More than 2 million people are being experienced by arrhythmias in UK. If it is diagnosed properly, arrhythmia patients can also lead a normal life. (National Health Service, 2021)

The diagnosis of arrhythmia is done by a common test known as electrocardiogram also known as ECG or EKG. It is used generally to check a person's electrical activity and rhythm of the heart. It is non-invasive and painless test that can detect heart problems including arrhythmias (National Health Service,2021)

During standard 12-lead ECG test ,10 electrodes are used. Six electrodes are attached to the chest and remaining electrodes are attached each to ankle and wrist of legs and arms respectively using sticky pads for recording electrical activity of patient's heart in twelve different views, while patient is lying down in resting position (British Heart Foundation,2017). But when symptoms of heart disease triggers by physical activity, then ECG test is carried out while patient is using treadmill or exercise bike whereas if symptoms come and go, continuous ECG monitoring is recommended for which Holter monitor, a small wearable device that usually records ECG for 24 to 48 hours is used or Event monitor, a portable device is recommended, that can be used longer than a Holter monitor, and it automatically records when irregular rhythm is detected (Moores, D. (2018))

The graphical representation of heart's electrical activity is known as ECG lead.12 lead ECG produces twelve separate graphs on ECG paper. Based on the irregularity shown by a particular lead, doctors can trace out myocardial infraction has occurred in which region of heart or whether the problem in heart rhythm is from right or left ventricle of heart. (Institute for Quality and Efficiency in Health Care, 2019)

12-lead ECG test and wearable ECG monitoring devices will generate data in substantial amounts and demands more time and effort for diagnosis by human experts. Sometimes it is common to mis-diagnose certain types of arrhythmias even by trained cardiologists. Therefore, to reduce workload on doctors and to diagnose quickly and accurately, it is necessary to automatically process and interpret ECG signals. Many arrhythmia data sets are made available as opensource like Physionet database, MIT-BIH database, UCI machine Learning Repository which plays leading role in improving area of research in diagnosing cardiac arrhythmia using Artificial intelligence (AI). A significant success has witnessed in the field of machine learning and deep learning due to comprehensive advances in several applications domains like computer vision, signal processing and natural language processing. Machine learning and deep learning and statistical tools can be used for ECG signal analysis and automated diagnosis of arrhythmia disease with more accuracy. (Zheng et al., 2020).

1.1 Research Question

Most of the research was done by researchers in developing classification models by applying artificial intelligence to diagnose arrhythmias using 1-lead or 2-lead ECG signal data. But standard 12-lead ECG test is utilized in clinical setting for detecting cardiac arrhythmias.12-lead ECG is complex and therefore finding tough to utilize existing models with 1-lead or 2-lead ECGs for applying on 12-lead ECG.

Since availability of 12-lead ECG signal data is limited for identifying cardiac arrhythmias, little or less research is done using 12-lead ECGs. Few researchers

- Conducted research using 1-lead ECG data from MIT-BIH and performed multiclass classification using combination of deep learning models and obtained accuracy of 99%. (Yildirim et al., 2019)
- utilized smaller dataset of 12-lead ECG, overseen class imbalance problem and by computing feature importance and applying feature selection performed multiclass classification using machine learning models and obtained accuracy of more than 90%. (Manju and Nair, 2019)
- created a 12-lead ECG dataset from diverse sources and performed multiclass classification after handling class imbalance problem and obtained accuracy of 90.7%. (Barišić and Jović, 2022)
- Research question raised from analyses of previous works is **"How can the accuracy of cardiac arrhythmias prediction be improved with machine learning and deep learning models using features extracted from 12 lead ECG?"**

  "How can dimensionality reduction or feature selection improve the performance of machine learning and deep learning models?"

1.2 Aim and Objectives

The main aim of this research is to develop a 12-lead ECG multiclass classification system to identify arrhythmias using machine learning and deep learning with classification accuracy and F1 Score of more than 90% in order to minimize waiting time of the patients for the results and increase their life expectancy.

The objectives of this research are:

- Collect the dataset.
- Perform data analysis and check distribution of classes in the dataset and handle imbalanced data.
- Carryout data pre-processing by data cleaning, feature extraction and dimensionality reduction.
- Perform model parameter optimization using hyper parameter tuning with Random Grid Search to improve classification accuracy.
- To apply machine learning models Random Forest classifier, XG Boost, Decision tree, Support Vector Machine classifier, Gradient boosting and deep learning model CNN and Bi- LSTM(hybrid model) to the collected 12-lead ECG data to identify Arrhythmias.
- Evaluate and compare the performance of machine learning models and deep learning model and aiming to get accuracy above 90%. Calculate metrics such as accuracy, F1 score, precision and recall for checking performance of trained models.
- Implement a web application (GUI) Which interact with users to predict the type of arrhythmias using Streamlit, Python and Machine learning models.

1.2 Proposed artefact and societal impact

The artefact for this project is development of web app using streamlit to identify patients with arrhythmias using trained models that gives best classification accuracy using 12-lead ECG data.

- The method can be utilized in medical domains to automate the ECG-based detection of heart problems like myocardial infraction, atrial fibrillation, and other heart conditions.

- The technology can be used to create healthcare apps that are compatible with cloud environments and can be implemented in the real world.

- The automated diagnosis method can be used in real time in Holter monitors and save lives by early detection.

- Individuals, health professionals, governments and global health community will benefit from the automated diagnosis of cardiac arrhythmias than depending on the human experts which is time consuming.

By diagnosing arrhythmias at the preliminary stages, mortality can be reduced and provide with treatment at the earliest. Failure to detect and treat cardiac arrhythmias leads to significant loss of lives globally due to heart disorders.

The research project report is structured as follows. Chapter 2 describes several related works. Chapter 3 give details about the dataset, data pre-processing , models implemented in the study and implementation of models to make predictions. Chapter 4 shows the results of the analysis and discuss them and compare them with the benchmark studies. Chapter 5 describes the limitations of this work. Chapter 6 explains the conclusion of this work and future work. Chapter 7 includes references in Harvard style. Chapter 8 contains appendices with Ethics release form and link to GitHub for artefact.

2. BACKGROUND AND LITERATURE REVIEW

This Chapter describes the related works. The articles reviewed are selected from IEEE Digital Library website using advanced search and using keywords like “machine learning”, ”Arrhythmias”, ”Deep Learning”,”12 lead ECG” etc. using year range from 2012 – 2022 and relevantly sorted.

(Bodini, Rivolta and Sassi, 2020) ensembled four classification models P-CNN, QRS-CNN, T-CNN, Rhythm-NN and designed a machine learning algorithm and trained a labelled dataset with about 40,000 records collected from various sources and performed multi-class classification to identify 27 different types of cardiac disorders (CD) by concatenating predictions of each model. Evaluation of the algorithms is done by using an unseen dataset. In pre-processing and Feature Extraction, ECG signals were downsampled or upsampled to 500 Hz and powerline interference and high frequency noise are reduced using bandpass Butterworth filter. For each lead they created an average PQRS template and concatenation was done later. From RR interval features like median, standard deviation, minimum and maximum distance were extracted. Among all the neural network models QRS-CNN performed best but on the whole intermediate classification performance is shown by ensemble model. Future improvements that can be done on this model are substituting Rhythm-NN with more efficient model, to show changes of heart RR binning can be used instead of averaging PQRS beats .

(Melgarejo-Meseguer et al., 2018) used Support Vector Machine (SVM) and Principal Component Analysis (PCA) for binary classification of patients with Hypertrophic cardiomyopathy (HCM) by detecting myocardial fibrosis in a standard 12-lead ECG. Dataset used contains 12-lead ECGs collected from 43 patients who were clinical diagnosed for HCM. During pre-processing, bandpass filter (0.5 to 75 Hz) was used to filter the signal and cube spline interpolation was used to reduce baseline noise and notch filter (50 Hz) is used to reduce powerline interference. Pan-Tompkins algorithm modified version was used to extract QRS complexes to use in algorithm that helps in detecting fibrosis. Signal transformation is done using PCA and independent component analysis (ICA) to extract signal having information of fibrosis. In feature extraction, features like power, kurtosis, standard deviation and local maxima are computed transformed signal containing QRS complex. Grid search with 5-fold cross validation was used for optimization of model. Accuracy of 76.92% was obtained. Future studies aim at applying the methods used in this research in new type of records obtained from wearable devices.

(Chen, Y.-J., Liu, C.-L., Tseng, V.S., Hu, Y.-F. and Chen, S.-A. (2019)) proposed a deep learning model by combining Bi-directional long short-term memory (Bi-LSTM) and convolutional neural network (CNN) to perform multiclass classification to identify six types of arrhythmias using 12-lead ECG dataset containing 7,704 samples. The model learnt representation of features using ResNet with 1D convolution and temporal relationship existing between these features using LSTM. Cross entropy loss was used as the loss function along with Adam optimizer ,5-fold accuracy cross validation for classification of 12 lead ECG data.81% is the average accuracy and 82% of F1 Score achieved for this model. Limitations of this model include room of improvement for classifying three among six classes because these classes have limited number of samples or because the dataset is imbalanced.

(Manju and Nair, 2019) used 12-lead ECG dataset with 452 records and 279 features from UCI. Out of 16 classes from the actual dataset, they considered only 10 classes for their study. Pre-processing is done first by standardizing the data followed by handling of imbalance in the dataset by using SMOTEENN which is combination of Synthetic Minority Oversampling Technique (SMOTE) and for under-sampling, Edited Nearest Neighbour (ENN). Feature importance is computed, and feature selection is done using XGBoost. Classifiers such as Decision Trees, Random Forest, K-Nearest Neighbour (KNN) and Support Vector Machine (SVM) were trained with SMOTEENN dataset and achieves accuracy of more than 90% for all the models. The future work of this study is to focus on using real time data and considering more arrhythmias.

(Han, Tu and Yang, 2022) constructed a deep learning model by combining hand-crafted features with DenseNet, multi-modal Densenet for classification of ECG signal data containing 43,101 samples from Physionet 2020 Challenge dataset.

In the pre-processing step 0.05Hz -40Hz second order IIR digital bandpass filter is used for denoising ECG signals and over-sampling is done to deal with imbalanced data. Hand crafted features like heart rate variability features and features based on P, T, QRS segments were extracted based on lead II.R-peak localization algorithm was used to find R peaks and to calculate heart rate variability. Complexity of different segments are measured using diverse types of entropy. Threshold optimization was used and Softmax function was set as activation function to improve the prediction of a class. F1-score of 0.58 was obtained. Future work expects at improving the accuracy and optimizing the classification task of the model using more datasets.

(Zuo et al., 2008) constructed a kernel difference weighted k-nearest neighbour (KDF-WKNN) for identifying arrhythmias by classifying 12 lead ECG database from UCI Machine Learning Repository containing 452 samples from 16 different classes. Problem with missing attribute values is coped up by use of modified principal component analysis approach. To determine classification accuracy, 10-fold cross validation (CV) method was applied and achieved accuracy of 70.66%.

(Cheng, Zou and Zhao, 2021) proposed ECG signal classification by deep CNN (DCNN) and BiLSTM using single-lead data from 2017 Physionet/CINC Challenge ECG dataset. After screening out too short signals ,7561 out of 8528 samples of data was finally used for analysis to classify ECG signals into normal, AF, other rhythm and noisy. During pre-processing, to reduce the interference of noise in the identification and classification of ECG signals, filtering process is done by using wavelet transform filter followed by median filter. Feature extraction was done using cross-convolution kernels of varying sizes and DCNN and better classification accuracy is obtained using Bi-LSTM. To transfer feature information and avoid overfitting, batch normalization and dropout were used. tan mean square error (TMSE) loss function was proposed to improve the classification accuracy. Accuracy of 89.3% and F1 score of 89.1% was obtained.

(D P and N, 2021) conducted the study with SVM classifier for classifying single lead ECG data obtained using IoT-based ECG data acquisition method for early detection of arrhythmia. Pan tompkins algorithm (PTA) was used to reduce noise and to detect QRS complexes. Statistical features like mean, minimum, standard deviation, kurtosis, maximum, skewness and dynamic features including the average heart rate, standard deviation of the R-R intervals were used for generating feature set. Statistical features are measured from the raw ECG signals collected and after detecting QRS complex, heart rate variability was determined from the extracted R-R interval. Classification accuracy was not specified.

(Zihlmann, Perekrestenko and Tschannen, 2017) proposed two architectures for classification of ECG signal data evaluated on Physionet/CINC challenge 2017 dataset. CNN with averaging-based feature aggregation across time and second one includes LSTM for temporal aggregation and CNN for extracting features. Dropout bursts and random sampling are two data augmentation that were used for improving the performance of classification. Of the two proposed architectures, LSTM with CNN performed well obtaining F1 score of 82.1%.

(Yildirim et al., 2019) proposed Convolutional Auto-Encoder (CAE) to compress arrhythmia beats signal size and LSTM takes these compressed signals as input and automatically diagnose arrhythmias. 1-lead ECG data from MIT-BIH arrhythmia database was used for conducting this research. The CAE model obtained percentage root mean square difference (PRD) of 0.7 % and accuracy of 99%.

(Barišić and Jović, 2022) created a 12-lead ECG containing five types of cardiac rhythms from CPSC-Extra, The Georgia 12-lead ECG challenge database and CPSC and used combination of deep learning approaches especially Convolutional Auto-Encoder (CAE) for compressing input signal which served as input to LSTM classifier. Data augmentation was used to deal with imbalance distribution of data after generating synthetic signals using CAE model. Evaluation of CAE model was done using metrics like Root Means Squared (RMS), Compression Ratio (CR) and percentage RMS Difference (PRD). For better reconstruction of signals PRD value should be less which indicates less loss of data. The model obtained 1.63 as overall PRD score. LSTM classifier obtained classification accuracy of 90.7%.

3. METHODOLOGY

This Chapter summarizes the methods and process of implementation of the proposed project. The chapter explains, source of data, how data was denoised, analysis of data and data pre-processing of ECGdenoised data. Followed by description of the models applied to solve the Research question and methods to evaluate the models using 12 lead ECG Denoised data. The web application for identification of Arrhythmias is implemented by applying machine learning models to the uploaded .csv files with 12 lead ECG signal data. The figure below explains the functional flow of Arrhythmias prediction system using 12-lead electrocardiogram.

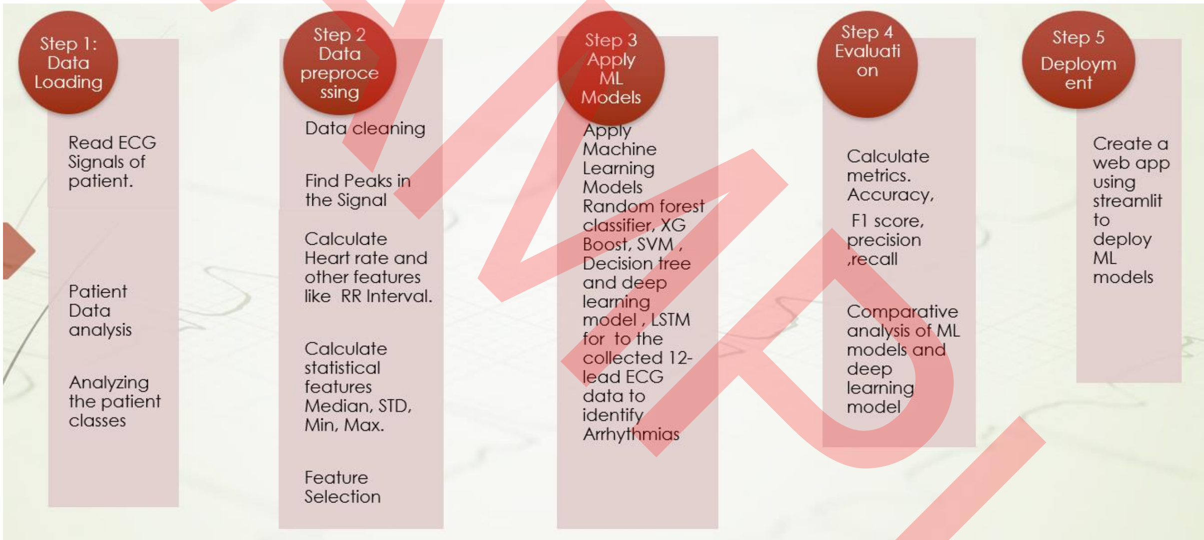


Figure 1- Arrhythmias prediction system functional flow

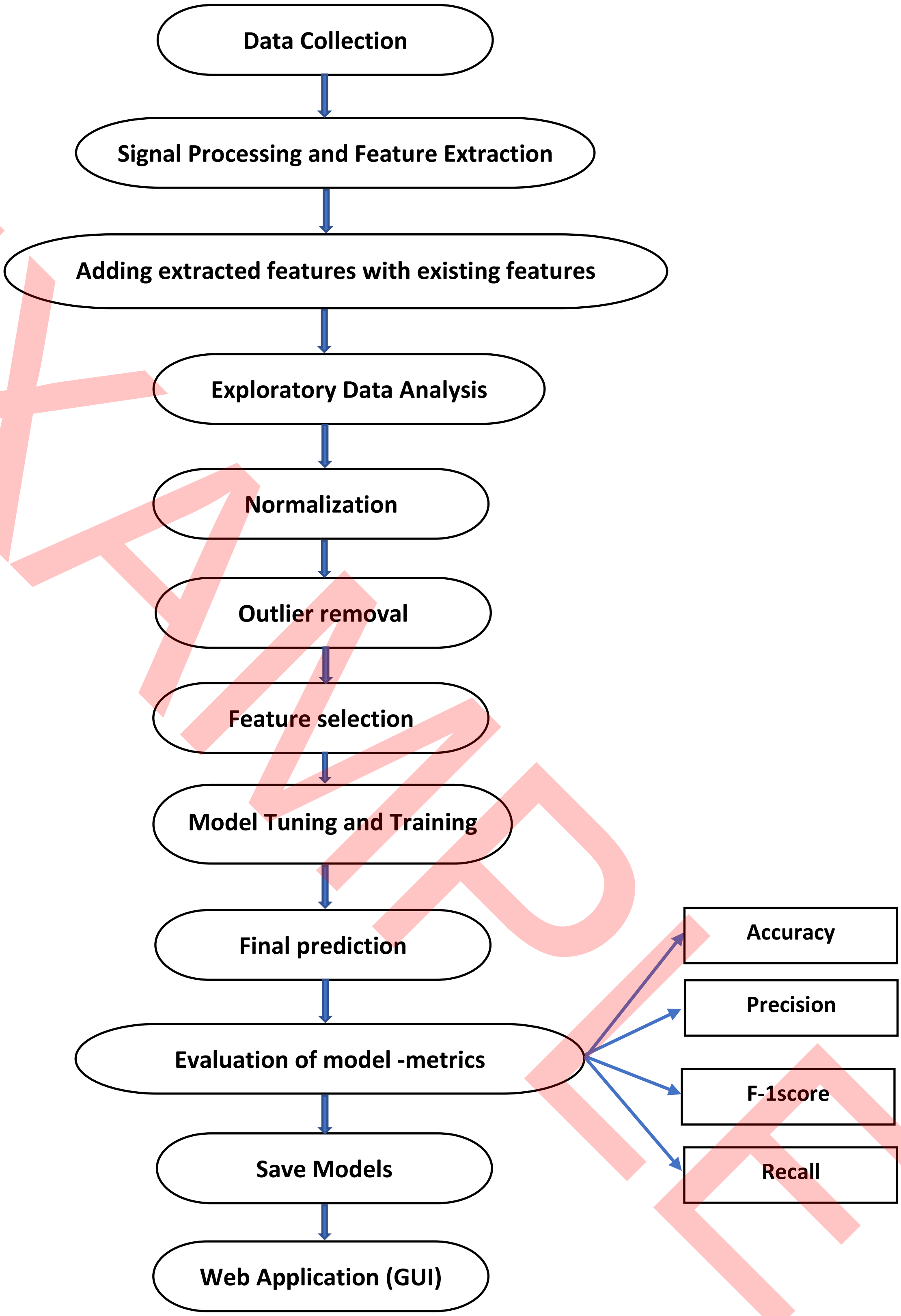


Figure 2 -Conceptual model of Arrhythmias Prediction System

3.1 Resources and project implementation

System Specifications	
OS	Windows 11 Home
Processor	12th Gen Intel(R) Core(TM) i9-12900HK 2.50 GHz
RAM	16.0 GB (15.7 GB usable)
Tools and Software	
IDE	Google Collab
Libraries for visualisation	matplotlib v3.6.2 seaborn v0.12.1 mlxtend v0.21.0
Libraries for Machine learning	scikit-learn v1.1.0 sklearn-model v0.0.6 TensorFlow v2.11.0 Keras v2.11.0
Libraries for GUI	streamlit v1.15 protobuf v3.19.4 pyngrok v5.2.1
Other python libraries	jupyter v1.0.0 pandas v1.5.1 numpy v1.23.4 imblearn joblib v1.2.0 pickle scipy v1.9.3

Table 1 Tools and Technologies used in Implementation

The project implementation will be done following the DCSA approach taught in the course module (Define, Collect, Select, Apply). Define the problem, collect the data, select and then apply the model. Research question is defined in chapter 1.1. collecting the data , selecting and applying models to the data are explained in the following section.

3.2 Data collection

- 12-lead ECG dataset for this study is collected from, part of a large scale 12-lead ECG dataset of Physionet data (10,646 patients) which was initially shared at figshare,

  https://figshare.com/collections/ChapmanECG/4560497/2

  and used Diagnostic.xlsx, RhythmNames.xlsx and ECGDataDenoised.zip folder containing 12-lead ECG data for each patient, saved as csv file for classification.

- These ECGs were all recorded at 500 Hz and were analysed by cardiology professionals to determine whether they contained 11 common rhythms or 67 additional cardiovascular illnesses. Each individual provided a total of 5,000 samples, which were collected from them within 10 seconds. This dataset may be used by researchers who are interested in arrhythmia and other cardiovascular disorders to design, compare, and enhance innovative and traditional statistical and machine learning approaches (Zheng, 2020).

3.3 Data Insight

- ECGDataDenoised.zip folder , denoised ECG data of every patient was saved as CSV file. Each CSV file contains 12 columns representing 12 ECG leads namely Standard Limb Leads: I, II, III , Augmented Limb Leads :aVR, aVL, aVF and Chest Leads: V1, V2, V3, V4, V5, V6 respectively with 5000 rows. Each CSV file was given unique ID .The unique ID of each these files were given 'FileName' as attribute and saved in the diagnostic file. These 12-lead electrocardiograms were (ECGs) taken from 10,646 different people and were all recorded at 500 Hz and analysed by cardiology professionals to determine whether or not they contained 11 common rhythms or 67 additional cardiovascular illnesses . Each individual provided a total of 5,000 samples, which were collected from them within 10 seconds. This dataset created and denoised under the auspices of Chapman University and Shaoxing People's Hospital may be used by researchers who are interested in arrhythmia and other cardiovascular disorders to design, compare, and enhance innovative and traditional statistical and machine learning approaches (Zheng, 2020).

Denoising ECG data

- The power line interference, electrode interface noise, motion artefacts, muscle function, baseline drifting, and random noise were the main contributors of noise contaminants in the ECG data. It is well knowledge that noise presents a formidable challenge to any attempt at statistical analysis and classification accuracy. Since a typical electrocardiogram has a frequency range of 0.5 Hz to 50 Hz, in denoising method performed by dataset providers, a Butterworth low pass filter was employed to eliminate the signal at frequencies over 50 Hz. Later, baseline wandering effects was eliminated with the help of LOESS smoother. Finally remaining noise was handled using the Non-Local Means (NLM) technique.

- The diagnostics file contains each patient's information including filename(unique ID of CSV file of each patient),rhythm, other conditions, gender and age of patient and other ECG summary attributes namely VentricularRate, AtrialRate, QRSDuration, QTInterval, QTCorrected, RAxis, Taxis, QRSCount, QOnset, Qoffset and Toffset.

- RhythmNames file contains 2 columns and 11 rows with names of 11 different rhythms as shown in figure3.

```
[ ] dataframe_rythm
```

	Acronym Name	Full Name
0	SB	Sinus Bradycardia
1	SR	Sinus Rhythm
2	AFIB	Atrial Fibrillation
3	ST	Sinus Tachycardia
4	AF	Atrial Flutter
5	SI	Sinus Irregularity
6	SVT	Supraventricular Tachycardia
7	AT	Atrial Tachycardia
8	AVNRT	Atrioventricular Node Reentrant Tachycardia
9	AVRT	Atrioventricular Reentrant Tachycardia
10	SAAWR	Sinus Atrium to Atrial Wandering Rhythm

Figure 3 – Rhythm names

3.4 Data Pre-processing

- Data pre-processing is one of the important step of data mining .It involves transforming the raw data into useful and understandable format (Jain, D. (2019)). Noise, missing numbers, and an unsuitable format are common in real-world data, making it impossible to feed it directly into machine learning models. A machine learning model's reliability and efficiency may be improved by data pre-processing, which entails cleaning the information and making it acceptable for the model to learn and predict.

- Various steps performed during the pre-processing includes:

  - Import necessary libraries: Libraries such as numpy, pandas, matplotlib, seaborn, imblearn, scipy, sklearn etc were imported.

  - Load the dataset: ECGdenoised csv dataset, diagnostics file and RhythmNames file were loaded and read using pandas.

  - Check the missing values: The dataset is checked for missing values and found no missing values recorded as in fig 4.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10646 entries, 0 to 10645
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   FileName         10646 non-null  object
 1   Rhythm           10646 non-null  object
 2   Beat             10646 non-null  object
 3   PatientAge       10646 non-null  int64
 4   Gender           10646 non-null  object
 5   VentricularRate  10646 non-null  int64
 6   AtrialRate       10646 non-null  int64
 7   QRSDuration      10646 non-null  int64
 8   QTInterval       10646 non-null  int64
 9   QTCorrected      10646 non-null  int64
 10  RAxis            10646 non-null  int64
 11  TAxis            10646 non-null  int64
 12  QRSCount         10646 non-null  int64
 13  QOnset           10646 non-null  int64
 14  QOffset          10646 non-null  int64
 15  TOffset          10646 non-null  int64
dtypes: int64(12), object(4)
memory usage: 1.3+ MB
```

Figure 4 -checking for missing values and information of data set

- Check the distribution of dataset: Since Rhythm names are target variables (class labels), their distribution is plotted to check whether the dataset is balanced or imbalanced . A dataset is said to be imbalanced when there are significant disparities in the way the classes are distributed within the dataset. This indicates that a dataset prefers one of the classes that it contains. If the dataset is weighted more heavily toward one class, then an algorithm that is trained on that dataset will also be weighted more heavily toward that class **(Chaudhary, K. (2020))**. The figure 5 shows that dataset is imbalanced. It can be observed that the count for class SB is the highest while on second number, the highest count is for class AFIB, and so on.

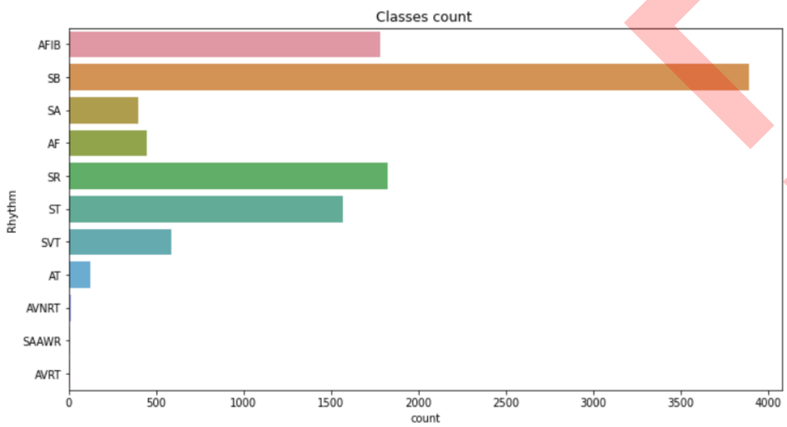


Figure 5 – Imbalanced dataset

- Balancing the dataset: The class imbalance problem can be handled by using one of the following methods depending on the type of dataset one deals with. These methods include Random undersampling, Random oversampling, SMOTE, by choosing the algorithm wisely, by playing with the loss function, or by solving an anomaly detection problem. (Chaudhary, K. (2020)) .

- In this study, the dataset was balanced by random undersampling method and considering random state as 42, using [imbalanced-learn Python library](). The top four class labels with the highest count were taken into consideration. The ST class label, which contains around 1568 rows, has the fewest rows from all the top four labels. Since there are only 1568 rows in the ST label, the same number of rows were considered for top three class labels namely SB, AFIB and SR also and concatenated to form a new dataframe to perform random undersampling and distribution is checked by plotting count plot using seaborn. Figure 6 shows that the dataset is balanced .

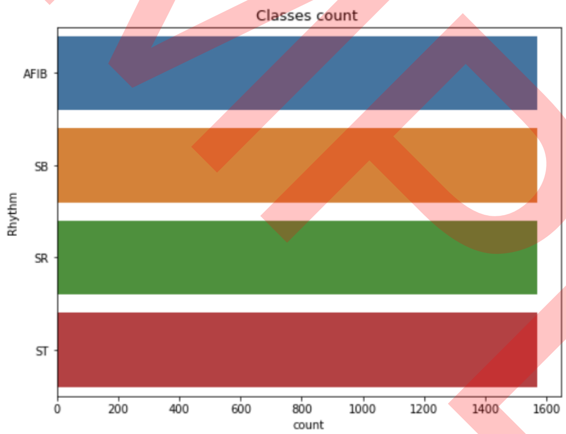


Figure 6 – Balanced dataset

- ECG Denoised data is loaded using pandas and ECG signals of 12-leads of the first record are plotted using matplotlib is shown in figure 7.

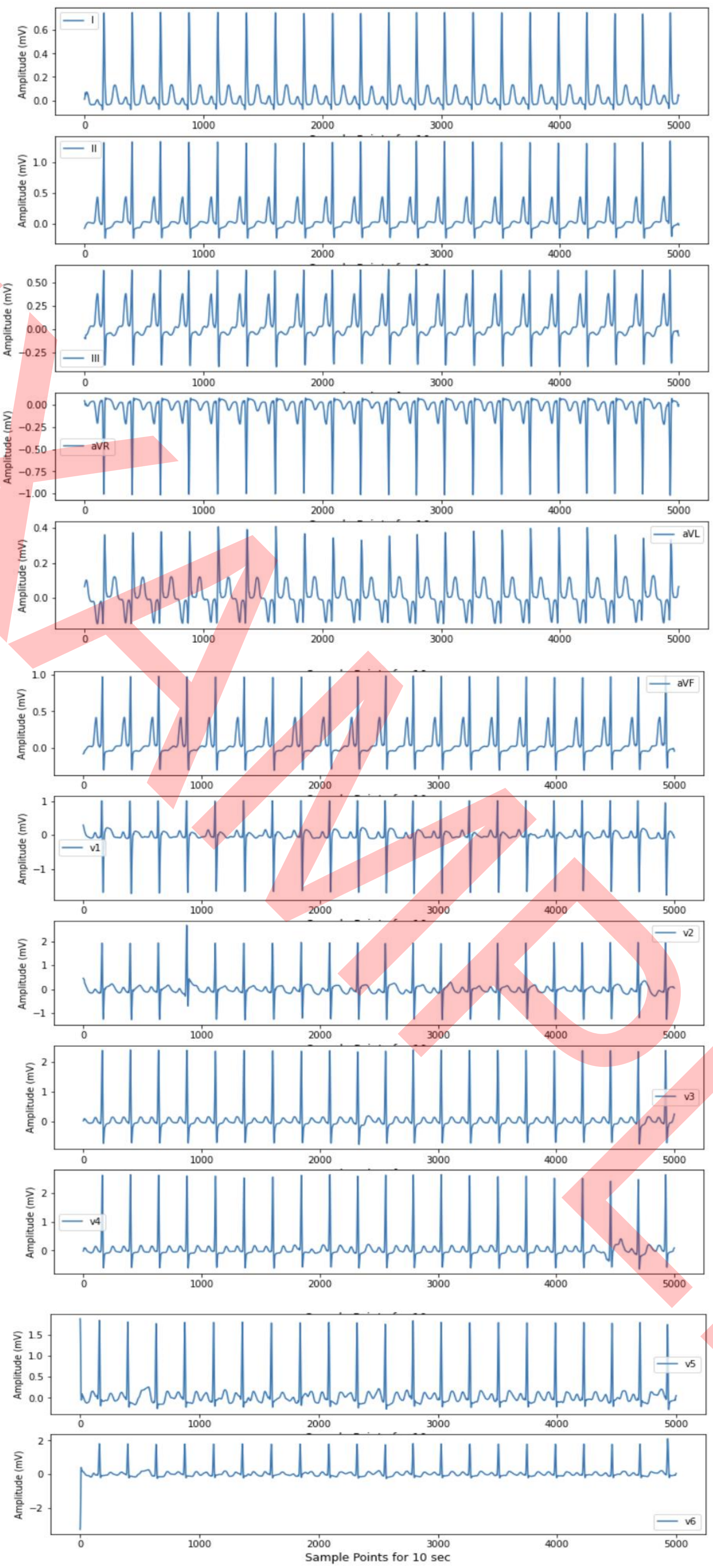


Figure 7 – 12-lead electrocardiogram of first record

3.4.1 Feature Extraction

- Feature extraction is performed to choose and extract diagnostic information and retain the relevant information from the original ECG signal.

- Feature extraction is done by extracting R-R intervals by detecting R- peaks using Template matching.

- Before feature extraction, there are few terms explained that are used in figure – 8 as follows

    - P-wave denotes atrial depolarisation, causing 2 atria of the heart to contract.

    - QRS-complex denotes ventricular depolarisation, causing ventricles of the heart to contract. It is a combination of 3 waves namely Q-wave, R-wave and S-wave respectively. The R-value refers to the peak that occurs in the midst of the QRS complex, and in order to compute the HRV, there is a need to identify this point **(Cai and Hu, 2020).**

    - T-wave denotes ventricular repolarisation, causing ventricles to relax.

    - Every depolarisation is followed by repolarisation that is contraction followed by relaxation. Atrial repolarisation takes place after P-wave and is found within QRS-complex. QRS-complex is a tall structure because ventricles tend to contract stronger than the atria and they tend to mask the atrial repolarisation.

    - RR-interval is the time between 2 R-waves or 2 QRS complexes (**Dr Lewis Potter (2011)).**

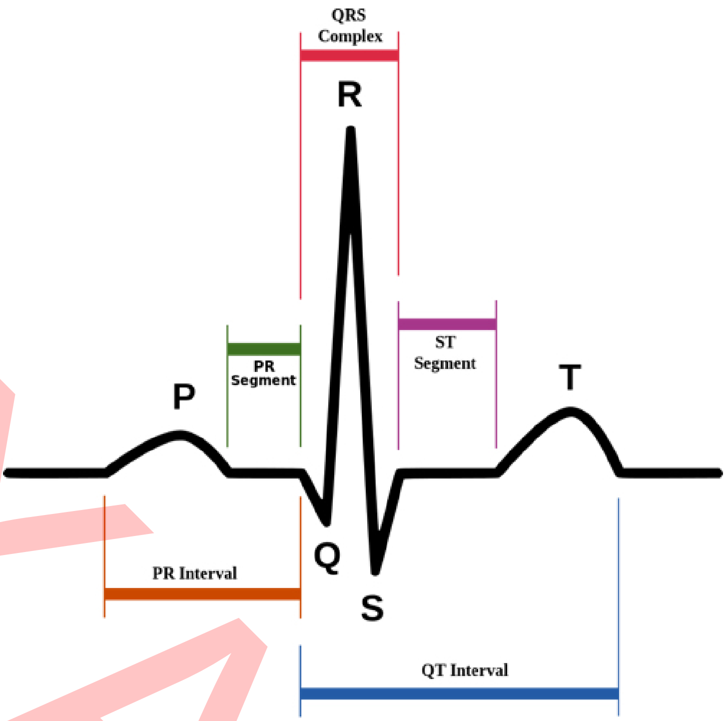


Figure 8 – The components of ECG

- The primary focus of feature extraction is localizing the R-points (sometimes called R-peak identification) in an ECG signal. Interbeat intervals (IBI), sometimes called RR-intervals as shown in figure 9, are calculated by employing these locations as reference. As a shorthand way to refer to regular heartbeats, these intervals are often referred to as NN-intervals.

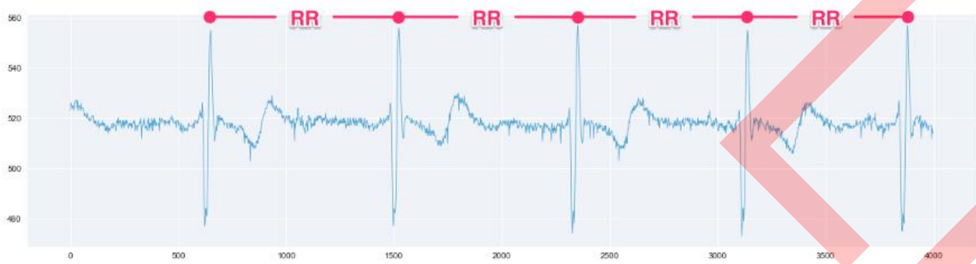


Figure 9 - RR intervals of an ECG signal *(Tetelepta, 2018)*

- Feature extraction is done by extracting R-R intervals by detecting R- peaks using Template matching.

Detecting peaks by matching templates

- In this study R-peaks were detected by template matching using cross-correlation threshold.

- Different QRS detectors may be used for a variety of purposes. Many of them have two parts:

- The First Phase: Modifying the Signal also called Signal Transformation.

    - Create a signal that makes the most of the qualities cared about—in this example, the QRS-complex.

- The Second Phase: Decision Rule

    - To extract the required characteristics from the signal, a threshold is to be used **(Tetelepta, 2018).**

- Most of the peak detectors use a nice approach for signal transformation. Template matching is the term for this process. It is a sophisticated method for locating a target feature in a wider signal by first applying that feature-containing filter (sometimes called a template or kernel).

- The fundamental concept is to move the filter together with the signal and calculate the cross-correlation. In this study a sinewave filter, which is a simplified model of QRS segment was used to find cross-correlation with ECG signal. When a feature in the filter strongly resembles a portion of the signal, it can be said that these two features have a high degree of similarity. In step 2, it discovers traits of interest by establishing a threshold of 0.3 **(Tetelepta, 2018).**

- An ECG data filter is nothing more than a list containing numbers that may be used to discover patterns in the data. For instance, finding positive slopes in a peak may be done using a list like [-1, 1], whereas finding negative slopes can be done with [1, -1].

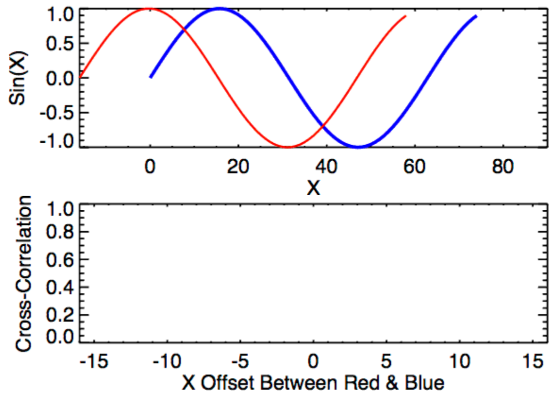


Figure 10 - Cross-correlation between two signals **(Tetelepta, 2018**)

- During the peak detection, multiple peaks were found for each QRS complex. To get a single value for each peak, collection of peaks that are very near(within in the threshold value of 5) are grouped and considered their median as peak index.

- Some RR-intervals are very large, which is around twice the mean and in some cases the RR-intervals are very short. This indicates the presence of outliers .Outliers are detected by choosing RR-intervals with z score greater than 2 and were corrected by replacing with median value of RR-intervals.

- Since Mean RR, SDNN, Mean HR, STD HR, Min HR, Max HR, measured in Beats Per Minute are some of the metrics for HRV (Heart Rate Variability) that are used for time domain analysis, the statistical calculation of these were done by first calculating the heart rate using RR-interval using the standard formula,

$$HR = 60000/RR \qquad \text{---------------(1)}$$

where,

HR = Heart rate

RR = Distance between two R peaks

- Therefore, in feature extraction step, 6 features namely mean RR, STD RR, mean HR, STD HR, min HR, max HR were calculated for each lead of ECG thus obtaining 72 features in total for all the 12 leads of ECG.

- These extracted 72 features were concatenated with 11 existing features namely VentricularRate, AtrialRate, QRSDuration, QTInterval, QTCorrected, RAxis, Taxis, QRSCount, QOnset, Qoffset, Toffset columns from the diagnostics file, thus forming total of 83 features.

3.4.2 Label Encoding

- In this study, label encoding is carried out using sklearn.preprocessing.LabelEncoder where each class label of target variable was encoded with values between 0 and N-1 where N is the total number of class labels. The class labels AFIB, SB, SR, ST are encoded as 0,1,2,3 respectively.

```
Encoded labels are  [0 0 0 ... 3 3 3]
Original form is  ['AFIB' 'SB' 'SR' 'ST']
```

Figure 11 -Encoded Label of Target Variable

3.4.3 Univariate Analysis - Histogram plot and Kernel Density plot

- Distribution, skewness and kurtosis of data were analysed by plotting histogram plots and kernel density plots as shown in figures 12, 13,14. KDE curve smooth things better than histogram and do not swift to zero **(MAINA, S.,2022)**.

- The variables did not follow Gaussian distribution and are exhibiting skewed distribution .

- The variables mean RR ms-I, STD RR/SDNN ms-I, mean HR beats/min-I, std HR beats/min-I, min HR beats/min-I, max HR beats/min-I, STD RR/SDNN ms-II, mean HR beats/min-II, std HR beats/min-II, min HR beats/min-II, max HR beats/min-II, mean RR ms-III, std HR beats/min-III, min HR beats/min-III, mean RR ms-aVF, STD RR/SDNN ms-aVF, min HR beats/min -aVF, STD RR/SDNN ms-aVR, min HR beats/min -aVR, STD RR/SDNN ms-aVL, min HR beats/min -aVL, mean RR ms-V1, STD RR/SDNN ms-V1, min HR beats/min-V1, mean RR ms-V2, STD RR/SDNN ms-V2, min HR beats/min-V2, mean RR ms-V3, STD RR/SDNN ms-V3, min HR beats/min-V3, mean RR ms-V4, STD RR/SDNN ms-V4, min HR beats/min-V4, mean RR ms-V5, STD RR/SDNN ms-V5, , mean HR beats/min-V5, std HR beats/min-V5, min HR beats/min-V5, max HR beats/min-V5, STD RR/SDNN ms-V6, mean HR beats/min-V6, std HR beats/min-V6, min HR beats/min-V6, max HR beats/min-V6,VentricularRate, AtrialRate, QRSCount are showing positive skewness while std HR beats/min-III, mean HR beats/min-III, max HR beats/min-III, std HR beats/min-aVR, max HR beats/min-aVR, mean HR beats/min-V3, std HR beats/min-V3, max HR beats/min-V3 are bimodal.

Figure 12 – Histogram plot of the features

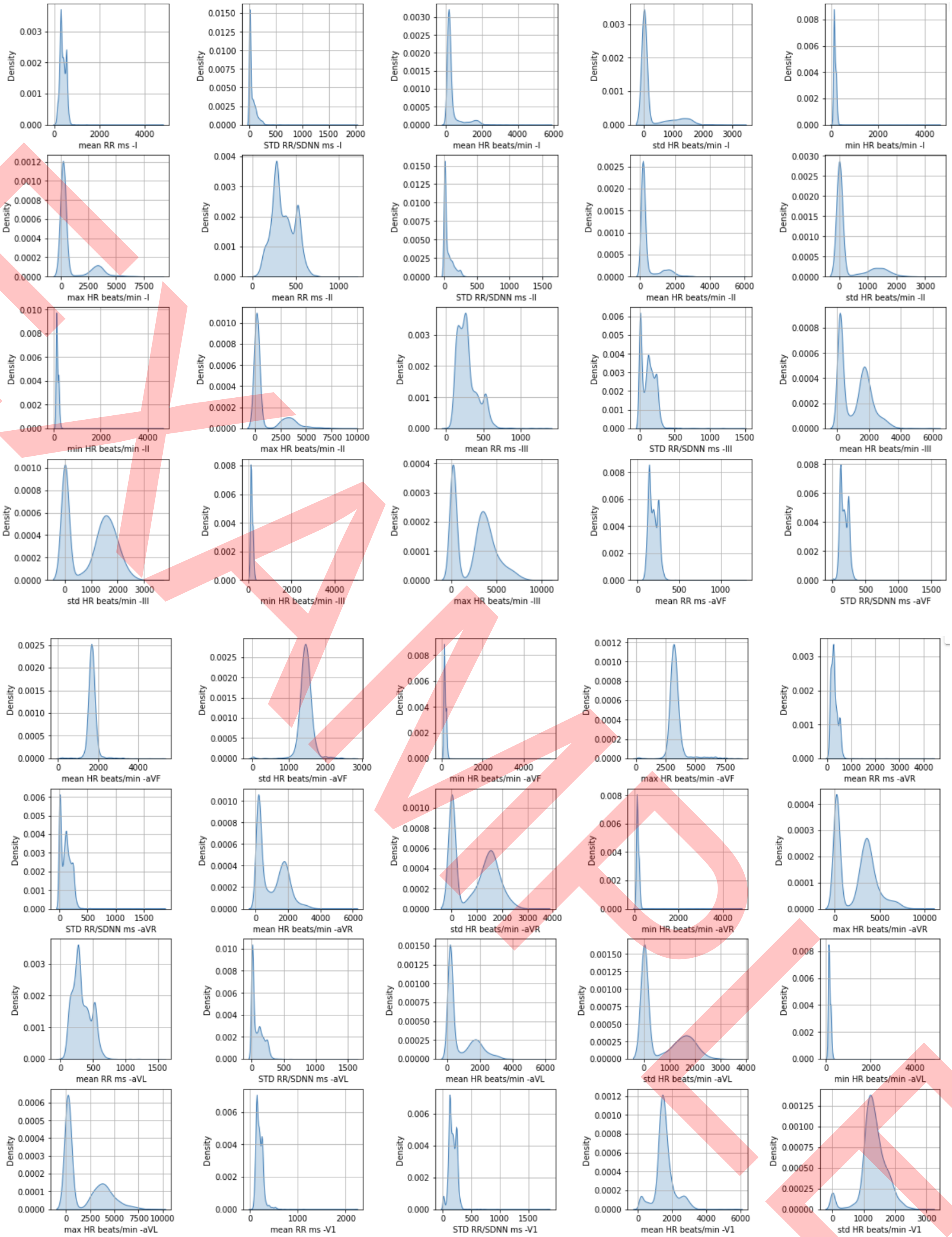


Figure 13 – Kernel Density Plot of the variables

Figure 14- Kernel Density Plot of the variables

3.4.4 Univariate Analysis - Box Plot

- The Box plot is plotted to analyse the distribution of data and presence of outliers. Figure -15 shows the presence of outliers .

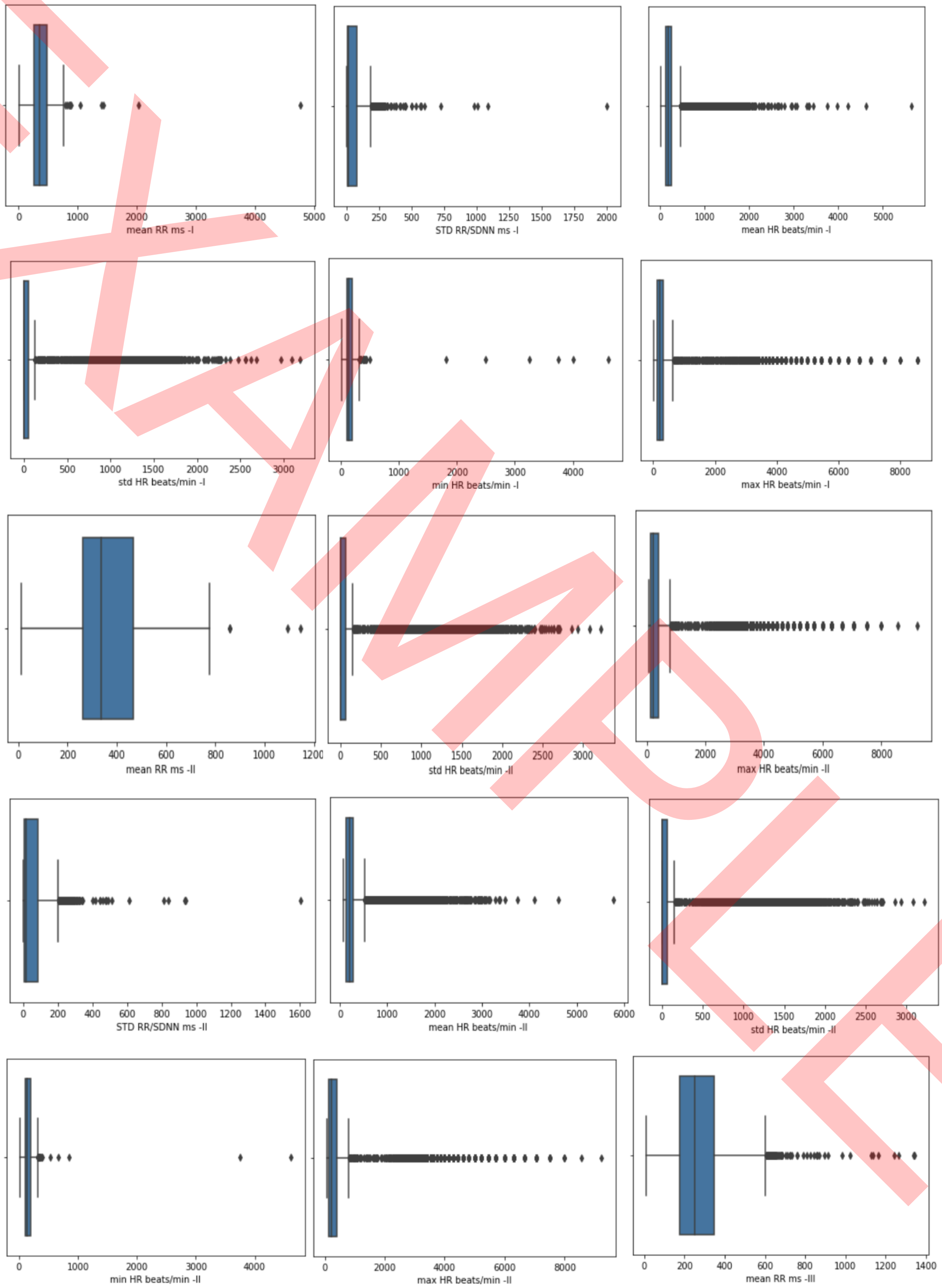


Figure 15 - Box plot of features before outlier treatment

3.4.5 Multivariate Analysis – Correlation matrix

- Correlation between the extracted features is checked using Pearson correlation and correlation matrix heatmap is plotted using seaborn library as shown in figure 16.
- Pearson Correlation coefficient (r) value 0 signifies no correlation ,while value +1 signifies positive correlation and value -1 signifies negativecorrelation **(Turney, 2022).**
- From figure 16 it can be observed that there exists a strong correlation between QTInterval and Toffset, QRSCount and Taxis, max HR beats/min – V6, V5, V4,V3,V2, aVL, III, II, I and std HR beats/min – V6, V6, V5, V4,V3,V2, aVL, III, II, I respectively, QRSCount and VentricularRate, max HR beats/min – V2, aVL, II, I and mean HR beats/min – V2, aVL, II, I respectively, min HR beats/min – aVF and min HR beats/min – II while rest other variables have weak correlation.

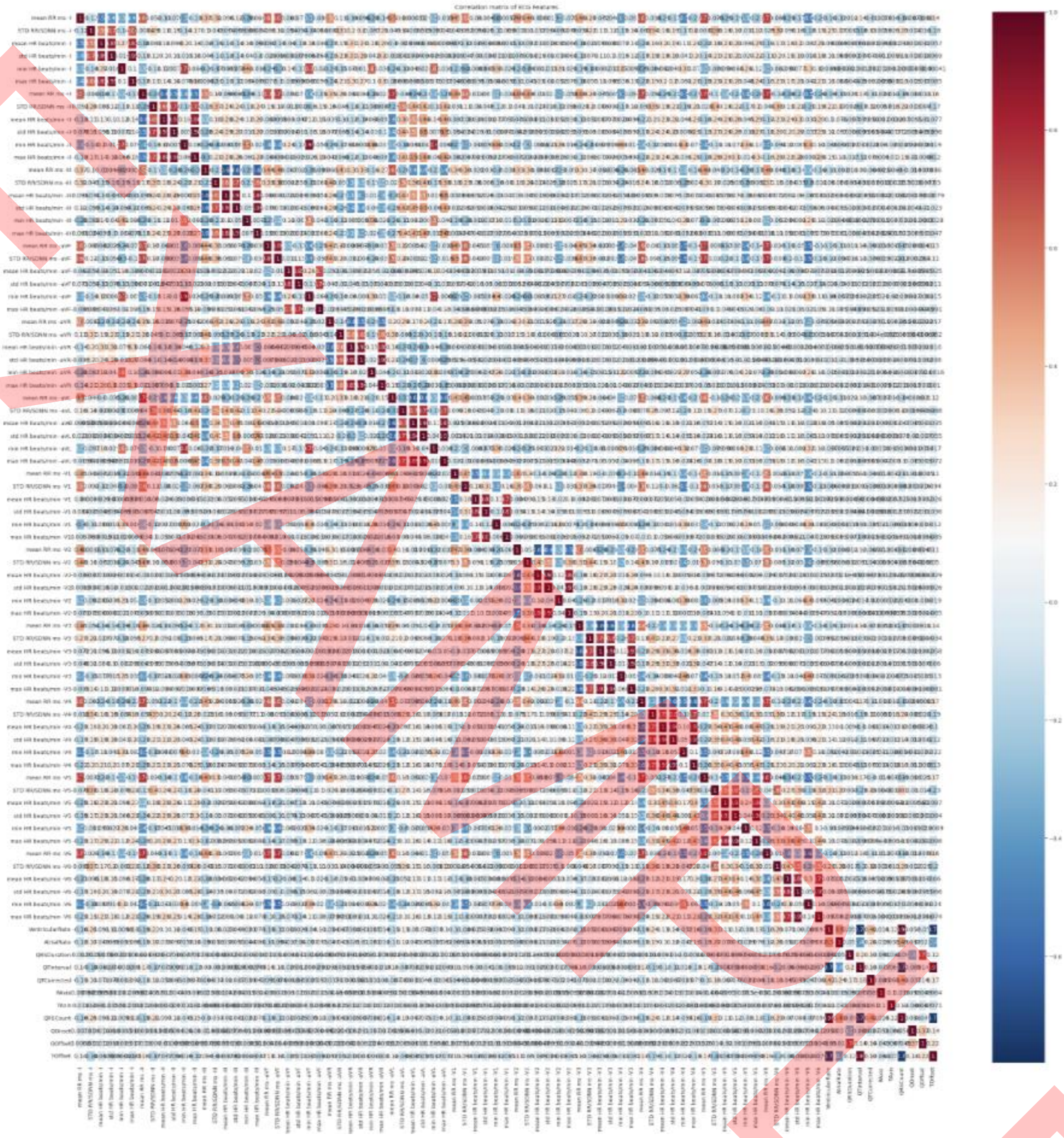


Figure 16 - Correlation Matrix heatmap between features of ECG data

3.4.6 Data Transformation Using l2 Normalisation

- The data was normalised by l2 normalization using sklearn.preprocessing. Normalizer that transformed data to a common scale with sum of squared values of each row to unit variance. As a result, all the features were given same weights and treated equally by the model and thus increasing the model performance.

```
# After normalizing
X_normalized
```

```
array([[0.03944683, 0.00730746, 0.01297003, ..., 0.01888957, 0.02406604,
        0.03505468],
       [0.0419123 , 0.00987763, 0.01330173, ..., 0.02024067, 0.02372395,
        0.03718634],
       [0.00630185, 0.00769756, 0.22248044, ..., 0.01911037, 0.02288215,
        0.03654439],
       ...,
       [0.03083534, 0.01106517, 0.01405276, ..., 0.01835229, 0.02100854,
        0.03662409],
       [0.03170997, 0.00725768, 0.05060635, ..., 0.02660337, 0.03161386,
        0.05141727],
       [0.02575122, 0.00949891, 0.03118331, ..., 0.01853254, 0.02239714,
        0.03407879]])
```

Figure 17- Features after l2 Normalization

```
# Sum over the rows.
X_sum_squared = np.sum(X_squared, axis=1)
print(X_sum_squared)
```

```
[1. 1. 1. ... 1. 1. 1.]
```

Figure 18 – Code showing that features are l2 Normalized

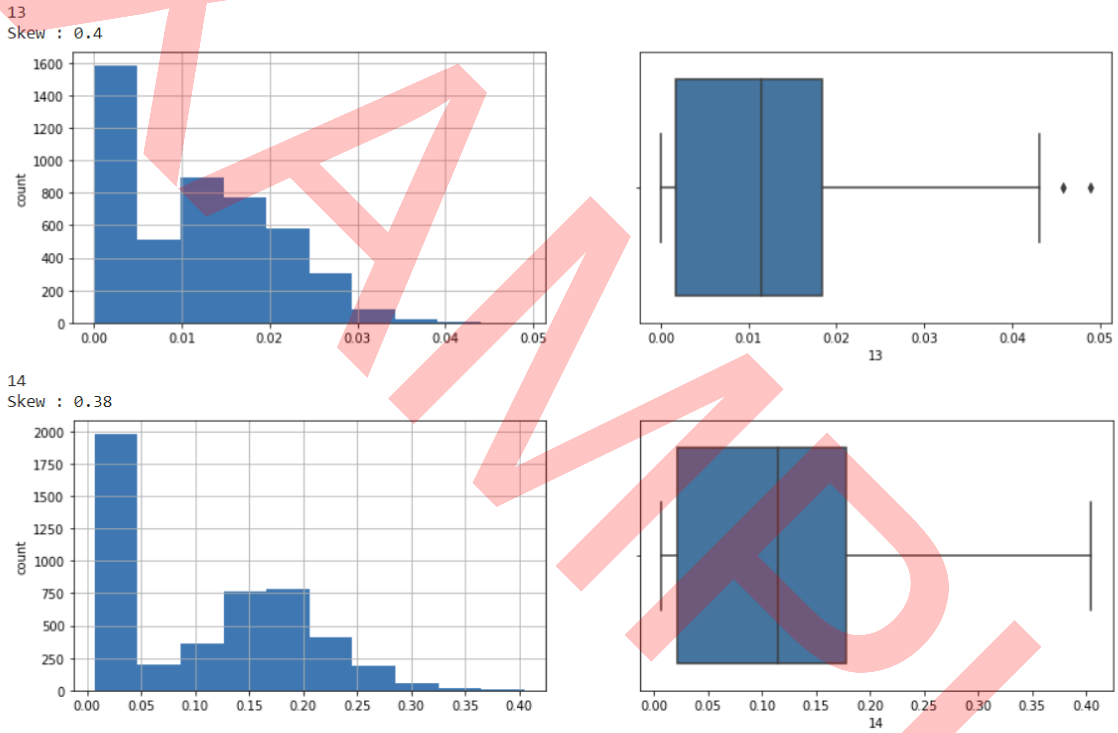
3.4.7 Outlier Treatment

For outlier treatment , the first approach used was removal of the outliers by using Interquartile range method. But the number of rows left after outlier removal is very less. As a second approach outlier removal using standard deviations was chosen. In this study the top 0.1% of extreme cases were removed by considering using 4 standard deviations . One may identify and exclude outliers from the dataset by setting the set boundary at 2 times the standard deviation or 3 times the standard deviation **(Stephen Allwright(2022)).**

- Upper limit = mean + 3 * std --------------------(2)
- Lower limit = mean — 3 * std --------------------(3)

- The histogram plot and box plot after outlier removal are shown in figure 19.

Figure 19 – Histogram Plot and Box Plot after Outlier removal for min HR beats/min-II and max



HR beats/min-II

3.4.8 Feature Selection

- Feature selection process is concerned with the selection of features that contribute to the prediction or output ,either automatically or manually. It is a score set down to each feature between 0 and 100, indicating how important is that particular feature in predicting the class label **(Mazzanti, S. (2022))**. By using feature selection, machine learning models can be trained rapidly, can reduce overfitting of the model, accuracy of the model can be improved by choosing appropriate subset, complexity of the model can be reduced, and interpretations can be made easier **(freeCodeCamp.org. (2021)).**

- In this study Univariate feature selection is done by selecting the **k** best features, from features with **k** highest score by using SelectKBest based on f_classif a univariate statistical test using sklearn.feature_selection.f_classif. In this study for final analysis, from 83 features ,using SelectKBest and f_classif 31 features were selected based on the **k** highest score. The feature importance of selected 31 features are plotted against the feature importance score as shown in figure 20.

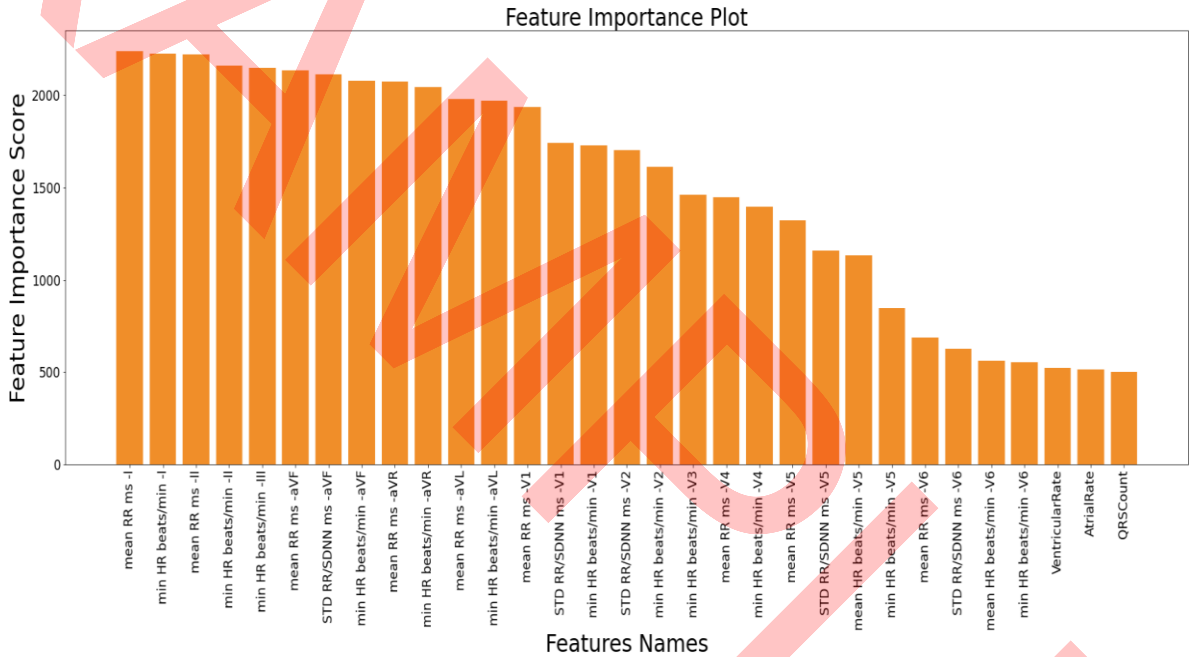


Figure 20 – Feature importance plot based on **k** highest score

3.4.9 Split dataset into train and test set

- Dataset is split into train and test set in the ratio of 80:20 using sklearn.model_selection.train_test_split from a robust library, sklearn for machine learning in Python with the random state set to 15 and the performance of the machine learning and deep learning models are evaluated. Predictions are done using test set by fitting the model with train set.

3.5 Model Development

3.5.1 Select Machine learning Algorithms

- Model development step is performed after data pre-processing step, which is essential for better performance of the model. It involves selecting the appropriate algorithms based on the use case and applying the model to the pre-processed data and evaluate the performance of the model using metrics.

- In this study, model learns from the labelled training data and predictions are made on unseen or test data, it is Supervised learning. Different types of Arrhythmias were predicted by the model, so it is a classification problem (multiclass classification).

- Since ,it is a classification problem, classification algorithms Logistic Regression, Support Vector Machine , Decision tree classifier, Random Forest Classifier, XGBoost and Gradient boosting algorithms were chosen to predict the input data belongs to which category of target variable or class label.

- **(Cheng, Zou and Zhao, 2021)** proposed ECG signal classification by deep CNN (DCNN) and BiLSTM using single-lead data from 2017 Physionet/CINC Challenge ECG dataset. After screening out too short signals ,7561 out of 8528 samples of data was finally used for analysis to classify ECG signals into normal, AF, other rhythm and noisy. Accuracy of 89.3% and F1 score of 89.1% was obtained. This holds true for large dataset of ECG signals. Hence CNN and Bi-LSTM are chosen to check the performance on the dataset collected for this study.

- The selected machine learning classifiers and deep learning models will be discussed in this session.

Logistic Regression

- Logistic regression is meant for binary classification, predicting the target variable using binomial probability distribution function. The target variable is mapped to 1 for positive class and 0 for negative class.

- Logistic regression by default can be used only for classification tasks that have 2 class labels. In case of multi-class classification problems two approaches are used. In first approach, before applying logistic regression model to multi-class classification problem, it is split into many binary classification problems by using One-Vs-Rest strategy. In the second approach, logistic regression model is changed in order to do predictions of class labels directly. Particularly, predictions are done to know the probability of the input record belongs to a specific class label **(Brownlee, J. (2020)).**

- In Multinomial logistic regression, the loss function is cross-entropy loss and output would be one probability for each class label. Logistic regression measures the relationship between one or many predictor variables and categorical class of target variable labels, using sigmoid or logistic function by evaluating the probabilities **(Parashar, P. (2020)).** Graphical representation of Logistic regression is shown if figure 21.

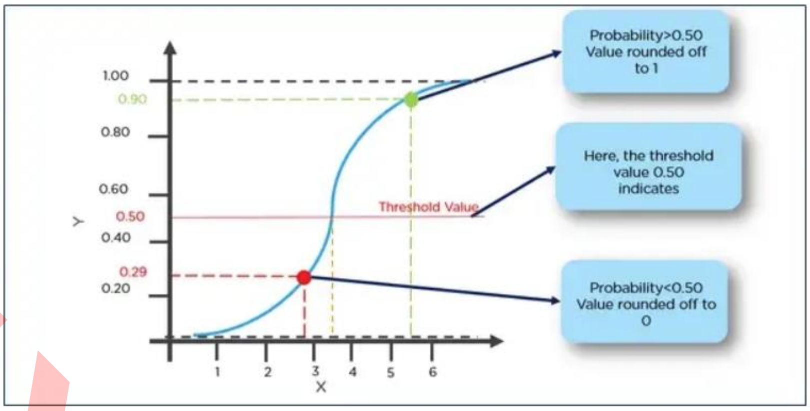


Figure 21 – Graphical representation of Logistic Regression

$$y = e^{(b0 + b1*x)} / (1 + e^{(b0 + b1*x)}) \quad \text{.. (4)}$$

Where:

- x is the input value
- y is the predicted output
- b0 is the bias or intercept term
- b1 is the coefficient for the single input value (x) **(Brownlee, J. (2016))**
- ß value or coefficient b is to be learned from train set which will be accompanied with every column of input data.

Decision Tree

- A decision tree is a non-parametric supervised algorithm that can predict possible outcomes from the given input by using particular rules in the process of making decision. It can be used for either classification or regression tasks.

- A decision tree has a structure similar to that of a tree with
  - The root node or parent that denotes whole sample, which then further splits into uniform sets of sub-nodes or internal nodes or child nodes.

    - The Internal node divides further into more sub-nodes with branches connected to either other sub-nodes or leaf nodes.

    - Leaf nodes or terminal nodes have no sub-tree or branch extending out of them **(Manav (2020)).**

- Decision tree structure resembles a flow chart structure and helps anyone to understand the reason behind the decision made. Decision tree structure is shown in figure 22.

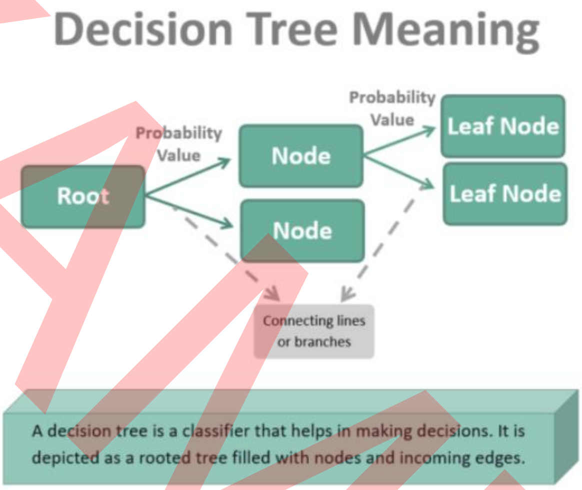


Figure 22 - Decision Tree Structure **(Team, W. (2022))**

- Algorithms ID3 ,C4.5 that were developed by Ross Quinlan and the algorithm CART developed by Leo Breiman can be used for building decision trees. Metrics for ID3 algorithm are entropy and information gain while that for C4.5 is information gain or gain ratios for evaluating the split points and CART uses Gini impurity **(IBM (n.d.))**

Support Vector Machine

- Support Vector Machine (SVM) is a supervised Learning algorithm, that can be used for both classification and regression tasks. But mostly it is used for classification tasks.

- The main aim of SVM is to locate a decision boundary or hyperplane which clearly segregates the data points in high or infinite dimensional space. The chosen hyperplane will have maximum margin (distance)or functional margin between the points or vector of the two classes to be classified **(scikit learn (2018)**. Hyperplane and maximum margin is shown in figure 23.

- The term Support Vector Machine is given to the algorithm because in locating a hyperplane, it selects the utmost points or vectors known as support vectors. These points or vectors will be nearer to the decision boundary.

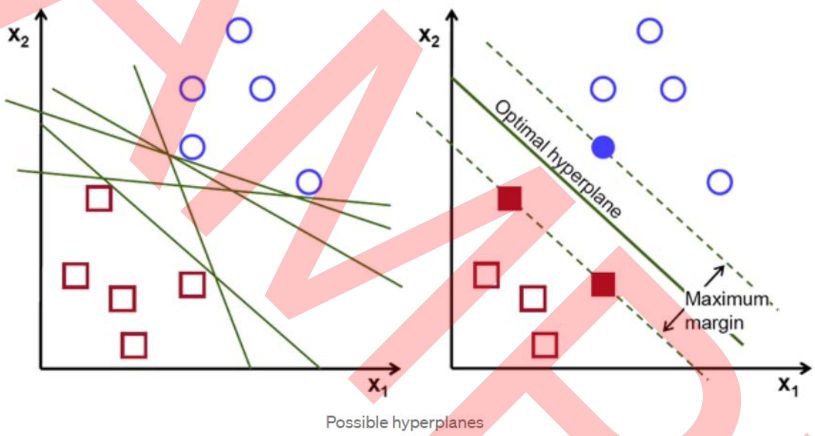


Figure 23 - SVM – showing hyperplane **(Gandhi, R. (2018))**

- Linear SVM an Non-Linear SVM are the two types of SVM.
    - If the data is linearly separable means if the two classes of dataset can be separated by SVM with a straight line ,it is called Linear support vector machine classifier.
    - If the data is non-linearly separable means if the two classes of dataset cannot be separated by SVM with a straight line ,it is called Non-Linear Support vector machine classier.

- If data is non-linearly separable, in such cases Kernelized SVM can be used. Generally, if the data is non-linearly separable ,to make it linearly separable the data is mapped to a higher dimension . For example, from 1-Dto 2-D. But SVM uses Kernel trick to separate the data points even when the feature space is infinite-dimensional without actually transforming data points or vectors to high dimensional space. The two popular kernel functions used are Radial Basis Function Kernel(RBF) and Polynomial Kernel **(GeeksforGeeks. (2020)).**

Random Forest

- Random Forest is a supervised Learning algorithm, that can be used for both classification and regression tasks. As it is well known, with more the number of trees more robust the forest will be. In the same manner the accuracy of the Random Forest will be high with higher number of trees in the algorithm.

- Random Forest Classifier possess lot many uncorrelated decision trees that function as ensemble on profuse subset of the dataset obtained by bootstrapping to solve complex problem. Each decision tree in the random forest makes class prediction and the model's prediction is done by considering the class with majority of votes. Whereas in case of Random Forest Regressor, the prediction is done by taking the average of predictions made by each decision tree**(Polamuri, S. (2017)).**

- The number of features sampled , the number of trees and node size are the three main hyperparameters to be set for Random Forest before applying on the training data.

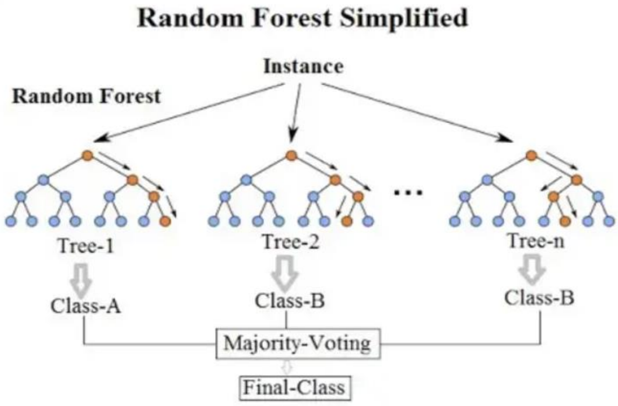


Figure 24 – Random Forest Simplified (Koehrsen, W. (2020))

Gradient Boosting Classifier

- Gradient Boosting developed by Friedman is a supervised Learning algorithm, that can be used for both classification and regression tasks. In this algorithm ,models are built sequentially, and the succeeding models will try to minimize the errors of the preceding model.

- Gradient Boosting Regressor uses Mean squared error whereas Gradient Boosting Classifier uses log loss as loss function. The main aim of Gradient Boosting algorithm is to add weak trees using gradient descent and minimize the loss function **((Saini, 2021)).**

- In Gradient Boosting algorithm, trees ate build by selecting the best split points either by using purity scores such as Gini index or to minimize the loss function. By minimizing the parameters, the error can be decreased by ensuring that the existing tree will not get changed upon addition of a tree.

- Optimizing the loss function, weak learner, additive model are the three main requirements for building Gradient Boosting algorithm.

XGBoost Classifier

- Extreme Gradient Boosting Algorithm also known as XGBoost was designed by Tianqi Chen can be used for both classification and regression tasks and to resolve ranking problems also **(Vishal Morde (2019)).**

- XGBoost is an ensembled supervised machine learning algorithm that uses optimized Gradient Boosting algorithm through parallel processing, out-of-core computation, tree-pruning by using max_depth as parameter , handling sparse data like data with missing values, and regularisation to penalize the model complexity in order to avoid overfitting and bias, thereby enhancing the speed and performance of the model **(Team, G.L. (2020)).**

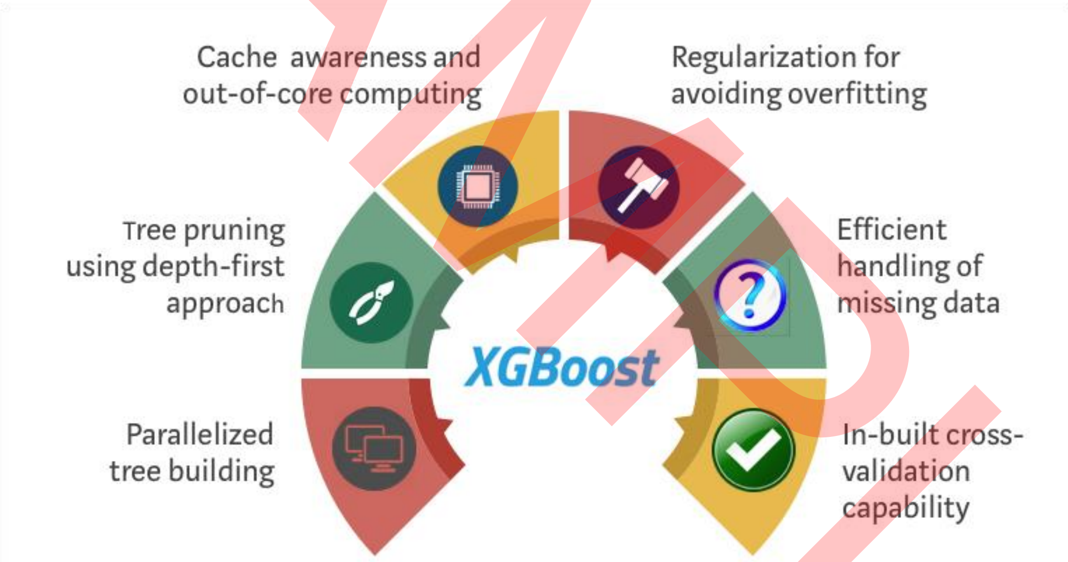


Figure 25 – Methods used by XGBoost to optimize standard Gradient Boosting algorithm **(Vishal Morde (2019))**

Deep Learning Model – CNN and Bi-LSTM (Hybrid Model)

Convolutional Neural Network (CNN)

- Convolutional Neural Network (CNN) is a type of deep learning algorithm that uses certain weights and biases to identify and distinguish features in the input it receives. 2D CNNs are used for processing 2-dimensional data, like images and videos, while 1D CNNs are used for processing 1-dimensional data like signal data.

- For 1D CNN computational cost of using 1D CNN is lower when compared to 2D CNN. Due to this,1D CNN is better suited for real-time and low-cost applications, especially on mobile devices where resources are limited. **(Kiranyaz, Ince and Gabbouj, 2016).**

- In this study, a combination of 1D CNN and Bi-LSTM is used. Convolution layer, Rectified Linear Unit, Batch Normalisation, Max pooling layer were used together in 1D CNN for extracting features from ECG signal data. The hyper parameters used to tune CNN to optimize the performance of the model are filter size used in convolution layers, included max pooling, ReLU activation function used in every CNN layer **(Rahul, J. and Sharma, L.D. (2022)).**

- A Convolution layer performs the process of convolution by applying a filter to an input in order to generate an activation. If the same filter is applied multiple times to the input, it produces a map of activations called a feature map, which highlights the locations and intensity of a specific feature in the input **(Jason Brownlee, 2019).**

- A non-linear activation function, ReLU is applied on the feature map to introduce non-linearity into the model. This allows the model to learn complex representations of the data and improve its performance.

- A convolution layer is followed by a pooling layer in the CNN architecture. The pooling layer helps to decrease the spatial dimensions of the convolved feature maps there by reducing the computational costs and thus making the model robust to small translations of the input data. In the proposed CNN model max- pooling is used. From the feature map, max pooling takes element of relatively great size. **(Gurucharan, M. (2020)).**

- In the proposed model, Batch normalisation is used in order to normalize the activations of a layer which helps in reducing the internal covariate shift that is caused due to change in input distribution during the training process. Since Batch Normalisation address the exploding and vanishing gradients problems, substantial learning rates can be used. It also helps in reducing overfitting of the model **(Vinod, R. (2020)).**

- ECG data was fed into the CNN architecture containing alternate layer of 1 dimensional Convolutional layer and 1 dimensional max pooling layers. Batch normalisation is placed after ReLU .The performance of the model shown improvement on increasing the dense layers and epochs. The features extracted from the model were given as input to the Bi-LSTM for further classification.

Bi-directional LSTM (Bi-LSTM)

- Bi-directional LSTMs are a variation of standard LSTMs that are designed to capture long term dependencies in the data and improve performance of model on sequence classification problems.

- LSTM networks were introduced by Horchreiter and Schmiduber and are largely used for tasks involving sequential data modelling because of their ability to maintain and propagate context through time. This allows them to better handle sequences of varying lengths and make predictions based on the previous inputs in the sequence.

- LSTM is the higher version of RNN that has the ability to back propagate through time and layers that helps in solving the 'short memory' problem and long-term dependencies are learnt effectively. The series of gates with individual RNN lets the LSTM either overlook or hold the datapoints.

- LSTM is efficient in solving two considerable issues associated with RNN like exploding gradient and vanishing gradient problems. Exploding gradient issue occurs during the procedure of weight update where weight changes are enormous or surplus. Vanishing gradient issue also occur during the weight update procedure where the weight changes are minute and makes tough for gradients updating thereby leading to extended training time. LSTMs are able to maintain a constant error across multiple timesteps, allowing them to continue learning ***(Olah, C. (2015))***.

- LSTM unit have a cell that stores the information, as well as three gates that regulate the flow of information into and out of the cell. The gates are the forget gate, input gate and the output gate. The Cell structure of LSTM is shown in figure 26.

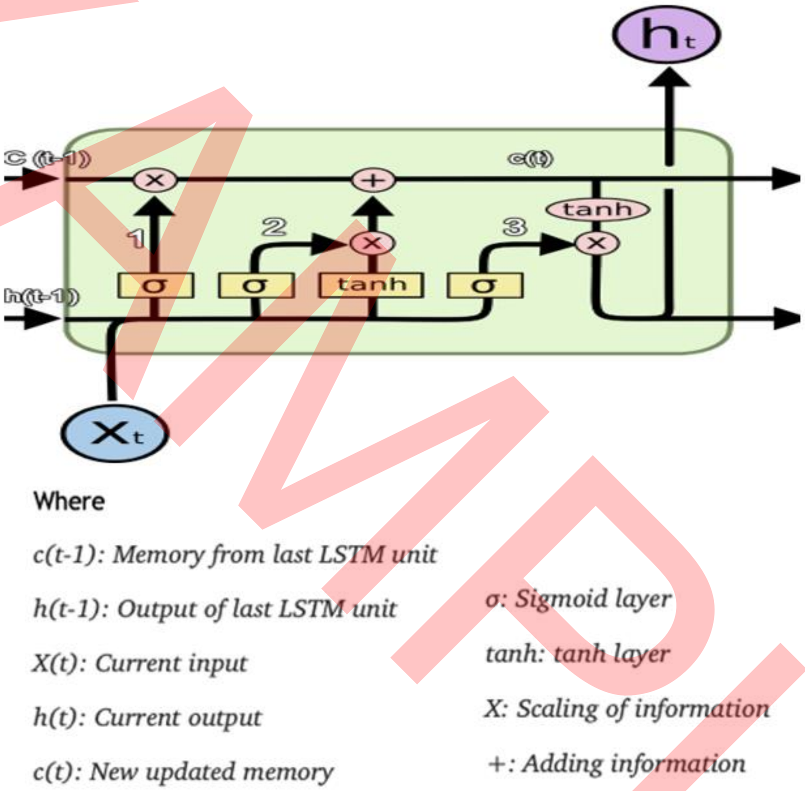


Figure 26 – Cell Structure of LSTM **(Olah, C. (2015))**

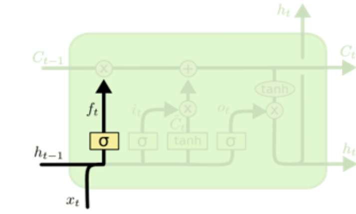
- The second derivative of tanh has the ability to sustain for long-term before it goes to zero, thus making tanh ,an appropriate function to control the vanishing gradient problem.

- A sigmoid function is used when information is to be retained or to be forgotten because it can give 0 or 1 as output , where 0 indicates 'remove it' and 1 indicates 'hold back it'.

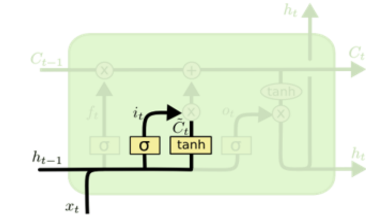
- The forget gate f(t) present in the LSTM architecture, directs the amount of information that has to be removed from the preceding cell state. The sigmoid layer of forget gate takes X(t) and h(t-1) as input and gives 0 as the output after deciding the portion of old output which is to be discarded . f(t)*c(t-1) is the output of the forget gate f(t).

- The input gate present in the LSTM architecture, directs the amount of information that has to be added to the current state of the cell. The sigmoid layer of input gate takes the new input X(t) and deduce the information that is to be stored in the cell state or to be removed. Then a vector of all value from new input is generated by tanh layer, helps in updating memory.

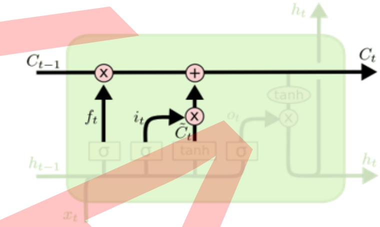
- The sigmoid layer at the output gate decides the amount of information that is to be passed on to the next layer of network **((Sinha, 2018)).** Step by step process taking place in LSTM unit is expressed mathematically as shown in figure 27.



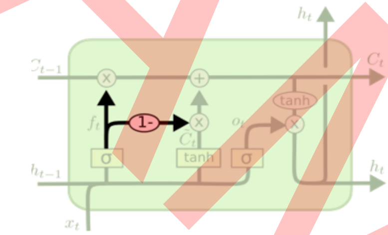
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



$$C_t = f_t * C_{t-1} + (\mathbf{1} - \boldsymbol{f_t}) * \tilde{C}_t$$

Figure 27 – LSTM – Step by step information **(Olah, C. (2015))**

Where,

ft = activation vector of the forget gate

W = weight matrices

b = bias vector

it = activation vector of the input gate

C~t = activation vector of the cell input

tan*h* = hyperbolic tangent function

ct = cell state

ot = activation vector of output gate

ht = output vector of the LSTM block

- Bi-directional LSTM processes the input sequence in two directions ,both forwards and backwards thereby allowing the network to capture information from the past and future context of each time step in the sequence. The outputs obtained from the two LSTMs is concatenated before passing it on to the next layer. Bi-LSTM network is shown in figure 28.

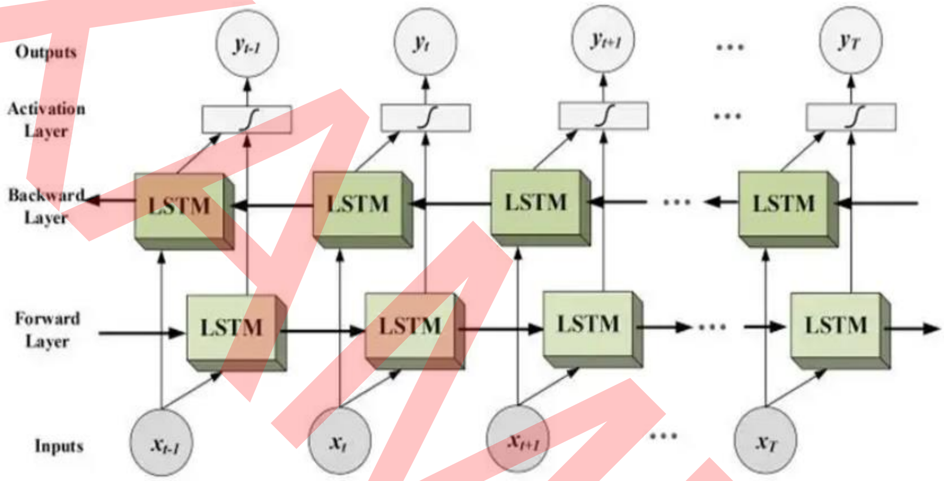


Figure 28 – Bi-LSTM network **(Mungalpara, J. (2021))**

- ECG data from CNN with reduced features obtained as output was applied to Bi-LSTM g ,where temporal information is extracted and classification of data into 4 different classes namely AFIB, SB, ST,SR(4 rhythm classes) is done using fully connected network.

- Since it is a multiclass classification problem, Softmax function is applied for the final dense layer of Bi-LSTM. During the compilation of the model, to slash the loss function, Adam optimizer is chosen as it consumes less memory and effective with enormous parameters and categorical cross entropy as the loss metric and accuracy as evaluation metric are chosen. The summary of CNN and Bi-LSTM model is shown in figure 29 and it is plotted using the keras plot_model as shown in the figure 30.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 31, 64)            448

 batch_normalization (BatchN  (None, 31, 64)           256
 ormalization)

 max_pooling1d (MaxPooling1D  (None, 16, 64)           0
 )

 conv1d_1 (Conv1D)           (None, 16, 64)            24640

 batch_normalization_1 (Batc  (None, 16, 64)           256
 hNormalization)

 max_pooling1d_1 (MaxPooling  (None, 8, 64)            0
 1D)

 conv1d_2 (Conv1D)           (None, 8, 64)             24640

 batch_normalization_2 (Batc  (None, 8, 64)            256
 hNormalization)

 max_pooling1d_2 (MaxPooling  (None, 4, 64)            0
 1D)

 conv1d_3 (Conv1D)           (None, 4, 64)             24640

 batch_normalization_3 (Batc  (None, 4, 64)            256
 hNormalization)

 max_pooling1d_3 (MaxPooling  (None, 2, 64)            0
 1D)


 bidirectional (Bidirectiona  (None, 256)              197632
 l)

 dropout (Dropout)           (None, 256)               0

 dense (Dense)               (None, 50)                12850

 dense_1 (Dense)             (None, 4)                 204

=================================================================
Total params: 286,078
Trainable params: 285,566
Non-trainable params: 512
_________________________________________________________________
```

Figure 29 – CNN and Bi-LSTM model summary

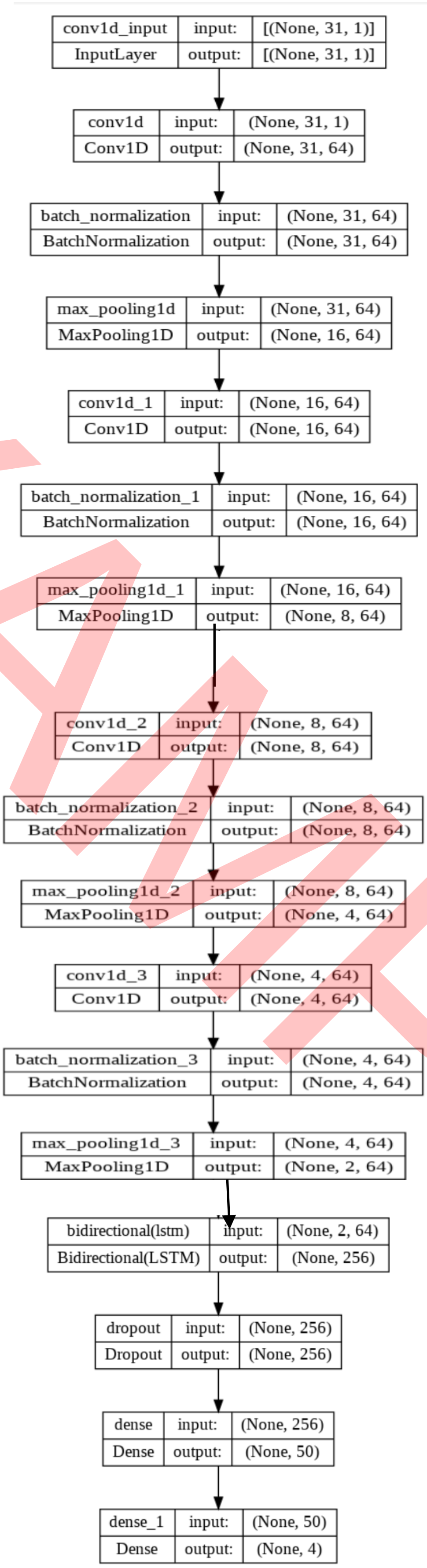


Figure 30 - CNN and Bi-LSTM model summary plot

3.5.2 Fit the models

- Model fitting is the next step after pre-processing the data and selecting the suitable models. It is a process that takes care that the selected machine learning models have specific parameters that are best suited to solve specific real-world business problems accurately. In other words, it measures how well a model generalizes to new data and make accurate predictions. Practical decision-making will not be possible unless model fits the pre-processed data correctly and make accurate predictions.

- In this study model fitting is done for six machine learning algorithms namely Logistic Regress, Support Vector Machine Classifier, Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier and Extreme Gradient Boosting Classifier and deep learning algorithm ,CNN and Bi-LSTM model.

Hyperparameter tuning

- Hyperparameter tuning also known as hyperparameter optimization is a process by which the best set of hyperparameters are selected for a model for its best performance. These parameters are not learned from the training data and are fixed before the beginning of training process, which includes learning rate and number of epochs, number of hidden layers in a neural network and in case of decision tree it will be maximum depth of a tree, criterion.

- In this study hyperparameter tuning is done by applying Randomized search method using `RandomizedSearchCV` of `sklearn.model_selection`. In this method from the available hyperparameters search space random selection is done to choose hyperparameter set. This is repeated as many times as required along with training and evaluating the model until chosen hyperparameter set shows best performance of the model **(DEI, M. (2019)).**

- Hyperparameters chosen for the best performance of the models implemented in this study include

    - penalty that is l1 or l2 regularisation for Logistic Regression.
    - Minimum sample leaves = 20, maximum depth = 20 and criterion = entropy were chosen as best parameters for Decision Tree Classifier.
    - C, sigma and kernel for Support Vector Machine Classifier.
    - Bootstrap = True, criterion = entropy, maximum depth = None, maximum features = 1, minimum sample leaves = 2 and n-estimators = 200 were chosen as best parameters for Random Forest Classifier.
    - Learning rate = 0.01 and n-estimators = 500 were chosen as best parameters for Gradient Boosting Classifier.
    - Sub-sample = 1.0, min_child_weight = 1, maximum depth = 4, gamma = 1, colsample_bytree = 0.6 were chosen as best parameters for Extreme Gradient Boosting Classifier.
    - Epochs = 90 and batch_size = 32 were chosen as best parameters for CNN and Bi-LSTM model.

3.5.3 Evaluating the performance of the model

- Model evaluation or evaluating the performance of the model is an important step in machine learning pipeline in order to check whether the model is able to generalize well to similar new data.
- In this study, multiclass classification is performed. Since it is a classification task, accuracy, F1 score, recall and precision are selected as evaluation metrics.

Confusion matrix

- A confusion matrix is visualization of performance of a machine learning model in a tabular format.

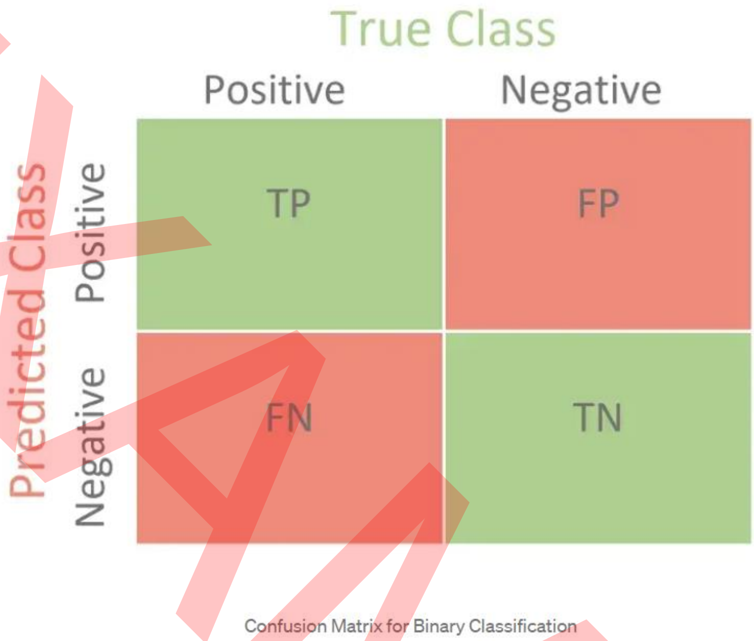


Figure 31 – Confusion matrix for binary classification **(Mohajon, J. (2020))**

- The columns in the confusion matrix represent the actual values of the target variables whereas rows represent the predicted values of the target variables. The four main parameters used for calculating confusion matrix are:
  - True Positive (TP) indicates the number of positive classes correctly predicted by the classifier as positive.
  - True Negative (TN) indicates the number of negative classes correctly predicted by the classifier as negative.
  - False positive (FP) indicates the number of negative classes incorrectly predicted by the classifier as positive.

- False negative (FN) indicates the number of positive classes incorrectly predicted by the classifier as negative.

- The evaluation metrics that can be calculated using confusion matrix are:

Accuracy

- Accuracy is the measure of the number of predictions for both positive and negative cases ,that were classified correctly. Mathematical expression for accuracy is

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$ -----------------------------(5)

Precision

- Precision also known as positive predictive value is the ratio of actually positive predictions to total positive predictions made by the model. Mathematical expression for precision is

$$Precision = \frac{TP}{TP + FP}$$ -----------------------------(6)

- For a good classifier precision should be high. Ideally it should be 1,which indicates FP is 0.

Recall

- Recall also known as true positive rate (TPR) or sensitivity is the ratio of correct positive predictions to all the positive instances in the dataset. Mathematical expression for recall is

$$Recall = \frac{TP}{TP + FN}$$ ----------------------------(7)

- For a good classifier recall should be high. Ideally it should be 1,which indicates FN is 0.

F1 Score

- F1 Score is the harmonic mean of recall and precision. Mathematical expression for F1 Score is

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$ ----------------------------(8)

- For a good classifier F1 Score should be high. F1 Score of 1 indicates best performance whereas F1 Score of 0 indicates the poor performance of the model. It is mostly used for evaluation of classification tasks particularly when the dataset is imbalanced **(B, H.N. (2020)).**

- In this study, multi-class classification is performed. The columns of confusion matrix for multi-class classification indicates original class distribution while that of rows indicate the predicted distribution of the classifying model. The confusion matrix for multi-class classification obtained for CNN and Bi-LSTM model using mlxtend library of Python is shown in figure 32.

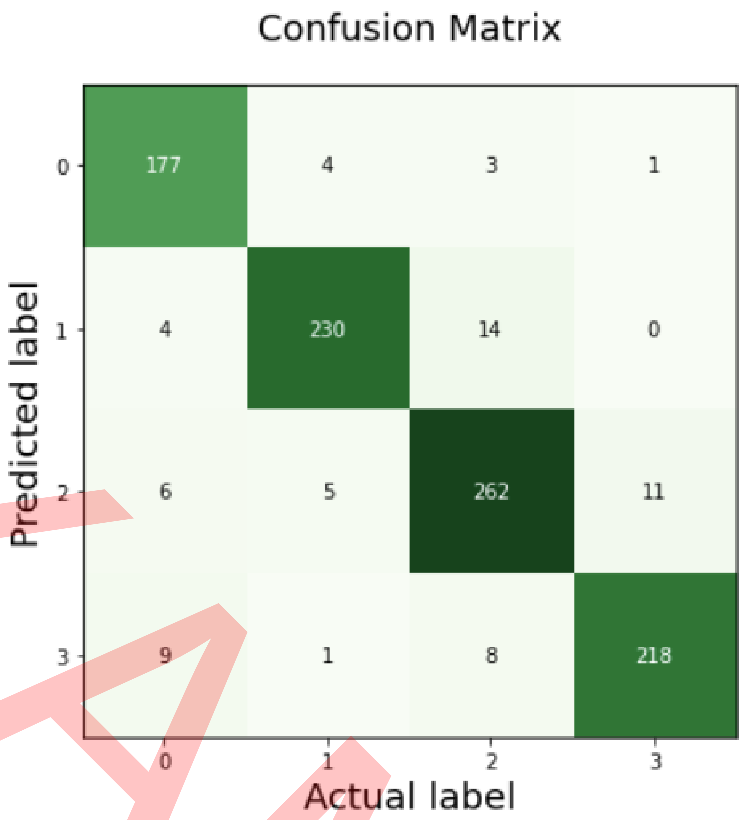


Figure 32 – Confusion Matrix for Multi-Class Classification

3.5.4 Deployment

- Deployment stage involves development and implementation of a web app to predict different types of arrhythmias which is user friendly. Graphical user interface is developed using streamlit, an open-source Python library ,especially intended for data science and machine learning. Streamlit framework is user-friendly because of its ease for use in deploying machine learning models using Python code .

- The Web application named Arrhythmias Prediction System is developed using Python code and run it using Python library, streamlit. Using this app user can either select a .csv file with ECG data and upload it or drag and drop a file by clicking once on Browse files option present on the top right corner.

- Two models, a traditional machine learning model Random Forest Classifier and deep learning model CNN and Bi-LSTM were selected for their best performance when compared to other implemented models for the deployment in predicting the disease. Any of one of these models can be selected using the dropdown to make the prediction.

- Selected model and prediction made is displayed in the web application as shown in figure 33.

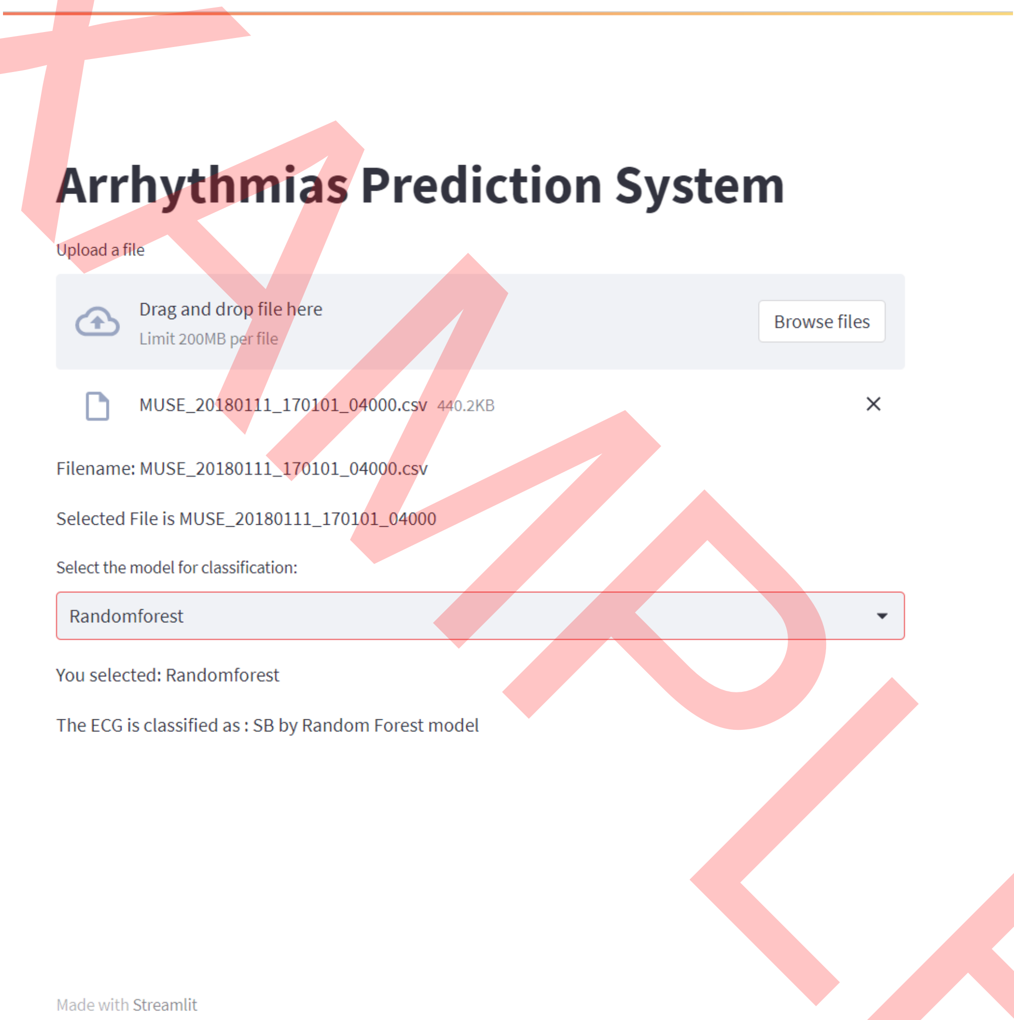


Figure 33 – Web application – Arrhythmias Prediction System

4 RESULTS AND DISCUSSION

This chapter lists the results attained during the implementation of model which includes results of training set and test set and comparison of results obtained for various implemented models and also comparing the results of this study with the benchmark studies.

Results

- Confusion matrix obtained for various implemented algorithms with values for different class labels giving details about performance of models as follows:
- The values that are present diagonally represent the correct predictions of each class respectively.
- The confusion matrix obtained by Logistic Regression is shown in figure 34. Out of 953 records (test set) , correct predictions are obtained for 744 records. Thus, the test accuracy of the model is 78%.

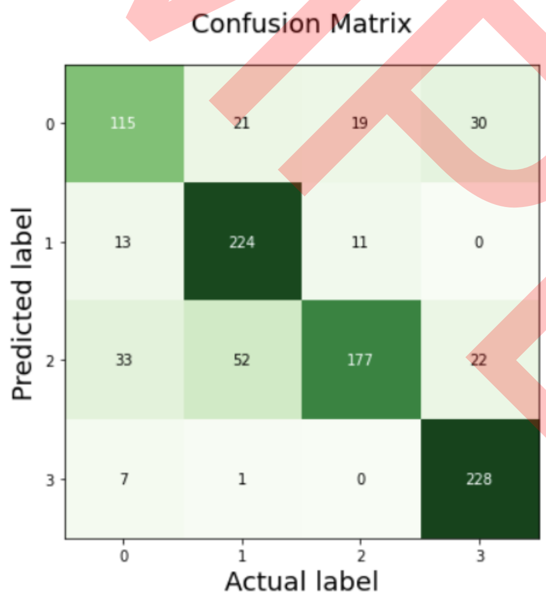


Figure 34 – Confusion matrix by Logistic Regression

- The confusion matrix obtained by Decision Tree Classifier is shown in figure 35. Out of 953 records (test set) , correct predictions are obtained for 878 records. Thus, the test accuracy of the model is 92.1%.

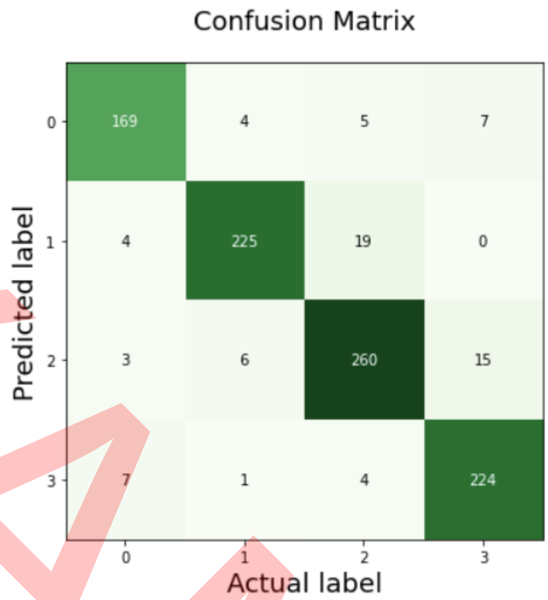


Figure 35 - Confusion matrix by Decision Tree Classifier

- The confusion matrix obtained by Support Vector Machine Classifier is shown in figure 36. Out of 953 records (test set) , correct predictions are obtained for 862 records. Thus, the test accuracy of the model is 90.4%.

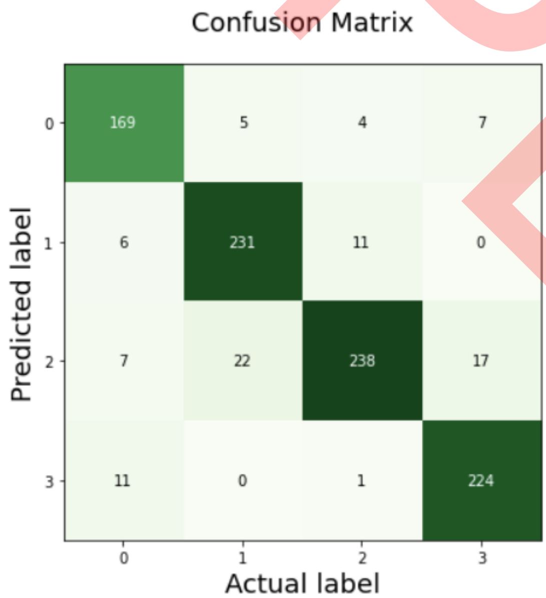


Figure 36 - Confusion matrix by Support Vector Machine Classifier

- The confusion matrix obtained by Random Forest Classifier is shown in figure 37. Out of 953 records (test set) , correct predictions are obtained for 882 records. Thus, the test accuracy of the model is 92.5%.

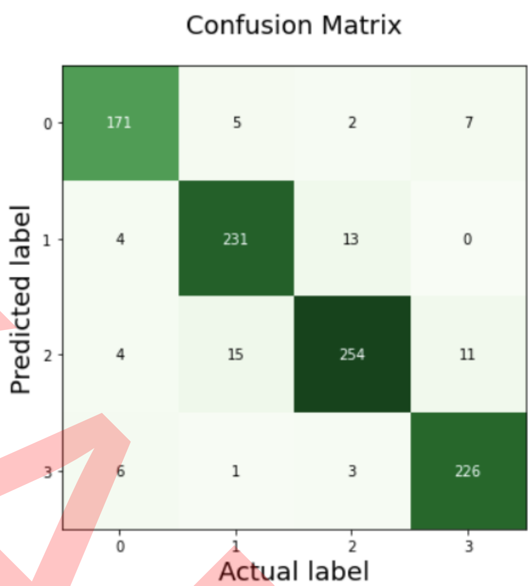


Figure 37 - Confusion matrix by Random Forest Classifier

- The confusion matrix obtained by Gradient Boosting Classifier is shown in figure 38. Out of 953 records (test set) , correct predictions are obtained for 875 records. Thus, the test accuracy of the model is 91.8%.

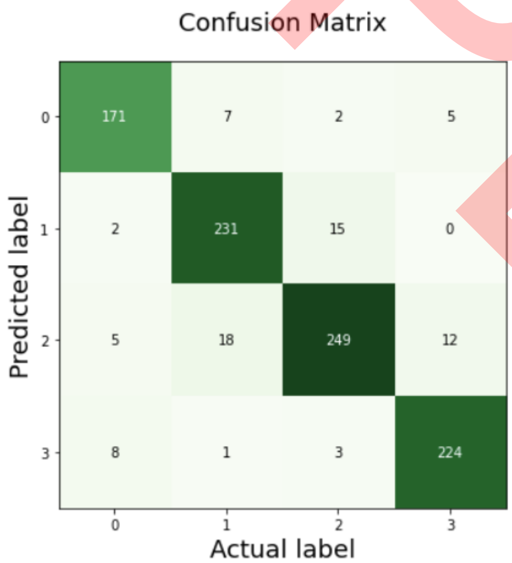


Figure 38 - Confusion matrix by Gradient Boosting Classifier

- The confusion matrix obtained by Extreme Gradient Boosting Classifier is shown in figure 39. Out of 953 records (test set) , correct predictions are obtained for 880 records. Thus, the test accuracy of the model is 92.3%.

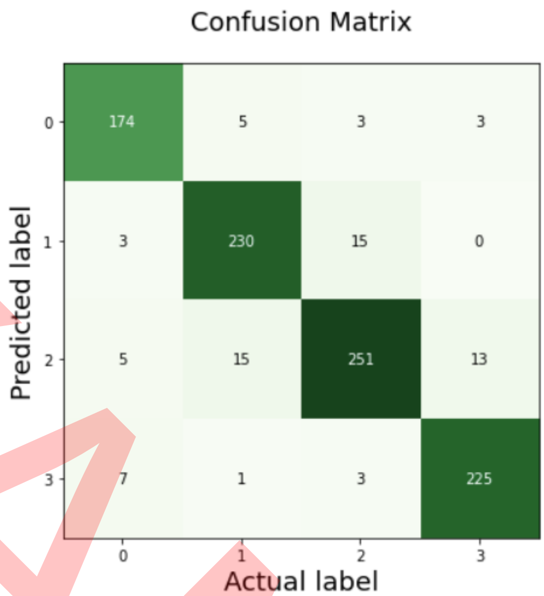


Figure 39 - Confusion matrix by Extreme Gradient Boosting Classifier

- The confusion matrix obtained for deep learning model CNN and Bi-LSTM (hybrid model) is shown in figure 40. Out of 953 records (test set) , correct predictions are obtained for 887 records. Thus, the test accuracy of the model is 93%.

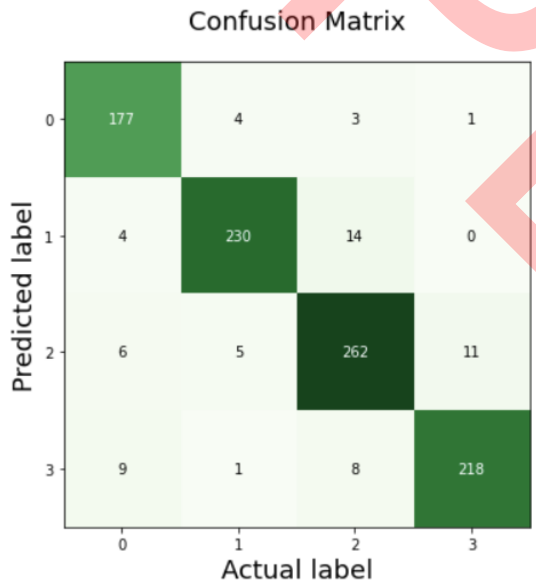


Figure 40 - Confusion matrix by CNN and Bi-LSTM (hybrid model)

Algorithms	Class label	Logistic Regression	Decision Tree Classifier	Support Vector Classifier	Random Forest Classifier	Gradient Boosting Classifier	Extreme Gradient Boosting	CNN and Bi-LSTM
Accuracy	Test	78%	92.1.%	90.4%	92.5%	91.8%	92.3%	93%
	Train	79.8%	93.8%	92.3%	98.3%	96.9%	98%	97.2 %
Precision	0	68%	92%	88%	92%	92%	92%	90%
	1	75%	95%	90%	92%	90%	92%	96%
	2	86%	90%	94%	93%	93%	92%	91%
	3	81%	91%	90%	93%	93%	93%	95%
Precision score		78.4%	92.1%	90.5%	92.5%	91.8%	92.3%	93.1 %
Recall	0	62%	91%	91%	92%	92%	94%	96%
	1	90%	91%	93%	93%	93%	93%	93%
	2	62%	92%	84%	89%	88%	88%	92%
	3	97%	95%	95%	96%	95%	95%	92%
F1 Score	0	65%	92%	89%	92%	92%	93%	93%
	1	82%	93%	91%	92%	91%	92%	94%
	2	72%	91%	88%	91%	90%	90%	92%
	3	88%	93%	93%	94%	94%	94%	94%
F1 Score		77.3%	92.1%	90.4%	92.5%	91.7%	92.3%	93%

Table 2 – Evaluation metrics – accuracy, precision, recall and F1 Score by implemented models

- Evaluation metrics – accuracy, precision, recall and F1 score were calculated from confusion matrix obtained by various implemented models as explained in chapter 3.5.3.
- Accuracy plot is obtained by plotting the value of 'categorical accuracy' for train and test data at the end of each epoch. Whereas Loss plot is obtained by plotting the value of 'categorical cross entropy' for train and

test data at the end of each epoch. Accuracy and Loss plots for Bi-LSTM(hybrid model) are plotted using matplotlib as shown in figure 41 which helps to visualize whether model is good fitting or under fitting or over fitting. Hyper parameter tuning is done by tuning the number of epochs and batch size. 90 epochs and 32 batch size are found to be the best parameters at which model performance is good showing test set accuracy of 93% and train set accuracy of 97.2%.

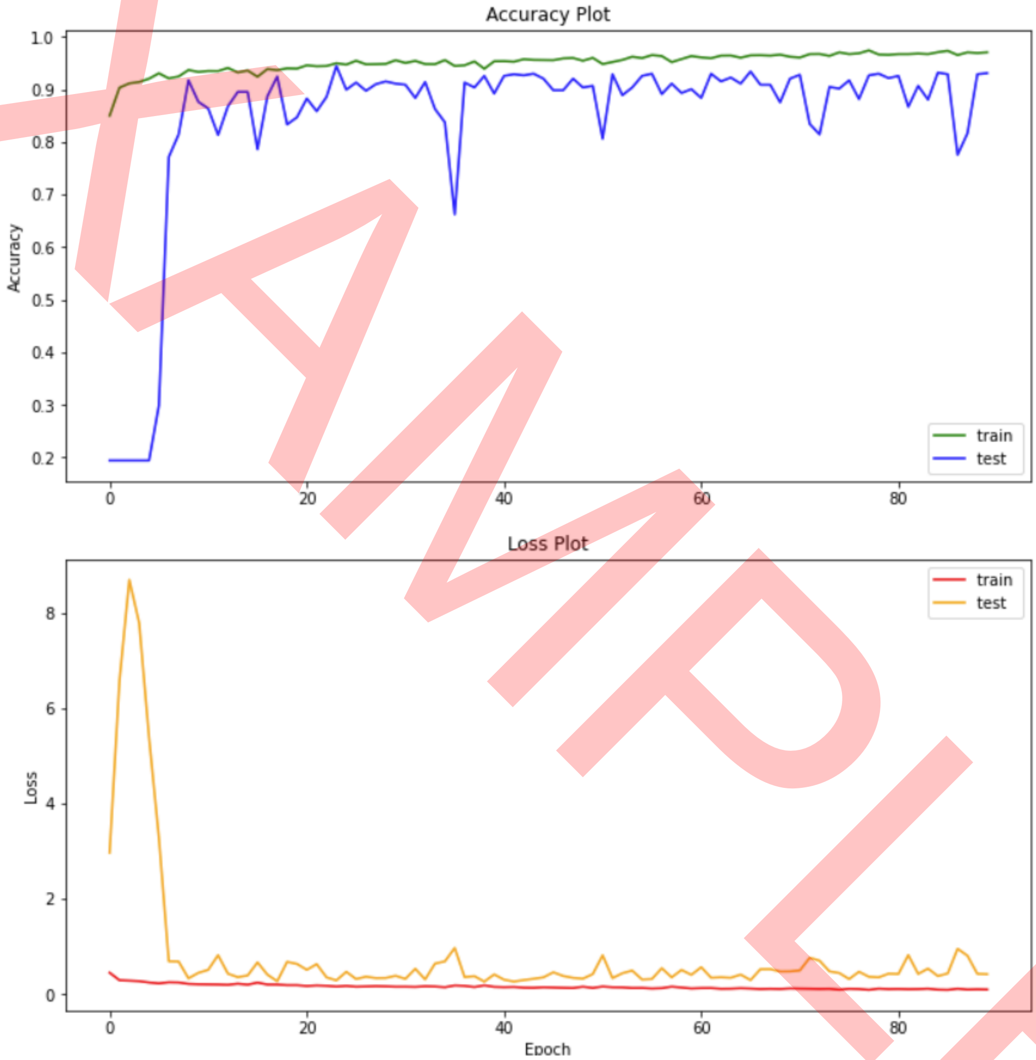


Figure 41 – Accuracy and Loss plot of CNN and Bi-LSTM

- Accuracy of both traditional and deep learning models on the test set is shown if figure 42.

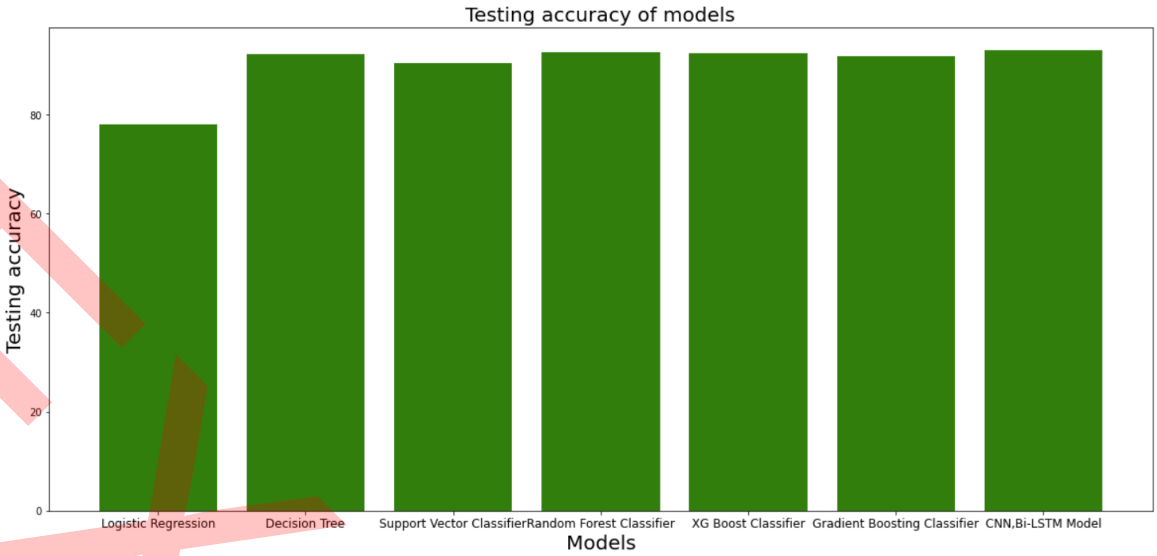


Figure 42 – Bar Chart displaying accuracy of implemented models on test set

- Accuracy of both traditional and deep learning models on the train set is shown if figure 43.

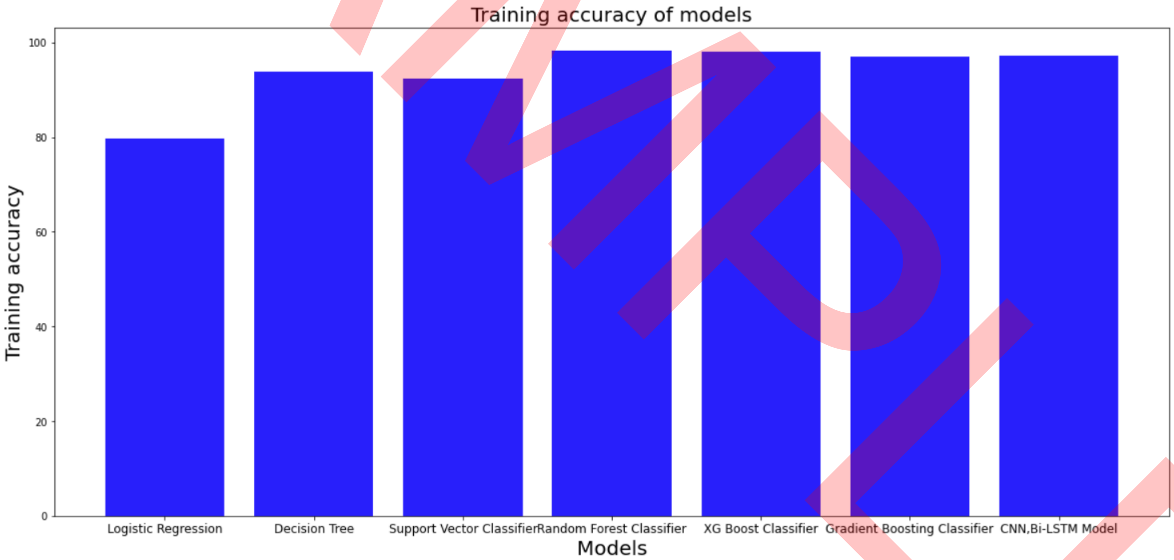


Figure 43 - Bar Chart displaying accuracy of implemented models on train set

- F1 Score of both traditional and deep learning models on the test set is shown if figure 44.

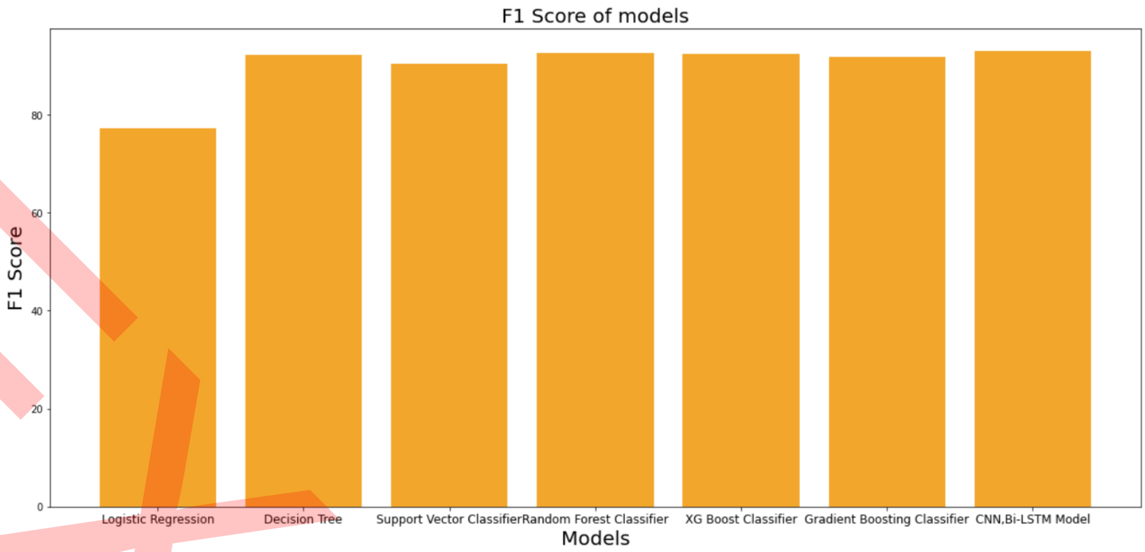


Figure 44 - Bar Chart displaying F1 Score of implemented models test

Discussion

- Evaluation metrics - accuracy, precision, recall and F1 score were calculated from confusion matrix obtained by various implemented models as explained in chapter 3.5.3.

- In this study, six machine learning algorithms namely Logistic Regression, Decision Tree Classifier, Support Vector Machine Classifier, Random Forest Classifier, Gradient Boosting Classifier, Extreme Gradient Boosting Classifier and one deep learning algorithm CNN and Bi-LSTM (hybrid model) were implemented. These models were applied to 31 features selected by feature selection using K best as explained in chapter 3.4.7 and predictions are made to classify different types of Arrhythmias.

- By comparing the performance evaluation from table 2, figure 42 and figure 44 all the implemented algorithms except Logistic Regression had promising results showing accuracy and F1 Score above 90%, achieving the accuracy and F1 score threshold of above 90% chosen for this study.

- Deep learning model CNN and Bi-LSTM and traditional machine learning algorithm Random Forest Classifier were chosen for deployment for making the arrhythmias prediction easy for the end user . CNN and Bi-LSTM is chosen for its best performance when compared to traditional machine learning algorithms showing accuracy and F1 Score of 93%. Random Forest Classifier is chosen for its promising results showing higher accuracy and F1 score of 92.5% when compared to other traditional machine learning algorithms.

- Extreme Gradient Boosting classifier and Decision Tree Classifier are in the next order in the best performance with accuracy and F1 Score of 92.3 and 92.1 respectively. Further train set accuracy for of 93.8 shows that Decision Tree Classifier is not overfitting and can generalise to unseen data. XG Boost is also best in terms of time taken for computation during the prediction when compared to Gradient Boosting Algorithm.

Comparison of Performance of Models against Benchmark Studies

- Model performance of this study are compared with that of the benchmark studies by Cheng, Zou and Zhao, 2021**(Cheng, Zou and Zhao, 2021)** and Chen, Y.-J., Liu, C.-L., Tseng, V.S., Hu, Y.-F. and Chen, S.-A. 2019 **(Chen, Y.-J., Liu, C.-L., Tseng, V.S., Hu, Y.-F. and Chen, S.-A. (2019))** because they carried out similar study on arrhythmias prediction using CNN and Bi-LSTM (hybrid model).

- **(Cheng, Zou and Zhao, 2021)** proposed ECG signal classification by deep CNN (DCNN) and BiLSTM using single-lead data from 2017 Physionet/CINC Challenge ECG dataset to classify ECG signals into normal, AF, other rhythm and noisy. Accuracy of 89.3% and F1 score of 89.1% was obtained.

- **(Chen, Y.-J., Liu, C.-L., Tseng, V.S., Hu, Y.-F. and Chen, S.-A. (2019))** conducted similar study on Arrhythmias prediction using traditional machine learning algorithms Random Forest Classifier, XGBoost and deep learning models, CNN ResNet-34, Bi-LSTM and combining CNN and LSTM to classify large dataset of 12-lead ECG containing 7,704 samples of real-world data collected from clinical institutions was given as input. The proposed model of their study was CNN and Bi-LSTM (hybrid model) and 81% is the average accuracy and 82% is F1 Score achieved for the model. The bar chart showing the experimental results(accuracy and F1 Score) of various models used in their study are shown in Fig 42.

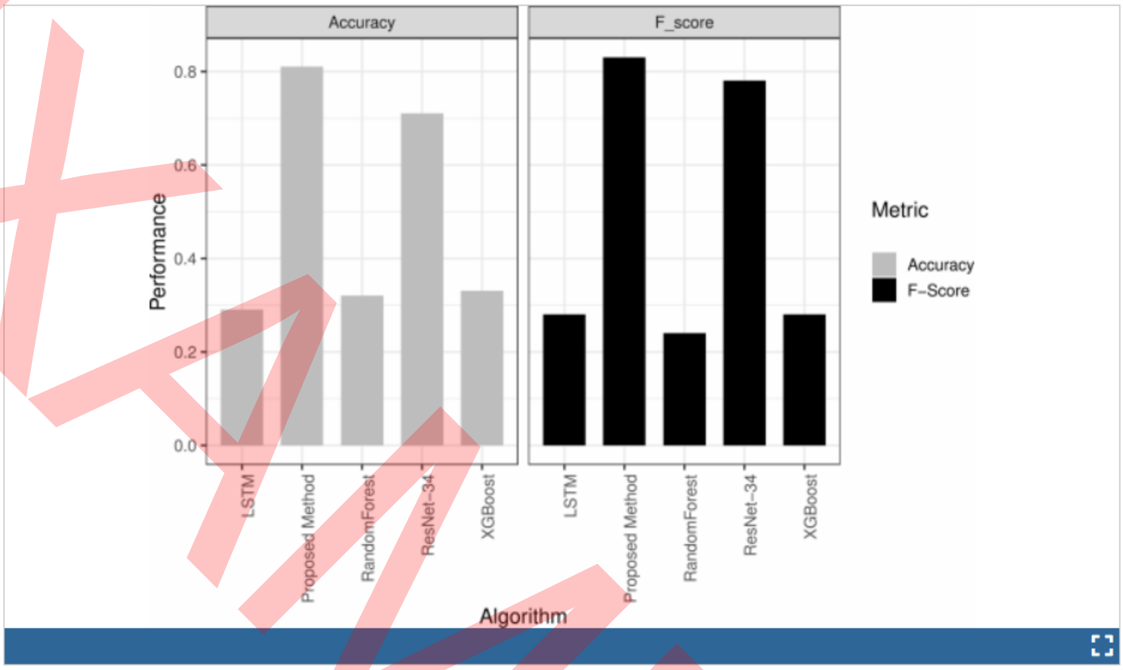


Fig. 2.
Experimental results

Figure 45 - Chen, Y.-J., Liu, C.-L., Tseng, V.S., Hu, Y.-F. and Chen, S.-A. (2019). Experimental results

- By comparison of the results of the two-benchmark studies by Cheng, Zou and Zhao, 2021**(Cheng, Zou and Zhao, 2021)** and Chen, Y.-J., Liu, C.-L., Tseng, V.S., Hu, Y.-F. and Chen, S.-A. 2019 **(Chen, Y.-J., Liu, C.-L., Tseng, V.S., Hu, Y.-F. and Chen, S.-A. (2019))** with the result of this study from table 2, figure 39 and figure 41, an improvement in the accuracy and F1 Score of CNN and Bi-LSTM model(hybrid model) of the proposed study to 93% can be observed. Hyperparameter tuning is the challenging task. It is performed by increasing the number of epochs and changing the batch size, which help in the improvement of model performance showing comparatively best results.

5 LIMITATIONS

- Due to the limited timeline for execution and implementation of the project, there are few limitations to contend with. These limitations comprise the following which are associated with this research:
    - Denoised ECG data is used for analysis due to the limited timeline for this study. The use of raw ECG data might allow to implement denoising steps which can be used to create healthcare apps where raw ECG signals are collected directly from patients.
    - Few outliers are still present in the data. Though outlier treatment is performed in this study, to have sufficient records for analysis only 0.1% of outliers are removed.
    - Considering the time limit for the project, performance of the models is evaluated using only test data. Checking the performance of models with similar new datasets might show how the implemented models generalize to unseen data.

6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

- Deep learning model CNN and Bi-LSTM(hybrid model) and traditional machine learning model Random Forest Classifier were proposed in this study for predicting different types of arrhythmias using a large scale 12-lead ECG dataset from Physionet containing 10,646 records.

- During implementation, the dataset is balanced and for feature extraction peak detection algorithm is used to identify R peaks , RR intervals are noted to calculate the heart rate and statistical calculations of heart rate and RR intervals such as mean RR, standard deviation of RR and min, max, mean and standard deviation of HR(heart rate) are done. Thus 6 features per lead and total of 72 features of all the 12 leads are extracted. These extracted 72 features are concatenated with 11 existing features namely Ventricular rate, Atrial rate, QRS Duration QT Interval, QT Corrected, R Axis, T Axis, QRS Count, Q Onset and Q Offset present in diagnostic file. Thus, obtaining total of 83 features. After data transformation and outlier treatment, based on feature importance 31 features were selected and applied to selected algorithms to predict 4 Rhythm names of target variable.

- Out of all the implemented models, deep learning model CNN and Bi-LSTM shows best performance with accuracy of 93% followed by traditional machine learning algorithm Random Forest Classifier with accuracy of 92.5% as shown in table 2 ,fig 42 and fig 44. The results of this study shows the improved performance of CNN and Bi-LSTM with the benchmark studies.

6.2 Future Work

- Further studies can be done using raw ECG data from the data source of this study or by using real time data. Thus, the technology can be used to create healthcare applications and can be implemented in the real world.
- Further the performance of the models can be tested with new datasets and generalisation of model to new ECG dataset can be checked.

7 REFERENCES

1. Bodini, M., Rivolta, M.W. and Sassi, R. (2020). *Classification of 12-lead ECG With an Ensemble Machine Learning Approach*. [online] IEEE Xplore. doi:10.22489/CinC.2020.406.

2. British Heart Foundation (2017). *Electrocardiogram ECG*. [online] Bhf.org.uk. Available at: https://www.bhf.org.uk/informationsupport/heart-matters-magazine/medical/tests/electrocardiogram-ecg.

3. Chen, Y.-J., Liu, C.-L., Tseng, V.S., Hu, Y.-F. and Chen, S.-A. (2019). *Large-scale Classification of 12-lead ECG with Deep Learning*. [online] IEEE Xplore. doi:10.1109/BHI.2019.8834468.

4. Cheng, J., Zou, Q. and Zhao, Y. (2021). ECG signal classification based on deep CNN and BiLSTM. *BMC Medical Informatics and Decision Making*, 21(1). doi:10.1186/s12911-021-01736-y.

5. D P, Y. and N, D.L. (2021). *Early Detection of Cardiac Arrhythmia Disease using Machine Learning and IoT Technologies*. [online] IEEE Xplore. doi:10.1109/ICOSEC51865.2021.9591884.

6. Han, X., Tu, E. and Yang, J. (2022). *Multimodal 12-lead ECG data classification using multi-label DenseNet for heart disease detection*. [online] IEEE Xplore. doi:10.1109/DSIT55514.2022.9943957.

7. Institute for Quality and Efficiency in Health Care (2019). *What is an electrocardiogram (ECG)?* [online] *www.ncbi.nlm.nih.gov*. Institute for Quality and Efficiency in Health Care (IQWiG). Available at: https://www.ncbi.nlm.nih.gov/books/NBK536878/.

8. Manju, B.R. and Nair, A.R. (2019). *Classification of Cardiac Arrhythmia of 12 Lead ECG Using Combination of SMOTEENN, XGBoost and Machine Learning Algorithms*. [online] IEEE Xplore. doi:10.1109/ISED48680.2019.9096244.

9. Melgarejo-Meseguer, F.M., Gimeno-Blanes, F.J., Rojo-Álvarez, J.L., Salar-Alcaraz, M., Gimeno-Blanes, J.R. and García-Alberola, A. (2018). *Cardiac Fibrosis Detection Applying Machine Learning Techniques to Standard 12-Lead ECG*. [online] IEEE Xplore. doi:10.22489/CinC.2018.174.

10. Moores, D. (2018). *Electrocardiogram*. [online] Healthline. Available at: https://www.healthline.com/health/electrocardiogram.

11. National Health Service (2021). *Electrocardiogram (ECG)*. [online] NHS. Available at: https://www.nhs.uk/conditions/electrocardiogram/.

12. World Health Organization (2021). *Cardiovascular Diseases (CVDs)*. [online] who.int. Available at: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

13. Zheng, J., Zhang, J., Danioko, S., Yao, H., Guo, H. and Rakovski, C. (2020). A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. *Scientific Data*, [online] 7(1), p.48. doi:10.1038/s41597-020-0386-x.

14. Zihlmann, M., Perekrestenko, D. and Tschannen, M. (2017). Convolutional Recurrent Neural Networks for Electrocardiogram Classification. *2017 Computing in Cardiology Conference (CinC)*, DOI: 10.22489/CinC.2017.070-060(2325-887X). doi:10.22489/cinc.2017.070-060.

15. Zuo, W.M., Lu, W.G., Wang, K.Q. and Zhang, H. (2008). *Diagnosis of cardiac arrhythmia using kernel difference weighted KNN classifier*. [online] IEEE Xplore. doi:10.1109/CIC.2008.4749025.

16. Zhang, J., Liang, D., Liu, A., Gao, M., Chen, X., Zhang, X. and Chen, X. (2021). MLBF-Net: A Multi-Lead-Branch Fusion Network for Multi-Class Arrhythmia Classification Using 12-Lead ECG. *IEEE Journal of Translational*

Engineering in Health and Medicine, 9, pp.1–11. doi:10.1109/jtehm.2021.3064675.

17. Yildirim, O., Baloglu, U.B., Tan, R.-S., Ciaccio, E.J. and Acharya, U.R. (2019). A new approach for arrhythmia classification using deep coded features and LSTM networks. Computer Methods and Programs in Biomedicine, 176, pp.121–133. doi:10.1016/j.cmpb.2019.05.004. A new approach for arrhythmia classification using deep coded features and LSTM networks - ScienceDirect

18. Barišić, M. and Jović, A. (2022). Cardiac Arrhythmia Classification from 12-lead Electrocardiogram Using a Combination of Deep Learning Approaches. [online] IEEE Xplore. doi:10.23919/MIPRO55190.2022.9803539.

19. Jain, D. (2019). Data Preprocessing in Data Mining - GeeksforGeeks. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/.

20. Chaudhary, K. (2020). How to deal with Imbalanced data in classification? [online] Drops of AI. Available at: https://dropsofai.com/how-to-deal-with-imbalanced-data-in-classification/ [Accessed 4 Jan. 2023].

21. Cai, W. and Hu, D. (2020). QRS Complex Detection Using Novel Deep Learning Neural Networks. IEEE Access. [online] Available at: https://www.semanticscholar.org/paper/QRS-Complex-Detection-Using-Novel-Deep-Learning-Cai-Hu/52acd580768b37362b6c10aef7e3e699c1aba0a5 [Accessed 18 Jan. 2023].

22. Dr Lewis Potter (2011). Understanding an ECG. [online] Geeky Medics. Available at: https://geekymedics.com/understanding-an-ecg/.

23. Tetelepta, S. (2019). Exploring Heart Rate Variability using Python. [online] Orikami blog. Available at: https://medium.com/orikami-blog/exploring-heart-rate-variability-using-python-483a7037c64d [Accessed 5 Jan. 2023].

24. MAINA, S., 2022. 8 Seaborn Plots for Univariate Exploratory Data Analysis (EDA) in Python . Available from: https://towardsdatascience.com/8-seaborn-plots-for-univariate-exploratory-data-analysis-eda-in-python-9d280b6fe67f

25. Turney, S. (2022). Pearson Correlation Coefficient (r) | Guide & Examples. [online] Scribbr. Available at: https://www.scribbr.com/statistics/pearson-correlation-coefficient/.

26. Stephen Allwright. (2022). Remove outliers from Pandas DataFrame (Updated 2022). [online] Available at: https://stephenallwright.com/remove-outliers-pandas/

27. Mazzanti, S. (2022). 6 Types of 'Feature Importance' Any Data Scientist Should Know. [online] Medium. Available at: https://towardsdatascience.com/6-types-of-feature-importance-any-data-scientist-should-master-1bfd566f21c9.

28. freeCodeCamp.org. (2021). Machine Learning Tutorial – Feature Engineering and Feature Selection For Beginners. [online] Available at: https://www.freecodecamp.org/news/feature-engineering-and-feature-selection-for-beginners/

29. Cheng, J., Zou, Q. and Zhao, Y. (2021). ECG signal classification based on deep CNN and BiLSTM. BMC Medical Informatics and Decision Making, 21(1). doi:10.1186/s12911-021-01736-y.

30. Brownlee, J. (2020). Multinomial Logistic Regression With Python. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/multinomial-logistic-regression-with-python/.

31. Parashar, P. (2020). Logistic Regression and it's Mathematical Implementation. [online] Analytics Vidhya. Available at:

https://medium.com/analytics-vidhya/logistic-regression-and-its-mathematical-implementation-722434ed01a5

32. Brownlee, J. (2016). Logistic Regression for Machine Learning. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/logistic-regression-for-machine-learning/.

33. Manav (2020). Decision Trees: Explained in Simple Steps. [online] Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/decision-trees-explained-in-simple-steps-39ee1a6b00a2.

34. Team, W. (2022). Decision Tree. WallStreetMojo. [online] 27 Dec. Available at: https://www.wallstreetmojo.com/decision-tree/

35. IBM (n.d.). What is a Decision Tree | IBM. [online] www.ibm.com. Available at: https://www.ibm.com/topics/decision-trees.

36. scikit learn (2018). 1.4. Support Vector Machines — scikit-learn 0.20.3 documentation. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/svm.html

37. Gandhi, R. (2018). Support Vector Machine — Introduction to Machine Learning Algorithms. [online] Towards Data Science. Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

38. GeeksforGeeks. (2020). Introduction to Support Vector Machines (SVM). [online] Available at: https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/

39. Polamuri, S. (2017). How the random forest algorithm works in machine learning. [online] Dataaspirant. Available at: https://dataaspirant.com/random-forest-algorithm-machine-learing/#:~:text=The%20random%20forest%20algorithm%20is%20a%20supervised%20classification

40. Koehrsen, W. (2020). Random Forest Simple Explanation. [online] Medium. Available at: https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d

41. Saini, A. (2021). Gradient Boosting Algorithm: A Complete Guide for Beginners. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/

42. Vishal Morde (2019). XGBoost Algorithm: Long May She Reign! [online] Medium. Available at: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

43. Team, G.L. (2020). Understanding XGBoost Algorithm | What is XGBoost Algorithm? [online] GreatLearning Blog: Free Resources what Matters to shape your Career! Available at: https://www.mygreatlearning.com/blog/xgboost-algorithm/

44. Kiranyaz, S., Ince, T. and Gabbouj, M. (2016). Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks. IEEE Transactions on Biomedical Engineering, 63(3), pp.664-675. doi:10.1109/tbme.2015.2468589

45. Rahul, J. and Sharma, L.D. (2022). Automatic cardiac arrhythmia classification based on hybrid 1-D CNN and Bi-LSTM model. Biocybernetics and Biomedical Engineering, 42(1), pp.312-324. doi:10.1016/j.bbe.2022.02.006.

46. Jason Brownlee (2019). How Do Convolutional Layers Work in Deep Learning Neural Networks? [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/

47. Gurucharan, M. (2020). Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network. [online] upGrad blog. Available at: https://www.upgrad.com/blog/basic-cnn-architecture/

48. Vinod, R. (2020). Batch Normalisation Explained. [online] Medium. Available at: https://towardsdatascience.com/batch-normalisation-explained-5f4bd9de5feb

49. Olah, C. (2015). Understanding LSTM Networks -- colah's blog. [online] Github.io. Available at: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

50. Sinha, N. (2018). Understanding LSTM and its quick implementation in keras for sentiment analysis. [online] Medium. Available at: https://towardsdatascience.com/understanding-lstm-and-its-quick-implementation-in-keras-for-sentiment-analysis-af410fd85b47.

51. DEI, M. (2019). Hyperparameter Tuning Explained — Tuning Phases, Tuning Methods, Bayesian Optim, and Sample Code! [online] Medium. Available at: https://towardsdatascience.com/hyperparameter-tuning-explained-d0ebb2ba1d35.

52. Mohajon, J. (2020). Confusion Matrix for Your Multi-Class Machine Learning Model. [online] Medium. Available at: https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826.

53. B, H.N. (2020). Confusion Matrix, Accuracy, Precision, Recall, F1 Score. [online] Medium. Available at: https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd.

8 APPENDICES

8.1 Appendix A:Ethics Approval

Ethical clearance for research and innovation projects

Project status

Status

Approved

Actions

Date	Who	Action	Comments
13:48:00 18 January 2023		Supervisor approved	
11:11:00 16 November 2022		Principal investigator submitted	

Get Help

Ethics release checklist (ERC)

Project details

Project name: Classification of 12 lead ECG to Identify Different Types of Arrhythmias Using Machine Learning and Deep Learning

Principal investigator:

Faculty: Faculty of Business, Law and Digital Technologies

Level: Postgraduate

Course: MSc Applied AI and Data Science

Unit code: COM726

Supervisor name:

Other investigators:

Checklist

Question	Yes	No
Q1. Will the project involve human participants other than the investigator(s)?	○	◉
Q1a. Will the project involve **vulnerable participants** such as children, young people, disabled people, the elderly, people with declared mental health issues, prisoners, people in health or social care settings, addicts, or those with learning difficulties or cognitive impairment either contacted directly or via a **gatekeeper** (for example a professional who runs an organisation through which participants are accessed; a service provider; a care-giver; a relative or a guardian)?	○	◉
Q1b. Will the project involve the use of **control groups** or the **use of deception**?	○	◉
Q1c. Will the project involve any **risk to the participants' health** (e.g. intrusive intervention such as the administration of drugs or other substances, or vigorous physical exercise), or involve psychological stress, anxiety, humiliation, physical pain or discomfort to the investigator(s) and/or the participants?	○	◉
Q1d. Will the project involve **financial inducement** offered to participants other than reasonable expenses and compensation for time?	○	◉
Q1e. Will the project be carried out by individuals unconnected with the University but who wish to use staff and/or students of the University as participants?	○	◉
Q2. Will the project involve sensitive materials or topics that might be considered offensive, distressing, politically or socially sensitive, deeply personal or in breach of the law (for example criminal activities, sexual behaviour, ethnic status, personal appearance, experience of violence, addiction, religion, or financial circumstances)?	○	◉
Q3. Will the project have detrimental impact on the environment, habitat or species?	○	◉
Q4. Will the project involve living animal subjects?	○	◉
Q5. Will the project involve the development for export of 'controlled' goods regulated by the Export Control Organisation (ECO)? (This specifically means military goods, so called dual-use goods (which are civilian goods but with a potential military use or application), products used for torture and repression, radioactive sources.) Further information from the Export Control Organisation	○	◉
Q6. Does your research involve: the storage of records on a computer, electronic transmissions, or visits to websites, which are associated with terrorist or extreme groups or other security sensitive material? Further information from the Information Commissioners Office	○	◉

Declarations

I/we, the investigator(s), confirm that:

- [x] The information contained in this checklist is correct.

- [x] I/we have assessed the ethical considerations in relation to the project in line with the University Ethics Policy.

- [x] I/we understand that the ethical considerations of the project will need to be re-assessed if there are any changes to it.

- [x] I/we will endeavor to preserve the reputation of the University and protect the health and safety of all those involved when conducting this research/enterprise project.

- [x] If personal data is to be collected as part of my project, I confirm that my project and I, as Principal Investigator, will adhere to the General Data Protection Regulation (GDPR) and the Data Protection Act 2018. I also confirm that I will seek advice on the DPA, as necessary, by referring to the Information Commissioner's Office further guidance on DPA and/or by contacting information.rights@solent.ac.uk. By Personal data, I understand any data that I will collect as part of my project that can identify an individual, whether in personal or family life, business or profession.

- [x] I/we have read the prevent agenda.

8.2 Appendix B : GitHub repository