# ECG CLASSIFICATION

TEAM MEMBERS:
Adepu Nithinvas - 20bcs017
Miryala Tarun Kumar- 20bcs132

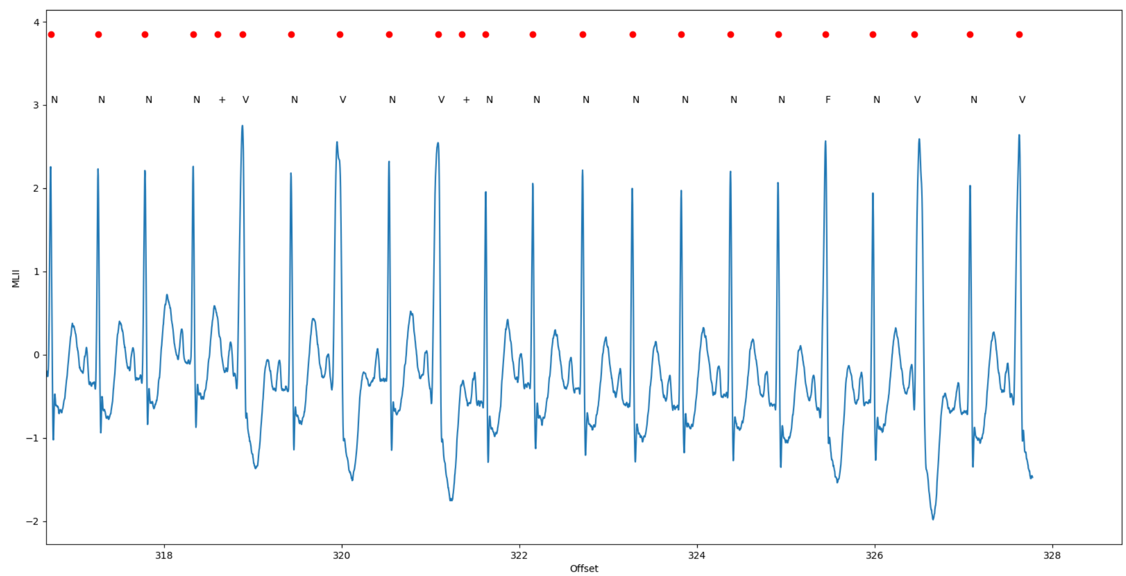
## ABSTRACT:

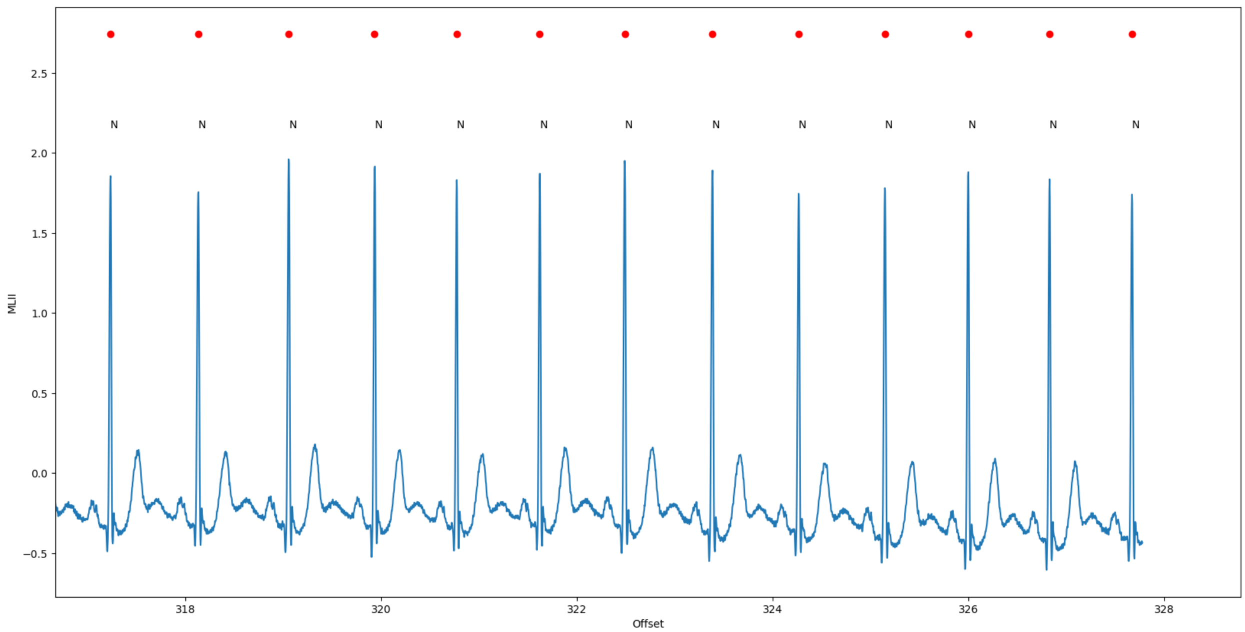
Electrocardiogram (ECG) is a widely used diagnostic tool for detecting cardiac abnormalities. Automatic classification of ECG signals is crucial for timely detection and treatment of cardiac diseases. The aim of this study is to classify different types of arrhythmias, such as Premature Ventricular Contraction (PVC), Atrial Premature Contraction (APC), and other types of beats. This project presents a machine learning-based approach for the classification of ECG signals from the MIT-BIH arrhythmia database using the Pan-Tompkins algorithm for QRS detection and feature extraction. A random forest classifier is employed for the classification of ECG signals based on the extracted features. The proposed approach achieved an accuracy of 97% and weighted average of 97% on the MIT-BIH arrhythmia database, demonstrating its potential for accurate and efficient ECG classification.

## ECG Dataset

The MIT-BIH database, an ECG database provided by the Massachusetts Institute of Technology and based on international standards and annotated information by multiple experts (Moody and Mark, 2001) is used in this study. The MIT-BIH database has been frequently used by the academic community in research for the detection and classification of arrhythmic heartbeats. The MIT-BIH database contains 48 ECG recordings, each recording time is 30 min, the sampling frequency is 360 Hz, and each ECG record is composed of two leads.

Representation of ECG signals of patients with records- 213 and 103 respectively






# Feature Extraction:

Feature extraction is done using the Pan-Tompkins algorithm, which is a widely used algorithm for QRS detection in ECG signals. The algorithm involves several stages of processing to extract relevant features from the ECG signal.

First, the ECG signal is filtered to remove any noise and baseline wander using a combination of high-pass, low-pass, and notch filters. Then, the QRS complex is detected using a combination of differential and integration operations. The algorithm also includes adaptive thresholding to distinguish between QRS complexes and other parts of the ECG signal.

Once the QRS complexes have been detected, several features are extracted, including the R peak amplitude, the R-R interval, and the QRS duration. These features are then used to train the random forest classifier to classify the ECG signals into different arrhythmia categories.

# Data Preprocessing steps:

1. For each ECG record, get the ECG values and generate the classifications of each heart beat based on the annotations.
2. Split the ECG values into individual heart beats, and append the classification of each beat to the beat data.
3. Normalize and resample the beat data, pad with zeroes, and save the beat data along with its classification to a CSV file.

# Labels in the data:

```
# 1.0 Normal
# 2.0 Supraventricular premature beat
# 3.0 Premature ventricular contraction
# 4.0 Fusion of ventricular and normal beat
# 5.0 Unclassifiable beat
```

| Label | Number of instances |
|---|---|
| 0.0 | 112447 |
| 4.0 | 31304 |
| 2.0 | 4783 |
| 3.0 | 820 |
| 1.0 | 4 |

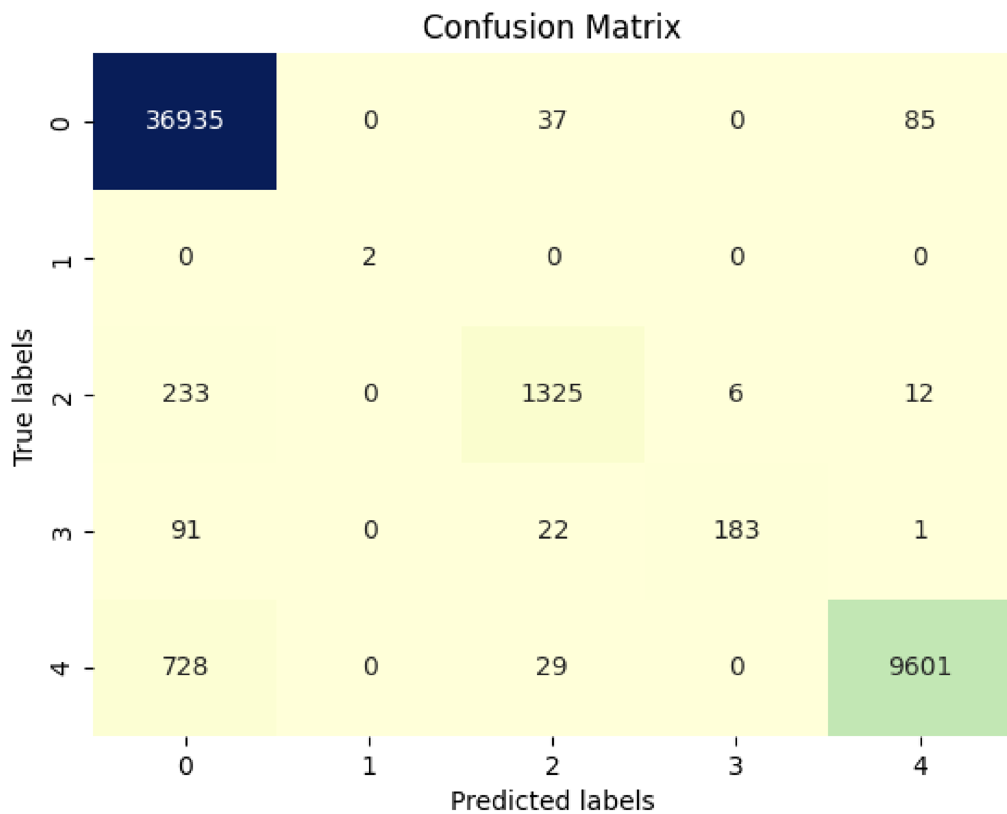
# Machine learning model:

Advantages of using a Random Forest for classification is its ability to handle high-dimensional feature spaces and to capture complex interactions between features.
In the case of the MIT-BIH Arrhythmia Dataset, the input data includes multiple features extracted from ECG signals, such as heart rate, QRS duration, and ST deviation. The dataset also contains multiple classes of arrhythmias, including atrial fibrillation, ventricular tachycardia, and others.

Given the high-dimensional nature of the feature space and the complex relationships between features that is an  indication of different arrhythmias, a Random Forest may be well-suited to handle this task. The algorithm can be trained on a subset of the dataset, and can then be used to classify new ECG signals into the appropriate arrhythmia class based on their extracted features. Overall, the Random Forest algorithm can provide accurate and robust classification of the MIT-BIH Arrhythmia Dataset.

# Results:

Evaluation Metrics used: PRECISION, RECALL, F1_SCORE



# Classification report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.97 | 1.00 | 0.98 | 37148 |
| 1.0 | 0.00 | 0.00 | 0.00 | 2 |
| 2.0 | 0.92 | 0.82 | 0.87 | 1525 |
| 3.0 | 0.96 | 0.65 | 0.77 | 249 |
| 4.0 | 0.99 | 0.93 | 0.96 | 10366 |
| accuracy | | | 0.97 | 49290 |
| macro avg | 0.77 | 0.68 | 0.72 | 49290 |
| weighted avg | 0.97 | 0.97 | 0.97 | 49290 |

# 3-fold cross validation results:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.82 | 0.97 | 0.89 | 37483 |
| 1.0 | 1.00 | 1.00 | 1.00 | 3 |
| 2.0 | 0.86 | 0.27 | 0.41 | 1595 |
| 3.0 | 0.97 | 0.51 | 0.67 | 273 |
| 4.0 | 0.75 | 0.33 | 0.46 | 10434 |
| accuracy | | | 0.81 | 49788 |
| macro avg | 0.88 | 0.62 | 0.68 | 49788 |
| weighted avg | 0.81 | 0.81 | 0.78 | 49788 |

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.81 | 0.94 | 0.87 | 37482 |
| 1.0 | 1.00 | 1.00 | 1.00 | 2 |
| 2.0 | 0.40 | 0.46 | 0.43 | 1594 |
| 3.0 | 0.41 | 0.65 | 0.50 | 274 |
| 4.0 | 0.64 | 0.22 | 0.33 | 10435 |
| accuracy | | | 0.77 | 49787 |
| macro avg | 0.65 | 0.65 | 0.63 | 49787 |
| weighted avg | 0.76 | 0.77 | 0.74 | 49787 |

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.82 | 0.96 | 0.89 | 37482 |
| 1.0 | 1.00 | 1.00 | 1.00 | 3 |
| 2.0 | 0.40 | 0.54 | 0.46 | 1594 |
| 3.0 | 0.56 | 0.28 | 0.37 | 273 |
| 4.0 | 0.71 | 0.24 | 0.36 | 10435 |
| accuracy | | | 0.80 | 49787 |
| macro avg | 0.70 | 0.61 | 0.62 | 49787 |
| weighted avg | 0.78 | 0.80 | 0.76 | 49787 |

# Code Explanation:

This code loads a set of ECG records and processes each record by extracting ECG values and annotations, classifying heartbeats as normal or abnormal based on annotations, splitting the ECG into individual heartbeats, and saving them to a CSV file. The code uses the wfdb library for reading ECG files, the biosppy library for processing ECG data, and the numpy and scipy libraries for data manipulation and resampling.
The code reads all .dat files in the current working directory and for each file, it extracts the ECG data and annotations using wfdb. It then processes each ECG channel separately, finds the R-peaks in the ECG signal using the biosppy library, splits the ECG signal into individual heartbeats based on the R-peaks, and classifies each heartbeat as normal or based on the annotation symbols. The code skips the first and last heartbeats and any heartbeats that do not have a classification. Each heartbeat is then normalized, resampled to 125Hz, and saved to a CSV file with the corresponding classification.

# REFERENCES:

1.https://www.frontiersin.org/articles/10.3389/fncom.2020.564015/full
2.Classification of ECG Arrhythmia using Recurrent Neural Networks Shraddha Singha, Saroj Kumar Pandeyb,∗ , Urja Pawarc , Rekh Ram Janghel.