ECG (Normal & LBBB)

**Team Number: 21**

**Team Members**

|  |  |
| --- | --- |
| Name | ID |
| محمد حسين رجب فرغلى | **2022170581** |
| مازن محمود مصطفى احمد | **2022170339** |
| مازن محمد عبد العال محمد | **2022170337** |
| يوسف محمود ابراهيم ابراهيم | **2022170528** |

# 1. Introduction

* **Objective**: This project aims to diagnose the Left Bundle Branch Block (LBBB) in ECG signals using a K-nearest neighbours (KNN) model.
* **Dataset**: The dataset includes normal and LBBB ECG signals, sourced from the MIT-BIH database with a sampling rate of 360 Hz.

# 2. Data Preprocessing

* **Steps**:
  + **Mean Removal**: The mean of the signal is removed to center it around zero.
  + **Bandpass Filter**: To remove noise, a Butterworth filter is applied with a range of 0.5 to 40 Hz.
  + **Normalization**: The signal is normalized to a standard range [-1, 1] for consistent feature extraction.

# 3. Feature Extraction

* **Wavelet Transform**: Daubechies wavelets are used to decompose the ECG signals into approximation coefficients.
  + **Used Parameters (After Several Trials)**:
    - **Type:** db2
    - **Level:** 3
* **Statistical Features**:
  + **Mean**: The average value of the wavelet coefficients.
  + **Standard Deviation**: The dispersion or variability of the coefficients.
  + **Skewness**: The asymmetry of the coefficient distribution.
  + **Kurtosis**: The peakedness of the coefficient distribution.

# 4. Model Training

* **Algorithms**:
  + **K-Nearest Neighbors (KNN) classifier:**
    - **n\_neighbors**: 11 (Number of neighbours to use)
  + **Support Vector Machine:**
    - **C**: 0.4 (Regularization parameter)
    - **kernel**: 'rbf' (Specifies the kernel type to be used in the algorithm)
  + **Random Forest:**
    - **n\_estimators**: 1 (Number of trees in the forest)
    - **random\_state**: 42 (Controls the randomness of the estimator)
    - **max\_depth**: 1 (The maximum depth of the tree)
* **Parameter Tuning**:
  + **GridSearchCV**: Used to find the best parameters (e.g., number of neighbours, weights) through cross-validation.
  + **Elbow Method**: Plotted error rate vs. Parameters to identify the optimal ones.

# 5. Model Evaluation

* **Metrics**:
  + **Accuracy**: The percentage of correctly classified instances.
  + **Confusion Matrix**: A table showing the true positives, false positives, true negatives, and false negatives.
  + **Permutation Importance**: Evaluated the importance of each feature.

# 6. Trials

## A graph with a blue line Description automatically generatedA graph with a line Description automatically generatedUsed Wavelet (db2, level 3):

# 7. Results

|  |  |  |
| --- | --- | --- |
| Model | Train Accuracy | Test Accuracy |
| KNN | 98.75% | 98.15% |
| SVM | 99.25% | 90.91% |
| Random Forest | 99.50% | 87.21% |

## A graph with a line graph and numbers Description automatically generatedA graph with text on it Description automatically generatedKNN Model

## SVM Model

## A graph with a line and a line Description automatically generatedRandom Forest Model

# 7. Model Deployment

* **Saving the Model**: Using joblib to save and load the trained model for deployment.
* **Creating GUI**: Using Tkinter to develop a simple graphical user interface (GUI) for users to input ECG signals and get predictions.