# Prototype One Report: ECG Arrhythmia Classification

## Methodology and Results

## 1. Executive Summary

This report presents a project focused on developing a deep learning-based signal classification model to identify cardiac arrhythmias using electrocardiogram (ECG) data from the MIT-BIH Arrhythmia Database. The methodology involved signal preprocessing and a Recurrent Neural Network (RNN) architecture incorporating Long Short-Term Memory (LSTM) layers to capture temporal dependencies in ECG signals. The primary objective was to achieve high accuracy in arrhythmia detection for automated cardiac monitoring. The final model achieved an overall test accuracy of **97%**.

## 2. Materials and Methods

### 2.1 Data Source and Tools

* **Dataset**:  
  + Training/Validation: mitbih\_train.csv
  + Testing: mitbih\_test.csv
  + Each sample contains 187 ECG features and 1 class label.
* **Tools & Libraries**:  
  + Python, Pandas, NumPy, Matplotlib, Seaborn, Plotly Express
  + imblearn (RandomOverSampler for balancing)
  + tensorflow.keras (model development)
  + sklearn.metrics (performance evaluation)

### 2.2 Data Characteristics and Exploratory Data Analysis (EDA)

* **Training Data**: 87,554 samples with 188 columns
* **Test Data**: 21,892 samples with 188 columns
* No missing values in either set.

#### Class Distribution:

* Class 0 (Normal): 82.9%
* Class 1 (Atrial Premature Beat - APB)
* Class 2 (Premature Ventricular Contraction - PVC)
* Class 3 (Fusion of Paced and Normal Beats - Fusion Paced)
* Class 4 (Fusion of Ventricular and Normal Beats - Fusion Ventricular)

Severe class imbalance observed, with a heavy skew toward the 'Normal' class.

### 2.3 Data Preprocessing

* **Feature-Target Separation**:  
   ECG signal features were separated from the class labels.
* **Class Imbalance Handling**:  
   Random oversampling was applied, resulting in **72,471 samples per class**.
* **Label Encoding**:  
   One-hot encoding was used for the target labels.
* **Reshaping**:  
   Input reshaped to (samples, 187 time steps, 1 channel) for the CNN-LSTM model.
* **Training-Validation Split**:  
   80% for training, 20% for validation.

### 2.4 Model Architecture

A **Sequential** model was built using Keras with the following layers:

* **Input Layer**:  
   Input(shape=(187, 1))
* **Convolutional Blocks (x2)**:  
  + Conv1D
  + BatchNormalization
  + MaxPooling1D
* **Recurrent Block**:  
  + Two LSTM layers
* **Fully Connected Layers**:  
  + Flatten
  + Dense (intermediate)
  + Dense (output): Dense(5, activation='softmax')
* **Compilation**:  
  + Optimizer: adam
  + Loss: categorical\_crossentropy

### 2.5 Model Training

* **Training Configuration**:  
  + Epochs: 10 (planned)
  + Batch size: 32
* **Callbacks**:  
  + EarlyStopping
  + ReduceLROnPlateau
  + ModelCheckpoint (saved at D:/model\_checkpoint.keras)

Note: Full training logs were unavailable during documentation.

## 3. Results and Discussion

### 3.1 Model Performance

* **Overall Accuracy**: 97% on test set

#### Classification Report:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| 0 | 1.00 | 0.97 | 0.98 | 18,118 |
| 1 | 0.56 | 0.87 | 0.68 | 556 |
| 2 | 0.93 | 0.96 | 0.94 | 1,448 |
| 3 | 0.60 | 0.85 | 0.71 | 162 |
| 4 | 0.95 | 0.99 | 0.97 | 1,608 |
| **Weighted Avg** | **0.97** | **0.97** | **0.97** | 21,892 |
| **Macro Avg** | **0.81** | **0.93** | **0.86** | 21,892 |

#### Interpretation:

* Excellent classification performance for Class 0 (Normal), Class 2 (PVC), and Class 4 (Fusion Ventricular).
* Lower precision for Classes 1 (APB) and 3 (Fusion Paced) indicates some misclassifications.
* High recall across all classes, which is important for medical applications.
* Weighted average F1-score of 0.97 reflects robust overall model performance.

### 3.2 Discussion

The combination of CNN and LSTM layers, supported by RandomOverSampler, proved highly effective for arrhythmia classification. While majority classes were well-learned, performance on minority classes suggests potential for improvement through:

* Advanced oversampling (e.g., SMOTE, ADASYN)
* Cost-sensitive loss functions
* Ensemble approaches for rare class emphasis

## 4. Conclusion and Future Work

This project successfully implemented an end-to-end pipeline for ECG arrhythmia classification using a CNN-LSTM deep learning model. The model achieved **97% accuracy** on the test dataset, demonstrating strong generalization capability.

### Future Directions:

* Enhance minority class classification through advanced imbalance handling.
* Test on additional ECG datasets to validate generalizability.
* Explore model interpretability techniques (e.g., Grad-CAM, saliency maps).
* Deploy as a clinical decision support system prototype.