**ECG Classification Project Report: PTB Diagnostic ECG Database**

**Introduction**

Electrocardiograms (ECGs) are essential for diagnosing heart conditions, but manual analysis is time-consuming and requires expertise. Automated classification with deep learning can improve efficiency, especially for datasets like the PTB Diagnostic ECG Database. Our goal was to distinguish Normal ECGs from Defective ones (including conditions like myocardial infarction, cardiomyopathy, and dysrhythmia). The dataset’s imbalance, with far fewer Normal samples, made this a challenging yet rewarding endeavor, pushing us to adapt and innovate.

Methodology

**Data Acquisition and Preprocessing**

The PTB Diagnostic ECG Database, with 549 records (80 Normal, 469 Defective), was used. Each ECG signal was processed as follows:

Extraction: The first lead was truncated to 5000 samples (5 seconds at 1000 Hz).

**Preprocessing:**

Baseline Wander Removal: Median filtering (kernel size 501) removed low-frequency noise.

**Bandpass Filtering:**

A 4th-order Butterworth filter (0.5–40 Hz) eliminated high-frequency noise and artifacts.

**Normalization:** Signals were standardized to zero mean and unit variance.

**Diagnosis Mapping:** Diagnoses were mapped to "Normal" or "Defective" based on record comments using a predefined dictionary.

The dataset was split into training (80%) and test (20%) sets, stratified to maintain class distribution:

**Training Set:** 375 Defective, 64 Normal

**Test Set**: 94 Defective, 16 Normal

**Model Architecture**

A 1D CNN was designed with:

Feature Extraction:

Three convolutional layers (16, 32, 64 filters) with kernel sizes 15, 11, and 7.

Batch normalization, ReLU activation, max-pooling, and dropout (0.3) to prevent overfitting.

Adaptive average pooling to reduce dimensionality.

Classification:

Two fully connected layers (64→32→1) with ReLU, dropout (0.3), and sigmoid activation for binary classification.

Training

Data Augmentation: Random noise and scaling were applied to training signals to enhance robustness.

Class Imbalance Handling: A WeightedRandomSampler and weighted BCE loss (1.5x weight for Normal class) addressed the imbalance.

Optimization: Adam optimizer (learning rate 0.0005), 22 epochs, with early stopping (patience=3) based on validation loss.

Threshold Optimization: An adaptive threshold (0.4–0.8 range) maximized a weighted score (0.5 Normal recall + 0.3 F1 + 0.2 Defective recall).

Evaluation

The model was evaluated on the test set using:

Metrics: Accuracy, precision, recall, F1-score, and AUC-ROC.

Visualization: Confusion matrix and ECG plots for Normal vs. Defective signals, plus individual predictions.

Results

Model Performance

Training stopped early at epoch 21 due to no improvement in validation loss. The best model (epoch 18) achieved:

Test Accuracy: 87%

Test AUC-ROC: 0.8883

Classification Report:

**Defective (class 0):**

Precision: 0.95

Recall: 0.89

F1-score: 0.92

Support: 94

**Normal (class 1):**

Precision: 0.55

Recall: 0.75

F1-score: 0.63

Support: 16

**Weighted Averages:**

Precision: 0.90

Recall: 0.87

F1-score: 0.88

**The confusion matrix:**

True Positives (Defective): 84

True Negatives (Normal): 12

False Positives: 4

False Negatives: 10

Individual ECG Testing

Normal ECGs: Three randomly selected Normal ECGs were tested:

ECG 1: Misclassified as Defective (probability 0.38).

ECG 2: Correctly classified as Normal (probability 0.54).

ECG 3: Correctly classified as Normal (probability 0.56).

Defective ECG: Correctly classified as Defective (probability 0.07).

Plots for these ECGs and a Normal vs. Defective comparison were saved in the output directory.

Training Dynamics

**Training Loss: Decreased from 2.1973 to 1.6370.**

**Validation Loss: Reached a minimum of 1.2368 at epoch 18.**

**Challenges**

The dataset’s imbalance (80 Normal vs. 469 Defective) was a major hurdle:

Bias Toward Defective Class: The model initially leaned toward Defective predictions due to the majority class.

Limited Normal Samples: Fewer Normal ECGs limited generalization, reflected in the lower Normal precision (0.55).

Threshold Sensitivity: Balancing recall for both classes required careful threshold tuning.

Solutions Implemented

Weighted Sampling and Loss: Prioritized Normal samples to mitigate imbalance.

Data Augmentation: Noise and scaling improved robustness for the Normal class.

Early Stopping and Threshold Optimization: Prevented overfitting and optimized classification boundaries.

Strengths

High Defective Class Performance: An F1-score of 0.92 is critical for clinical settings, where missing abnormalities is costly.

Robust Preprocessing: Ensured clean, standardized input signals.

Adaptive Thresholding: Improved Normal class recall (0.75), addressing the imbalance.

Limitations

Normal Class Precision: At 0.55, false positives for Normal ECGs suggest room for improvement.

Small Test Set: Only 16 Normal test samples limited evaluation robustness.

Single Lead: Using only the first lead may miss multi-lead diagnostic patterns.

Visualizations

Confusion Matrix: Highlighted strong Defective classification but some Normal misclassifications.

Normal vs. Defective ECG Plot: Showed distinct signal patterns for interpretability.

Individual ECG Plots: Offered insights into correct and incorrect predictions.

Conclusion

This project delivered a CNN-based ECG classifier with 87% accuracy and excellent Defective class performance (F1-score 0.92). Despite the challenge of an imbalanced dataset, strategic preprocessing, augmentation, and weighted training achieved reasonable Normal class performance (F1-score 0.63). Building this model was very difficult as with the low number of normal ecgs available in my chosen dataset's, it was very difficult to find somewhat of a balance in my model to identify both and though it performs very good in identifying defective data, it can still be improved in identifying normal ecg datas.

AI use:

Used AI in formatting the code and fine tuning the AI model, other sources were online google soruces.

