University of Science and Technology of Hanoi
Information and Communication Technology Department

Machine Learning in Medicine

Practice 1 - Report

ECG Heartbeat Categorization

Student: Do Thi Huong Tra - BA12-174

1 Introduction

This report presents a project for the Machine Learning in Medicine course. In this report, I analyze the MIT-BIH Arrhythmia dataset and propose a model for performing the classification task.I tried to analyze on the data and apply 2 models based on RandomForest and CNN classify 5 types of heartbeats.This lab work is inspired by the findings presented in Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks [1]

2 Dataset

2.1 Overview

The dataset used in this study is sourced from the MIT-BIH Arrhythmia Database and comprises five distinct types of heartbeats—Normal, Supraventricular, Ventricular, Fusion, and Unknown—encoded as 0, 1, 2, 3, and 4 respectively. Each record contains 187 measurements of the heartbeat signal.

2.2 Data Analysis

Figure 1's histogram clearly shows that the Kaggle training dataset is strongly bias by the Normal class, which is the most common heartbeat type. This imbalance may cause the model to overfit specifically to the Normal class.

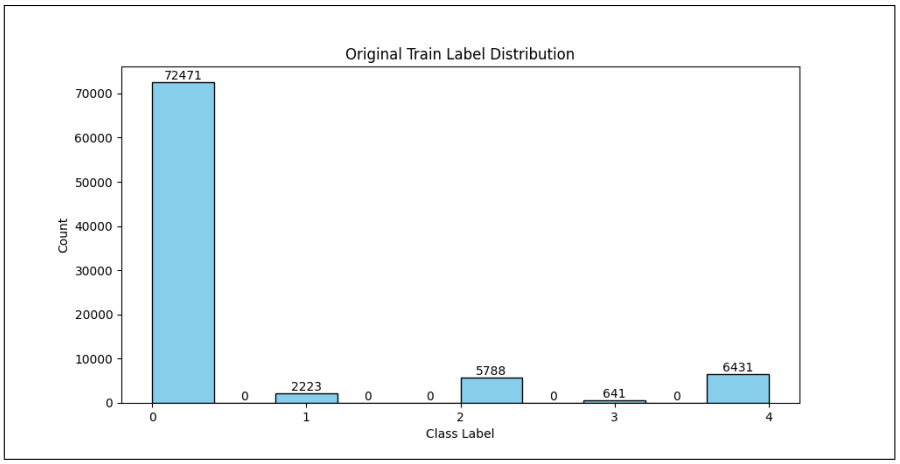

Figure 1: Histogram showing the distribution of heartbeat classes in the training dataset.

To prevent overfitting, I applied an oversampling strategy to balance the dataset. After dividing the total number of training samples by 5, I obtained roughly 17,510 samples per class. Therefore, I resampled each class to achieve approximately 17,500 samples per class.

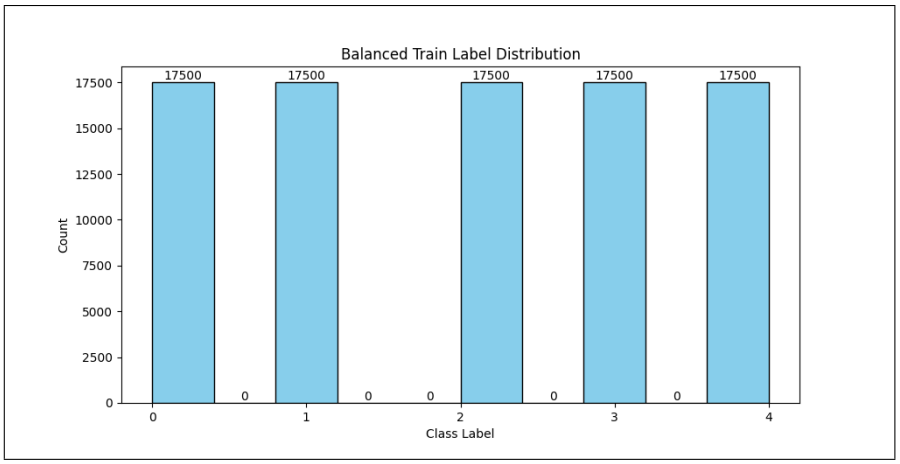

Figure 2: Histogram showing the balanced distribution of heartbeat classes in the training dataset.

3 Model Architecture

3.1 Convolutional Neural Network

I used a simple one-dimensional convolutional neural network is designed for classifying sequential data into five categories. It begins with a 1D convolutional layer that takes a single-channel input and applies 16 filters with a kernel size of 5, thereby extracting local temporal features from the signal. A ReLU activation function introduces nonlinearity right after the convolution, allowing the model to learn complex patterns. The output is then passed through a max pooling layer with a kernel size of 2, which reduces the dimensionality and highlights the most salient features while providing some translational invariance. Finally, the pooled feature maps are flattened into a one-dimensional vector and fed into a fully connected linear layer that produces the final logits corresponding to the five classes.

3.2 Random Forest

I also implemented the Random Forest classifier, as it is a cost-effective choice for structured data classification. I follow to the structured machine learning pipeline, starting with data preprocessing, splitting into training and validation sets, and then training a RandomForestClassifier with 100 decision trees. I set 42 for the random state.

4 Evaluation and Discussion

To evaluate the model's performance, I split the test dataset into two parts: one for validation and one for final testing, maintaining a **50:50** ratio. The evaluation was conducted using three key performance metrics: **F1-score**, **Precision**, and **Recall**.

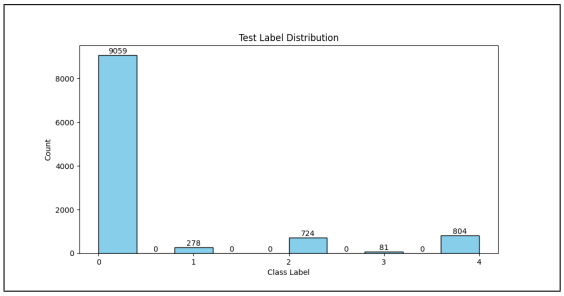

Figure 3: Test set distribution

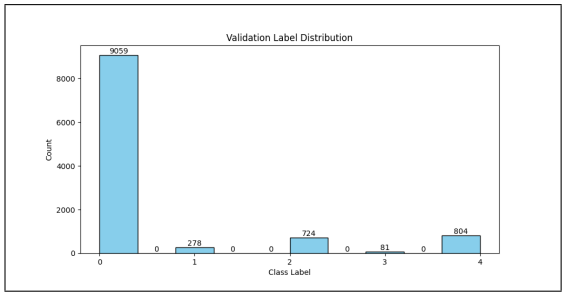

Figure 4: Validation set distribution

Class	Precision	Recall	F1-score	Support
0	0.99	0.87	0.93	9059
1	0.33	0.75	0.46	278
2	0.76	0.90	0.82	724
3	0.12	0.95	0.21	81
4	0.91	0.96	0.94	804
Accuracy			0.88	10946
Macro Avg	0.62	0.89	0.67	10946
Weighted Avg	0.95	0.88	0.90	10946

Table 1: CNN Test Classification Report

Class	Precision	Recall	F1-score	Support
0	0.98	0.99	0.99	9059
1	0.81	0.74	0.78	278
2	0.92	0.93	0.93	724
3	0.76	0.70	0.73	81
4	0.97	0.97	0.97	804
Accuracy			0.97	10946
Macro Avg	0.86	0.87	0.88	10946
Weighted Avg	0.97	0.97	0.97	10946

Table 2: Random Forest Validation Classification Report

The model performs well on majority classes (0, 4) but struggles with minority classes (1, 3). For further work, I can do refining loss functions, and tuning hyperparameters could significantly enhance performance.

References

[1] Pranav Rajpurkar, Awni Y. Hannun, Masoumeh Haghpanahi, Codie Bourn, and Andrew Y. Ng. Cardiologist-level arrhythmia detection with convolutional neural networks, 2017.