Heartbeat Classification: Further Analysis and Applied Deep Learning Model

Dang Quang Bui

February 2025

Abstract

An electrocardiogram (ECG) has been a reliable tool for monitoring the functioning of the cardiovascular system. It records the electrical signals in the heart. Today, many research topics focus on categorizing heartbeats, where advanced technology is used to analyze and classify different types of heartbeats, helping to detect irregularities or abnormalities. By categorizing heartbeats, these tools can assist healthcare professionals in diagnosing various heart conditions more accurately and efficiently.

1 Introduction

In this report, we will focus on the EDA analysis and propose a method based on the alternative RNN-based model, Long Short-Term Memory (LSTM). Additionally, we evaluated the proposed method using the MIT-BIH Arrhythmia Database, which will be discussed in more detail later.

2 Dataset

The MIT-BIH Arrhythmia Database consists of 48 half-hour ECG recordings with two channels, collected from 47 people by the BIH Arrhythmia Laboratory between 1975 and 1979. Twenty-three of these recordings were randomly selected from a set of 4000 24-hour ECG recordings, which were gathered from a mix of inpatients (about 60%) and outpatients (about 40%) at Boston's Beth Israel Hospital. The remaining 25 recordings were chosen to include rarer but clinically important arrhythmias that wouldn't be well-represented in a smaller random sample. The Arrhythmia Dataset contains a total of 109,446 samples, which are categorized into five distinct classes.

The data is provided in a CSV file format, with each sample represented by a 188-feature vector. The five classes in the dataset are as follows: 'N' (Normal), 'S' (Supraventricular ectopic beat), 'V' (Ventricular ectopic beat), 'F' (Fusion beat), and 'Q' (Unclassifiable beat), each representing different types of heartbeats or abnormalities.

Figure 1 presents a line graph that compares the feature values of the first row of the ECG training and test data sets.

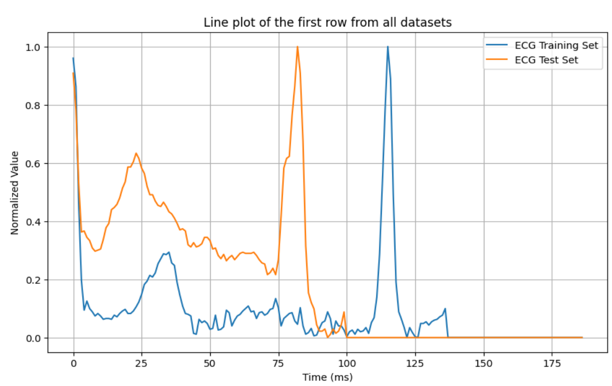

Figure 1: Line plot of the first row from all datasets

Category	Annotations
N	Normal Left/Right bundle branch block Atrial escape Nodal escape
S	Atrial premature Aberrant atrial premature Nodal premature Supra-ventricular premature
V	Premature ventricular contraction Ventricular escape
F	Fusion of ventricular and normal
Q	Paced Fusion of paced and normal Unclassifiable

Table 1: Summary of mappings between beat annotations and AAMI EC57 categories.

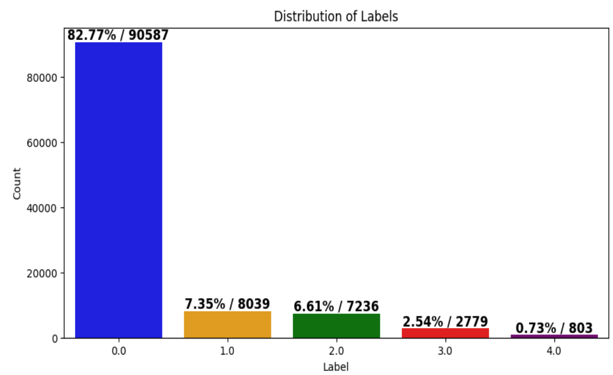

Figure 2: Label Distribution across ECG data set

3 Data Preprocessing

One of the significant challenges associated with the ECG data set is the presence of class imbalance, where certain data categories are underrepresented compared to others. As we observe in Figure 2, the majority of the data belongs to label 0, which accounts for 82.77 % (90,587 samples), making it the most dominant class. In contrast, label 1 takes 7.35% (8,039 samples), and label 2 represents 6.61% (7,236 samples). The remaining 2 minority classes, label 3 and label 4, report 2. 54% (2,779 samples) and 0.73% (803 samples), respectively.

With such an imbalance, machine learning models can negatively impact, as they may become biased toward the majority class while struggling to correctly classify the minority classes.

To address this problem, we generally applied SMOTE, an oversampling technique where synthetic samples are generated for the minority class. However, SMOTE alone may introduce noise by creating synthetic samples too close to the majority class. Therefore, the combination of SMOTE and Tomek Links is introduced in such a hybrid technique that aims to clean overlapping data points for each of the classes distributed in sample space. A Tomek Link is a pair of nearest-neighbor samples from different classes that are very close to each other. These pairs typically exist near decision boundaries, making classification more difficult. By removing these links, often from the majority class, the separation between classes improves, reducing noise and enhancing class distinction.

The SMOTE-Tomek process is described in the following steps:

1. **Generate Synthetic Samples for Minority Classes:** For each sample x_i belonging to the minority class, synthetic samples are generated using linear interpolation:

$$x_{\text{synthetic}} = x_i + \lambda(x_{\text{nn}} - x_i), \tag{1}$$

where x_{nn} is a randomly selected nearest neighbor of x_i from the same minority class, and $\lambda \in [0, 1]$ is a random scalar.

2. **Identify Tomek Links:** A Tomek Link is defined as a pair of samples x_i and x_j that are mutual nearest neighbors, meaning x_i is the closest sample to x_j and vice versa. Also, x_i and x_j belong to different classes, with one from the minority class and the other from the majority class. The equation for identifying Tomek links is given by:

$$d(x_i, x_j) = \min\{d(x_i, x_k) \mid x_k\}, \tag{2}$$

where $d(x_i, x_j)$ is the distance (commonly Euclidean) between x_i and x_j, and x_k belongs to a different class than x_i.

3. **Remove Majority Class Samples:** For each identified Tomek Link, remove the sample belonging to the majority class to enhance class separation and reduce noise.

Figure 4 shows the results of this method. Each class comprises approximately 20% of the total dataset, with nearly equal sample counts of around 72,470 instances per class. Eventually, the resampled dataset now shows a perfectly balanced distribution among all five classes. We will do a similar task in the testing set, as we illustrated in Figure 3.

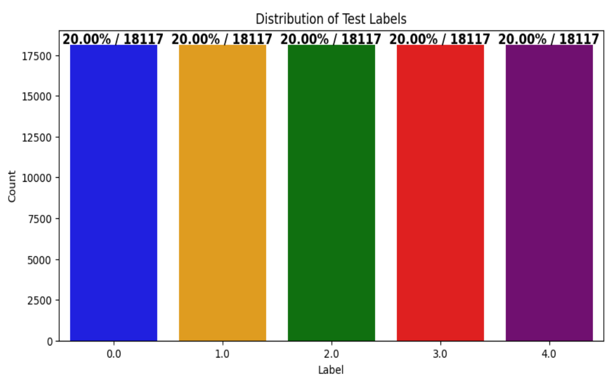

Figure 3: Label Distribution across training data set before and after SMOTE + Tomek Links

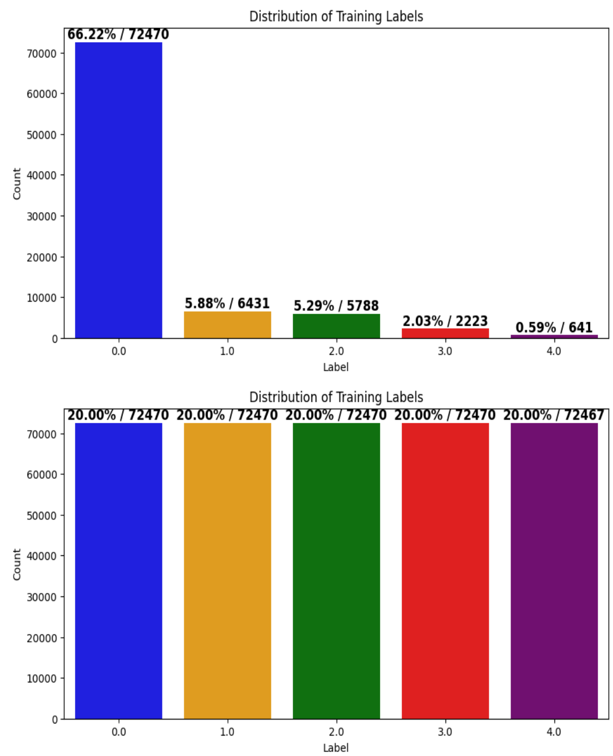

Figure 4: Label Distribution across training data set before and after SMOTE + Tomek Links

4 Methodology

With all the conditions met, our dataset is ready for training, and significant insight is learned through the proposed training model: LSTM.

4.1 LSTM Model Overview

Long Short-Term Memory (LSTM) networks are the type of deep learning model designed to process sequential data while keeping important information over time. Unlike traditional Recurrent Neural Networks (RNNs), LSTMs are built to overcome the vanishing gradient problem, which makes it difficult for standard RNNs to learn long-term dependencies. This makes it highly effective in understanding and predicting patterns in sequential data like time series, text, and speech.

LSTM networks introduce memory cells, which can retain information over long sequences. Each memory cell has three main components: an input gate, a forget gate, and an output gate. These gates help regulate the flow of information in and out of the memory cell.

4.2 Model Architecture

Figure 5 provides a detailed representation of the overall architecture of our model:

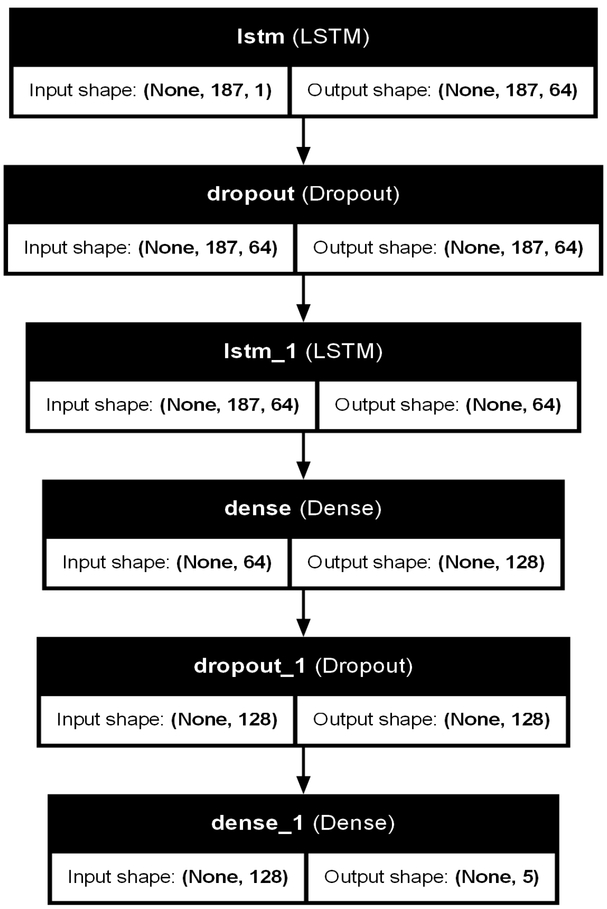

Figure 5: Architecture of the proposed network

This model consists of a sequential arrangement of layers designed for ECG heartbeat classification. The first layer is an LSTM layer with 64 units, which takes input sequences of shape (187,1). A dropout layer with a rate of 0.3 follows to reduce overfitting. Another LSTM layer with 64 units processes the output further before passing it to a dense (fully connected) layer with 128 neurons and ReLU activation for feature extraction. Another dropout layer with a rate of 0.1 is added for regularization. Finally, a dense output layer with 5 neurons (indicated 5 labels outcome) and a softmax activation function classifies the input into five categories. The model is compiled using the Adam optimizer and sparse categorical cross-entropy loss, making it suitable for this classification task.

4.3 Evaluation

We proposed several evaluation benchmarks to conclude the efficiency of the Long Short-Term Memory model we have been trained on. While the model performs the heartbeat classification, we observe some insight by employing four distinct types of metrics - Precision, Recall, F1-score, and Accuracy. Finally, we also assess the impact of running epochs to understand their effect on performance, such as the Loss/Accuracy tendency.

4.3.1 Epochs Evaluation

During the training step, we obtained some significant insight into each observation step. Figure 6 will display the training and validation loss (left) and accuracy (right) over epochs for a model.

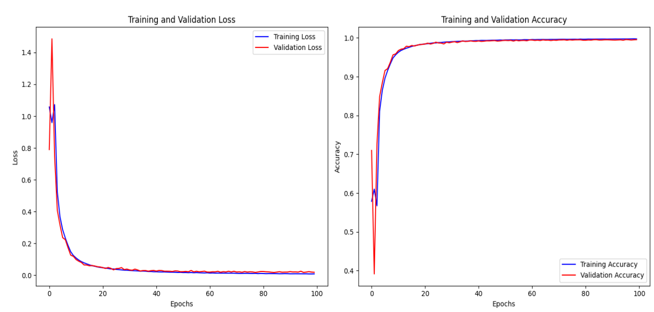

Figure 6: Train and Validation Accuracy/Loss Plots

Observing the loss graph, both training and validation loss consistently decrease, with validation loss dropping more sharply than training loss around the 15 epochs. This suggests the model is learning effectively and not overfitting significantly, as there isn't a sharp divergence between the two curves. One noticeable note is that there are fluctuations in loss, but as training progresses, both curves steadily decline and eventually stabilize near zero. In the accuracy plot, validation accuracy reaches a high level after over 25 epochs and remains stable to over 90 % accuracy at 100 epochs before we apply early stopping, which indicates no further improvement in the model's performance. As we conducted, the model recorded a good performance, indicating a good generalization.

4.3.2 Metrics Evaluation

After training a model, we return its classification report and confusion matrix. The classification report provides a better aspect of the model's performance. It gives information about the model, such as its precision, recall, f1-score, and accuracy.

The confusion matrix library is used to analyze the model's real-time performance. It offers a confusion matrix that counts the total number of correctly predicted labels. It also makes it possible to see the deviation from the prediction in cases where the model makes incorrect predictions. The results of the confusion matrix for the models are shown in Figure 7.

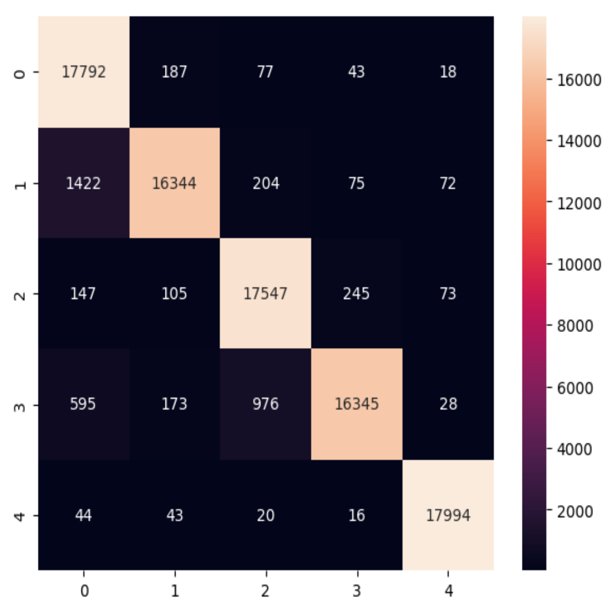

Figure 7: Confusion matrix for the LSTM model

Table 2: Classification Report

Class	Precision	Recall	F1-score	Support
0.0	0.89	0.98	0.93	18117
1.0	0.97	0.90	0.93	18117
2.0	0.93	0.97	0.95	18117
3.0	0.98	0.90	0.94	18117
4.0	0.99	0.99	0.99	18117
Accuracy		0.95		90585
Macro Avg	0.95	0.95	0.95	90585
Weighted Avg	0.95	0.95	0.95	90585

Observing the confusion matrix plot between label 0 and label 1, we can see that the False Positives, where label 1 was incorrectly predicted as label 0 1422 times, are a greater concern than the False Negatives, where label 0 was incorrectly predicted as label 1 only twice. Therefore, Precision is a suitable metric for this situation. Regardless of the measurements for precision from TABLE II, it ranges from 0.89 to 0.99, signifying the model's high confidence in its positive predictions, with slight variation across different classes.

Notably, class 0 has slightly lower precision (0.89) compared to other classes, indicating a higher false positive rate for this category. However, its recall of 0.98 compensates for this, ensuring that most true positives are correctly identified.

From the analysis of both Figure 7 and Table 2., it can be observed that the classification model's overall performance is highly satisfactory, witnessing an accuracy of 95%, which indicates that the majority of the predictions are correct across all classes.

References

[1] Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. Ecg heartbeat classification: A deep transferable representation. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, page 443–444. IEEE, June 2018.

[2] Barak Or. Improving requirements classification with smote-tomek preprocessing, 2025.

[3] Amrita Rana and Kyung Kim. Ecg heartbeat classification using a single layer lstm model. pages 267–268, 10 2019.