

# ECG Heartbeat Classification

Can Trung Hieu

## I. INTRODUCTION

Medical diagnostics and machine learning research both benefit greatly from the use of the Electrocardiogram (ECG) Classification Dataset. The electrocardiogram signals that were collected from patients, spanning a range of physiological states and cardiac circumstances, are included in this collection. Heart health and function can be greatly benefited by examining ECG signals, which are a continuous recording of the heart's electrical activity.

In cardiovascular medicine, prompt diagnosis, prognosis, and therapy planning depend heavily on the accurate classification of ECG signals. Using these algorithms to automate the process of ECG analysis and interpretation has gained a lot of attention with the development of advanced machine learning and deep learning techniques. The effectiveness of these algorithms, however, is strongly dependent on the availability of high-quality datasets covering a wide range of cardiac abnormalities and normal fluctuations.

Accurate ECG signal classification is critical for cardiovascular medicine's timely diagnosis, prognosis, and treatment planning. As deep learning and sophisticated machine learning techniques have developed, there has been a growing interest in using these algorithms to automate the process of ECG analysis and interpretation. However, high-quality datasets covering a wide range of cardiac anomalies and normal variations are essential for the efficacy of these algorithms.

## II. BACKGROUND

A vital diagnostic technique in cardiology, electrocardiography (ECG) records the heart's electrical activity throughout time. Electrodes are applied to the skin of the body to detect electrical changes in the heart muscle that take place during a heartbeat. These alterations are subsequently captured and shown as waves on paper or the ECG monitor. The electrocardiogram (ECG) is a useful tool for diagnosing and treating a variety of cardiac disorders, including arrhythmias, myocardial infarction, and electrolyte imbalances. It offers important information about the heart's rhythm, pace, and conduction system. The electrocardiogram (ECG) is still a mainstay of cardiovascular care because of its non-invasive nature and fast assessment of heart function.

In machine learning and pattern recognition applications, the Multilayer Perceptron (MLP) is a basic kind of artificial neural network (ANN). MLP is made up of several layers of feedforward-organized, networked nodes, or neurons, that are modeled after the architecture of the human brain. After applying a weighted sum to the inputs from the layer before it, each neuron processes the information through an activation function to generate an output. MLPs can reduce the

discrepancy between expected and actual outputs by adjusting the weights of connections during training through a process called backpropagation. This allows MLPs to discover intricate patterns and correlations within the data. Because of their capacity to represent nonlinear relationships, MLPs have found use in a variety of domains, such as financial forecasting, speech recognition, and picture recognition.

Integrating ECG data with MLP-based machine learning techniques has become a viable field in cardiovascular medicine and research. MLPs can be trained to automatically identify and categorize cardiac problems from ECG signals, providing advantages including real-time monitoring, early arrhythmia diagnosis, and customized risk assessment. MLP models help clinicians make timely and well-informed judgments by using the temporal and spectral information found in ECG waveforms to distinguish between normal and abnormal cardiac rhythms with high accuracy. Additionally, the integration of MLP and ECG makes it easier to create intelligent systems for telemedicine, predictive analytics, and remote patient monitoring, all of which improve the efficacy and efficiency of the provision of cardiac care.

## III. METHODOLOGY

### A. Data Preprocessing

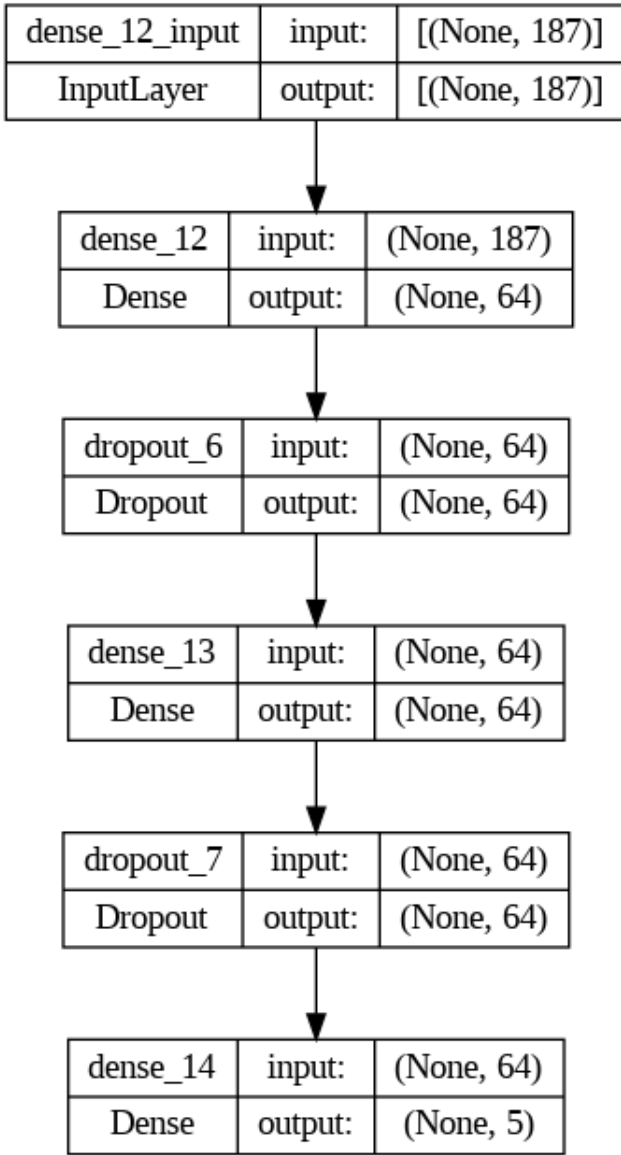
To make sure the input dataset is suitable for MLP training, preparatory steps are first applied. Normalization is a common step in this preprocessing to help with feature scale and distribution problems. In this case, the dataset is standardized using normalization techniques, ensuring that every feature follows a standard distribution with a mean of 0 and a standard deviation of 1.

### B. Model Architecture

The MLP model's architecture is carefully designed to support strong generalization and efficient learning. The model is made up of Dropout layers strewn in between Dense layers, which are densely coupled layers built with the Keras Sequential API. Rectified linear unit (ReLU) activation functions are used by the Dense layers to encourage non-linearity and feature extraction. In the meantime, Dropout layers are positioned carefully to reduce overfitting by deactivating a portion of neurons at random while training. L2 regularization is also used on the Dense layers to manage model complexity and improve the model's capacity to generalize to new inputs.

### C. Model Compilation and Optimization

The whole model is carefully set up for training using the Adam optimizer, an adaptive learning rate optimization technique that is highly regarded for its durability and effectiveness. The core of the Adam optimizer is its ability to dynamically modify learning rates throughout the training



**Fig. 1: MLP Architecture**

phase, enabling quick convergence and ideal weight updates. Through the dynamic adjustment of learning rates, the optimizer effectively navigates the complex topography of the loss function, facilitating the model's efficient convergence toward an optimal solution.

To measure the difference between the predicted and real labels, the model's training routine is based on the sparse categorical cross-entropy loss function. This loss function is designed to measure the difference between predicted probability and ground truth labels in multi-class classification settings where target labels appear as integers. The model is iteratively refined by its prudent use, minimizing the difference between expected and actual results each time.

#### D. Model Training

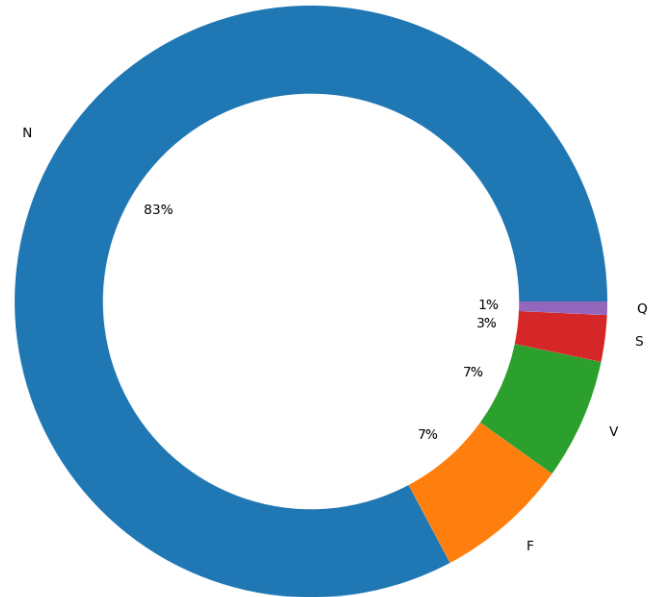
The model is trained thoroughly across 50 epochs, with each epoch consisting of iterating through all 64 samples in

mini-batches from the training dataset. Twenty percent of the training data is set aside for validation during training, which helps to identify overfitting early on. To reduce overfitting, the model architecture consists of three Dense layers with ReLU activation strewn among Dropout layers. Softmax activation is used in the last layer for multi-class categorization. The model optimizes parameters to minimize classification errors by utilizing the Adam optimizer and sparse categorical cross-entropy loss function. Training dynamics are observed through the monitoring of epoch-wise measures, such as accuracy and loss, by a custom callback function.

After being trained on a separate test dataset, the model is systematically evaluated to determine how well it can generalize. Comprehensive test results are documented, including measures for accuracy and loss. By providing an accurate assessment of the model's performance on unobserved data, this structured evaluation process guarantees the model's preparedness for practical deployment. To sum up, strict guidelines are followed during the model's training and assessment processes, ensuring excellent results and resilience in categorization assignments.

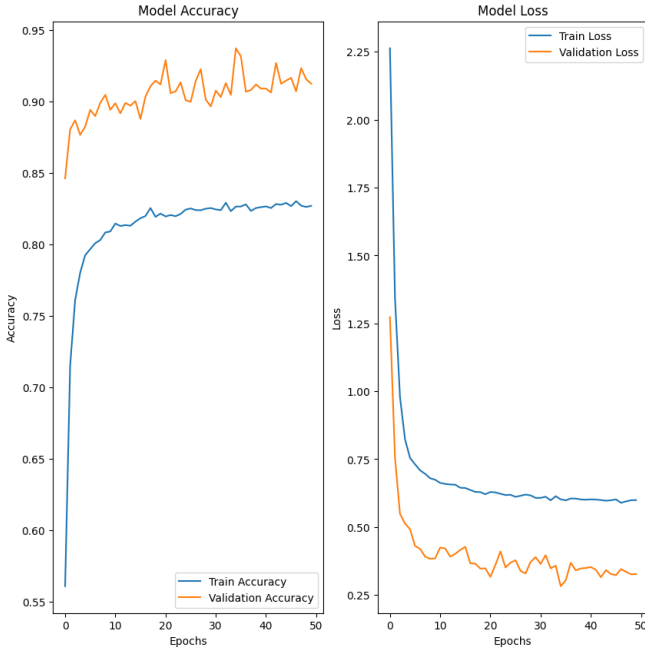
#### IV. EVALUATION

The ECG dataset is imbalanced when the proportions between groups are not uniform and there is a clear difference. An unbalanced dataset also affects the model's predictions and can lead to overfitting



**Fig. 2: Dataset Ratio**

After 50 epochs with batch size 64 and validation with the ratio 8:2, the model's accuracy is 87.2 percent with a loss of 0.4582



**Fig. 3: Model Accuracy and Model Loss**

The model's accuracy curve, depicted in blue, consistently outperforms the model's loss curve, represented in red, across both the training and validation datasets. Throughout the training process, the model's accuracy exhibits a dynamic range between 0.55 and 0.95, with its zenith reaching an impressive 0.95. Meanwhile, the validation accuracy fluctuates between 0.60 and 0.90, peaking at 0.90, reflecting a robust performance across different data subsets.

Delving into the intricacies of model loss, the training loss curve (Train Loss) consistently overshadows its counterpart, the validation loss curve (Validation Loss). The training loss trajectory spans from 0.25 to 2.25, reaching its pinnacle at 2.25, while the validation loss oscillates between 0.50 and 2.00, with a maximum value of 2.00. This nuanced exploration of loss dynamics underscores the model's capacity to adapt and refine its parameters throughout the training process.

After a comprehensive analysis, the model is a strong candidate since it exhibits high accuracy and low loss in the training and validation sets. The model may have grown unduly specialized to the subtleties of the training data, as indicated by the noticeable difference in accuracy between the training and validation runs. Similarly, the need for regularization strategies to support a more generic model architecture is highlighted by the divergence between training and validation losses.

Overall, the model demonstrates high accuracy and low loss for both the training and validation sets. The higher training accuracy compared to validation accuracy suggests potential

overfitting to the training set. Similarly, the higher training loss compared to validation loss indicates potential overfitting.

## V. CONCLUSION

By consistently displaying high accuracy and relatively low loss across the training and validation datasets, the model demonstrates impressive performance characteristics, that is, in conclusion. Criticism, however, points up possible areas for development. The model may have overfitted to the subtleties of the training data, as indicated by the difference in accuracy between training and validation runs. It also highlights the necessity of regularization strategies to improve the model's generalization capabilities, given that the training loss was higher than the validation loss. To encourage a more resilient and flexible model architecture in the future, techniques like dropout or weight regularization that attempt to reduce overfitting may be used.

Moreover, the observation of higher training loss compared to validation loss further emphasizes the need for regularization strategies. The disparity in loss metrics indicates that the model's parameters might have been excessively tailored to the nuances of the training set, compromising its ability to generalize well. Implementing regularization techniques such as dropout or weight regularization can help mitigate overfitting by introducing constraints on the model's parameters during training, thereby encouraging a more robust and adaptable architecture.

## VI. FUTURE WORK

Firstly, exploring alternative model architectures or ensemble methods could provide insights into building more resilient and flexible models. Ensemble methods, in particular, can combine the predictions of multiple models to improve overall performance and reduce the risk of overfitting.

Furthermore, investigating the use of data augmentation techniques during training could also be beneficial. Data augmentation involves artificially increasing the size of the training dataset by applying transformations such as rotation, scaling, or flipping to the existing data samples. This approach can help expose the model to a wider range of variations within the data and reduce the likelihood of overfitting.

Finally, conducting thorough hyperparameter tuning experiments to optimize the model's architecture and training parameters could lead to better performance and generalization capabilities. Techniques such as grid search or random search can be employed to systematically explore the hyperparameter space and identify configurations that yield the best results.

We may improve the model's resilience and versatility by implementing similar tactics in subsequent studies, which will ultimately boost the model's performance on a range of tasks and datasets.