



HIGHER SCHOOL OF COMPUTER SCIENCE

DEEP LEARNING PROJECT REPORT

ECG Signal Classification

Submitted By :

CHERGUI Abdelmadjid

FERRAR Hadjer Aroua

BAHA Meriem

SLIMANI Medjda Rihab

Submitted To :

N.DIF

Contents

1	Introduction	3
2	Related Work	4
3	Methodology	5
3.1	DataSet	5
3.2	Preprocessing	6
3.3	Model	6
3.3.1	Main model	6
3.3.2	CNN - BiLSTM	13
3.3.3	LSTM Autoencoder	16
4	Deployment	23
4.1	Home Page:	23
4.2	Prediction Page	23
4.3	Result Page	24
5	Conclusion	26
6	References	26

List of Figures

1	pie chart of imbalanced dataset	5
2	pie chart of balanced dataset	6
3	main model architecture	8
4	CNN Lstm results	9
5	CNN Lstm Classification report	10
6	CNN Lstm Confusion matrix	10
7	CNN Lstm results	11
8	CNN Lstm Classification report	11
9	CNN Lstm Confusion matrix	12
10	CNN Bilstm architecture	14
11	CNN bi-lstm learning curves	15
12	CNN bi-lstm learning curves	15
13	CNN Lstm Classification report	16
14	Classification report cnn-Bilstm	16
15	LSTM autoencoder architecture	18
16	Lstm Auto-encoder results	19
17	reconstruction error on the training set	20
18	reconstruction error on normal testing set	21
19	reconstruction error on anomalies testing set	22
20	main page	23
21	predict page	24
22	result page	24
23	ECG plot window	25

1 Introduction

Cardiovascular disease (CVD) — often regarded as the ‘silent pandemic’ — is the leading cause of death worldwide. In 2019 alone, it was responsible for 18.6 million deaths according to the World Health Organization.

Efforts have been made to implement programs and policies to reduce their incidence. The electrocardiogram (ECG) has become a widely used biosignal for the detection of CVDs in diverse communities. ECG signals provide a graphical representation of heart electrical activity, which enables the identification of various heart diseases and abnormalities. For over 70 years, doctors have used ECG signals to detect heart diseases, such as arrhythmia and myocardial infarctions. ECG signals consist of several waves, including P, QRS complex, T, and potentially U waves, which can be analyzed to diagnose many cardiac diseases.

Electrocardiogram (ECG) machines provide an affordable and safe method for detecting cardiac diseases by analyzing waveform variations in the P, QRS complex, T, and U waves. However, noise and artifacts caused by factors like body or electrode movements and power line interferences can lead to inaccurate diagnoses. Preprocessing techniques, such as the widely used wavelet transform, are essential for removing these artifacts and ensuring accurate ECG analysis. Despite its effectiveness, manual analysis of ECG signals is a complex and time-consuming task that is prone to errors. Given the significant impact of cardiovascular diseases on global health, researchers are exploring machine learning techniques to improve the accuracy and efficiency of ECG analysis. These studies aim to provide a low-cost and accurate diagnosis of arrhythmic heartbeats, which can be life-threatening for millions of people.

ECG heartbeat classification can be approached using two methods: classical and deep learning. Classical methods include decision trees, support vector machines, Naive Bayes Classification, multimodal feature fusion, multimodal image fusion, wavelet transform, independent component analysis, interval information, discrete cosine transform, and Fisher’s linear discriminant analysis. These algorithms extract features from the ECG signal, such as frequency, temporal, statistical, and more, to classify ECG heartbeats. While these methods are relatively simple, their accuracy can be improved. For instance, the SVM algorithm achieved an F1-score of 0.87.

Deep learning models, on the other hand, use neural networks such as convolutional neural networks and long short-term memory to classify ECG signals. While deep learning models have shown superior performance compared to classical methods, they are prone to over-fitting and require significant computational resources, making them unsuitable for long-term ECG monitoring.

2 Related Work

ECG signal classification is an important task in medical diagnosis, and has been the subject of extensive research over the years. Various approaches have been proposed for this task, including rule-based methods, feature-based methods, and machine learning-based methods. Among these, machine learning-based methods have shown promise in achieving high accuracy and robustness.

Recurrent Neural Networks (RNNs) are a type of neural network that is particularly well-suited for processing sequential data, such as time-series data. RNNs have been successfully applied to a variety of tasks, including speech recognition, natural language processing, and image captioning. Recently, there has been growing interest in using RNNs for ECG signal classification.

One advantage of using RNNs for ECG signal classification is that they can capture temporal dependencies between ECG segments, which is important for distinguishing between different arrhythmias. In addition, RNNs can handle variable-length inputs, which is useful for ECG signals that may have different lengths.

Several studies have demonstrated the effectiveness of RNNs for ECG signal classification. For example, Li et al. (2019) used a bidirectional RNN to classify ECG signals into four categories with an accuracy of 98.17%. Similarly, Acharya et al. (2017) used a deep RNN with attention mechanism to classify ECG signals into five categories with an accuracy of 97.50%.

However, there are also some limitations to using RNNs for ECG signal classification. One challenge is the issue of overfitting, particularly when dealing with small datasets. Another challenge is the difficulty of interpreting the model’s decisions, which is important for clinical applications.

In summary, RNNs have shown promise for ECG signal classification, and have achieved state-of-the-art results on several benchmark datasets. However, there are still some challenges that need to be addressed, particularly in terms of model interpretability and generalization to new datasets.

3 Methodology

3.1 DataSet

In this project, we used a dataset composed of two collections of heartbeat signals derived from two famous datasets in heartbeat classification, the MIT-BIH Arrhythmia Dataset and The PTB Diagnostic ECG Database as data source for labeled ECG records. Furthermore, we demonstrate that the knowledge learned from the former database can be successfully transferred for training inference models for the latter. In all of our experiments, we have used ECG lead II re-sampled to the sampling frequency of 125Hz as the input.

The dataset contains Time Series examples (obtained with ECG) with 187 timesteps. Each sequence corresponds to a single heartbeat from a single patient with congestive heart failure.

We have 5 types of hearbeats (classes):

1. Normal
2. Artial Premature
3. Premature ventricular contraction
4. Fusion of ventricular and normal
5. Fusion of paced and normal

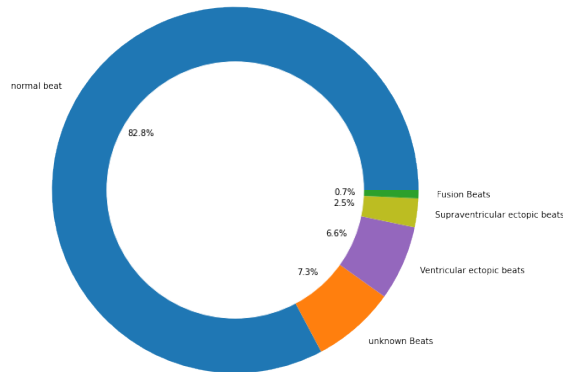


Figure 1: pie chart of imbalanced dataset

3.2 Preprocessing

The dataset is already preprocessed, but since the dataset is greatly imbalanced we tried to balance the dataset by upsampling minority classes (class 1 to 4) and downsampling the majority class (class 0).

The dataset is divided into classes based on the value in the 187th column. The minority classes are upsampled to have 20,000 instances. The majority class is then downsampled to have the same number of samples as the upsampled minority classes. This imbalance can lead to biased model training, where the model tends to predict normal conditions more accurately than those with heart disease, even though it may have a higher impact on patient outcomes.

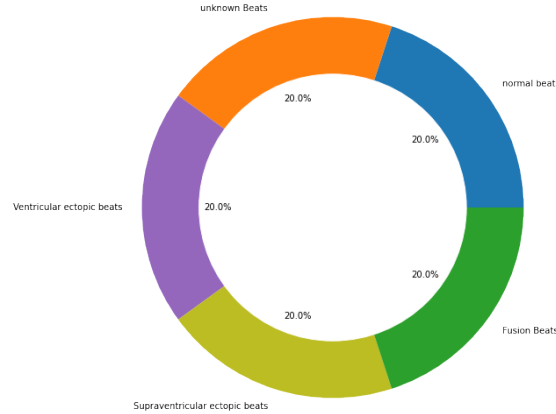


Figure 2: pie chart of balanced dataset

We balanced the dataset by upsampling minority classes and downsampling majority classes. We first create separate dataframes for each class label (0-4), then we oversample each of the minority class dataframes to 20,000 samples (with replacement). Next, it downsamples the majority class (class 0) to 20,000 samples. Finally, we concatenate all the upsampled and downsampled dataframes into one balanced dataframe. Is shown in Figure 2.

3.3 Model

3.3.1 Main model

The implemented model architecture for the ECG signals classification project comprises a sequential neural network.

The model starts with a series of convolutional layers to extract features from the

input data. The first convolutional layer consists of 32 filters with a kernel size of 5 and a stride of 2. This is followed by batch normalization to stabilize the model's training. The subsequent convolutional layer employs 64 filters with similar settings. Another batch normalization layer follows, ensuring the model's stability. The final convolutional layer has 128 filters with a kernel size of 3 and a stride of 2.

After batch normalization, the extracted features are passed through two LSTM layers.

The first LSTM layer has 128 units and returns the output sequence, while the second LSTM layer with the same number of units does not return sequences. The model concludes with a dense layer that produces predictions for the different egg classes using the softmax activation function. This architecture enables the model to learn and classify eggs based on their distinctive features, contributing to accurate classification outcomes.

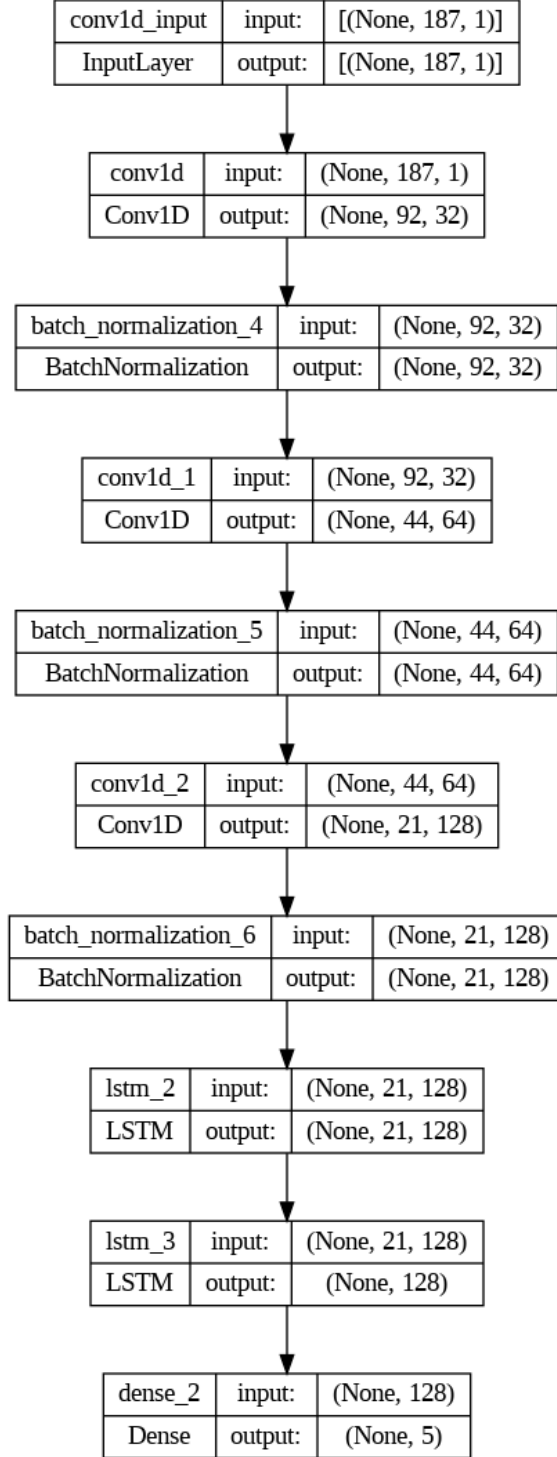


Figure 3: main model architecture

The chosen optimizer for training this model is Stochastic Gradient Descent (SGD), which updates the model's weights based on the gradients of the loss function. The learning rate is set to 0.01, determining the step size during parameter updates. To mitigate the risk of gradient explosions, the gradients are clipped by specifying a clipnorm value of 1. The model is compiled using categorical cross-entropy as the loss function, appropriate for multi-class classification tasks. This loss function measures the discrepancy between the predicted probabilities and the ground truth labels. The SGD optimizer and categorical cross-entropy loss function are then combined with the model.

Results on the original dataset (unbalanced):

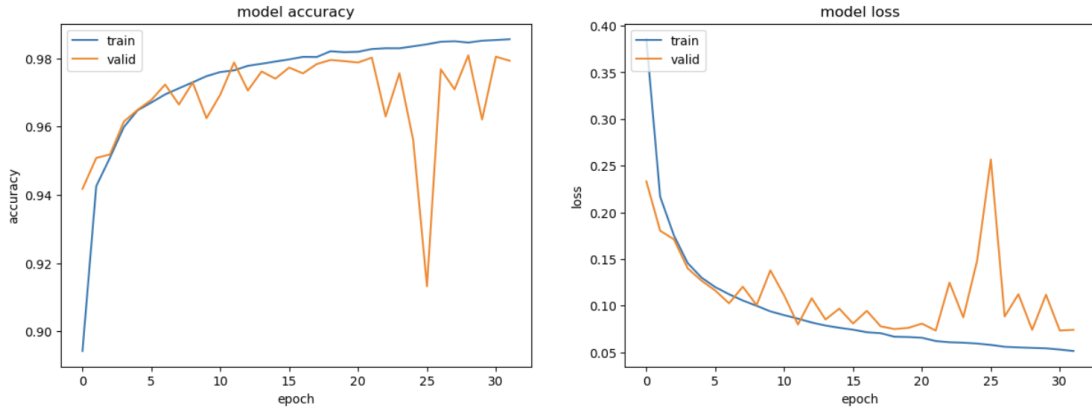


Figure 4: CNN Lstm results

The model achieved an accuracy of 97.84% on the test set and had high precision, recall, and F1-score for all classes. And here's the classification report to evaluate the performance of the model on the test dataset that includes metrics such as precision, recall, F1-score, and support for each class in your classification problem.

	precision	recall	f1-score	support
Normal	0.98	1.00	0.99	18118
Artial Premature	0.89	0.64	0.75	556
Premature ventricular contraction	0.95	0.94	0.95	1448
Fusion of ventricular and normal	0.93	0.50	0.65	162
Fusion of paced and normal	0.99	0.98	0.98	1608
accuracy			0.98	21892
macro avg	0.95	0.81	0.86	21892
weighted avg	0.98	0.98	0.98	21892

Figure 5: CNN Lstm Classification report

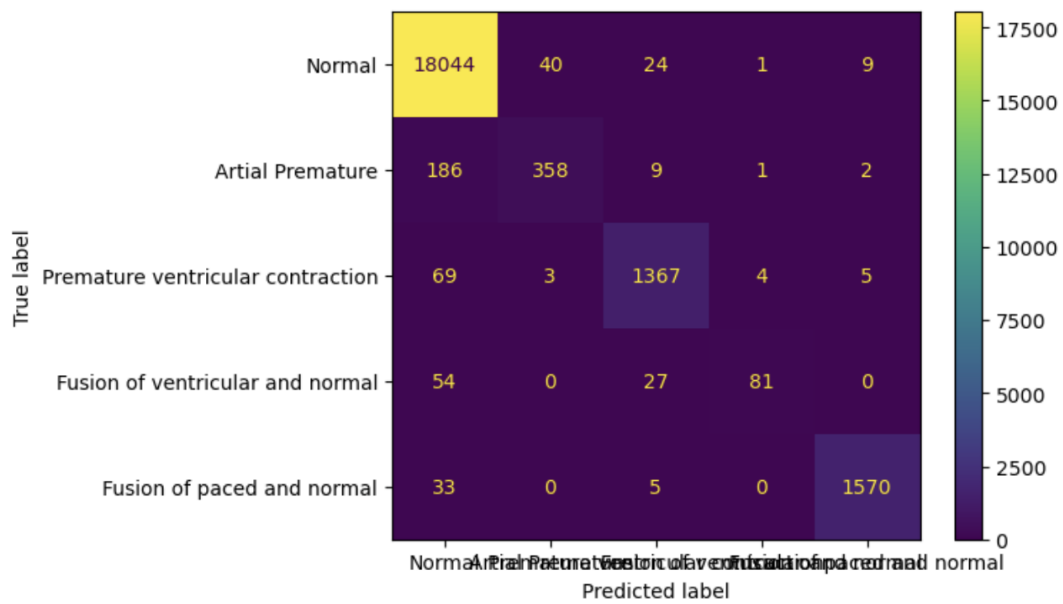


Figure 6: CNN Lstm Confusion matrix

Results on the balanced dataset:

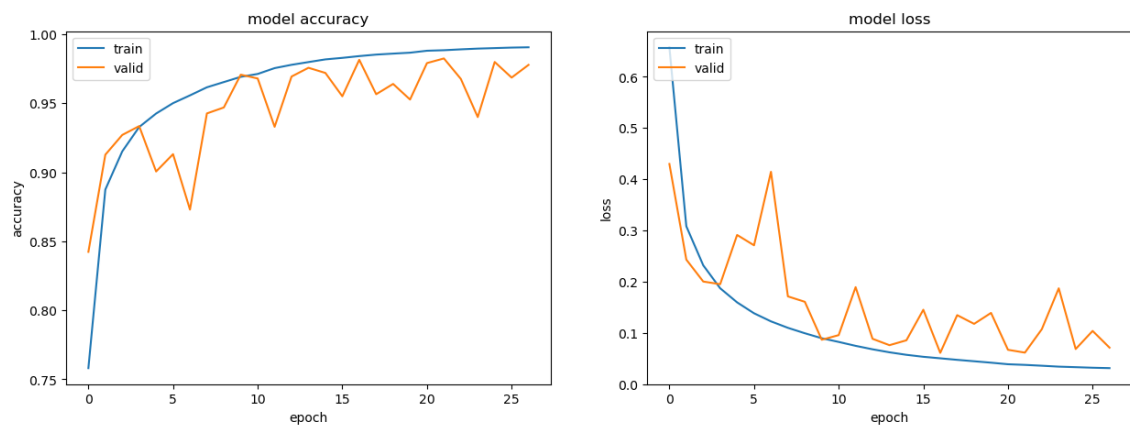


Figure 7: CNN Lstm results

	precision	recall	f1-score	support
Normal	0.99	0.94	0.97	18118
Artial Premature	0.42	0.84	0.56	556
Premature ventricular contraction	0.85	0.96	0.90	1448
Fusion of ventricular and normal	0.46	0.91	0.61	162
Fusion of paced and normal	0.97	0.99	0.98	1608
accuracy			0.94	21892
macro avg	0.74	0.93	0.80	21892
weighted avg	0.96	0.94	0.95	21892

Figure 8: CNN Lstm Classification report

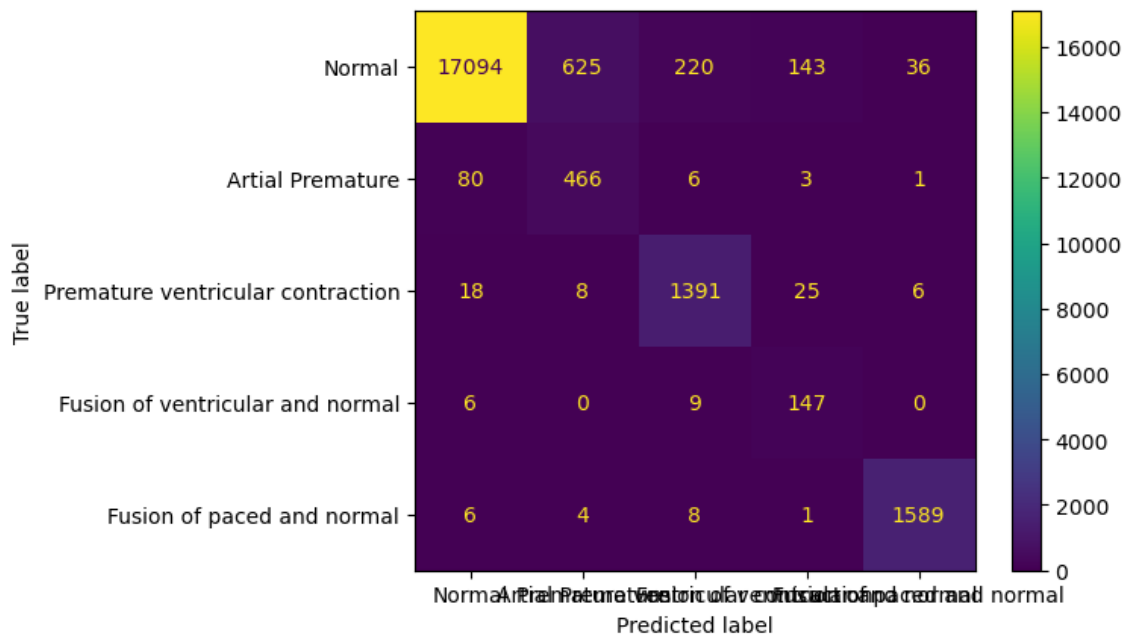


Figure 9: CNN Lstm Confusion matrix

3.3.2 CNN - BiLSTM

CNN-BiLSTM is a neural network architecture that combines the convolutional neural network (CNN) and bidirectional long short-term memory (BLSTM) models. Bidirectional Long Short-Term Memory (BiLSTM) is used to deeply mine the hierarchical and time-sensitive features of ECG data. The CNN component of the architecture is used to extract local features from the input sequence.

The model begins with a convolutional layer with 64 filters, a kernel size of 5, and a stride of 2. This layer is followed by a batch normalization layer, which normalizes the output of the previous layer to improve the stability and performance of the model.

The next layer is another convolutional layer with 128 filters, a kernel size of 5, and a stride of 2, followed by another batch normalization layer.

The third layer is a convolutional layer with 256 filters, a kernel size of 3, and a stride of 2, followed by another batch normalization layer.

The output of the third convolutional layer is then fed into a Bidirectional LSTM layer with 128 units, which processes the input sequence in both the forward and backward directions. This layer is followed by another Bidirectional LSTM layer with 256 units, which further processes the sequence and captures long-term dependencies.

A dropout layer with a rate of 0.2 is added to regularize the model and prevent overfitting.

Finally, the output of the second Bidirectional LSTM layer is fed into a dense layer with the number of units equal to the number of classes. The activation function of the dense layer is set to softmax, which produces a probability distribution over the output classes.

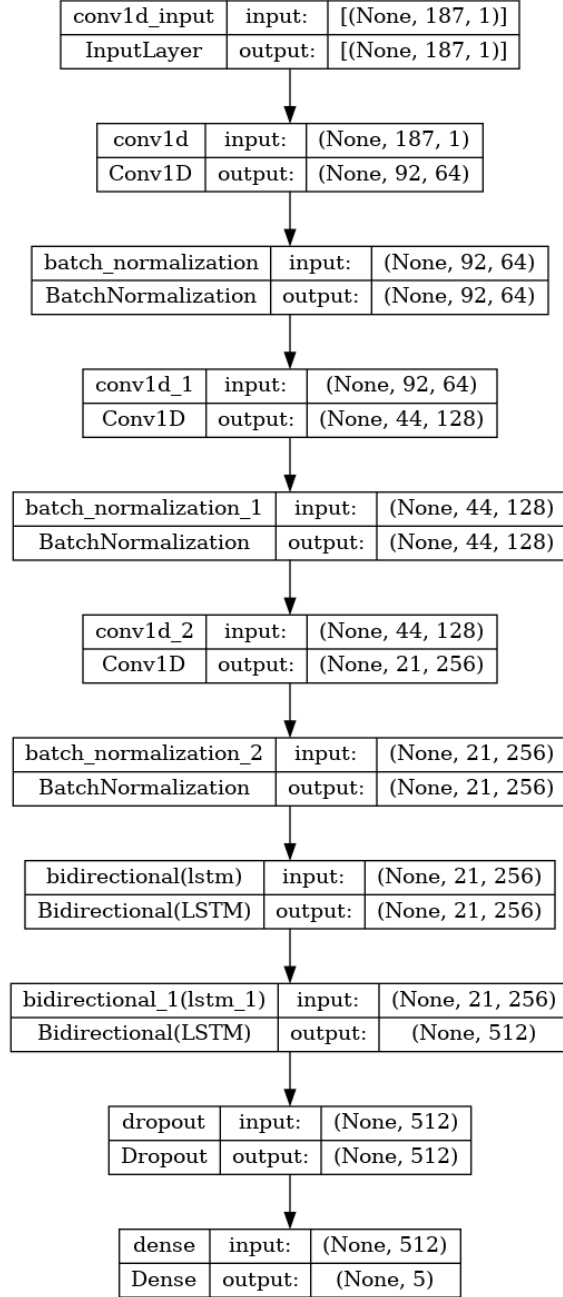


Figure 10: CNN Bilstm architecture

Results on the imbalanced dataset :

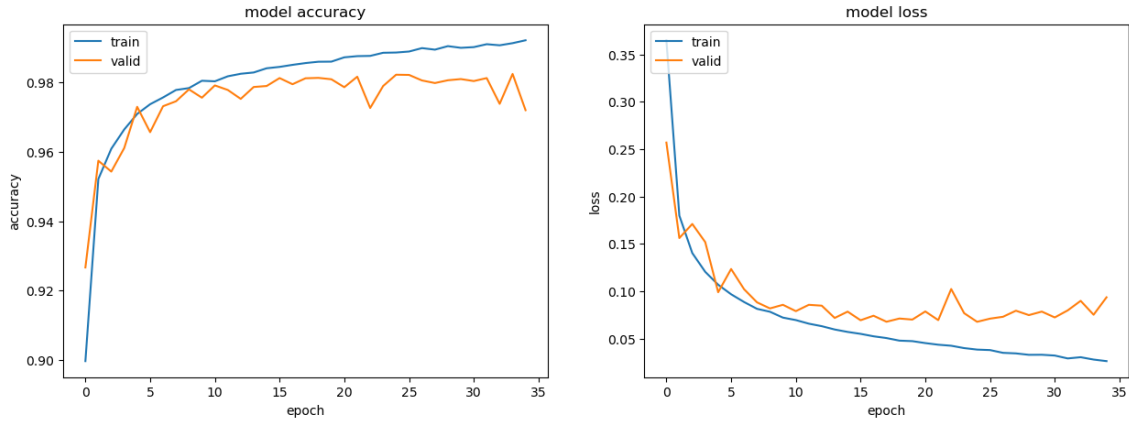


Figure 11: CNN bi-lstm learning curves

Results on the balanced dataset:

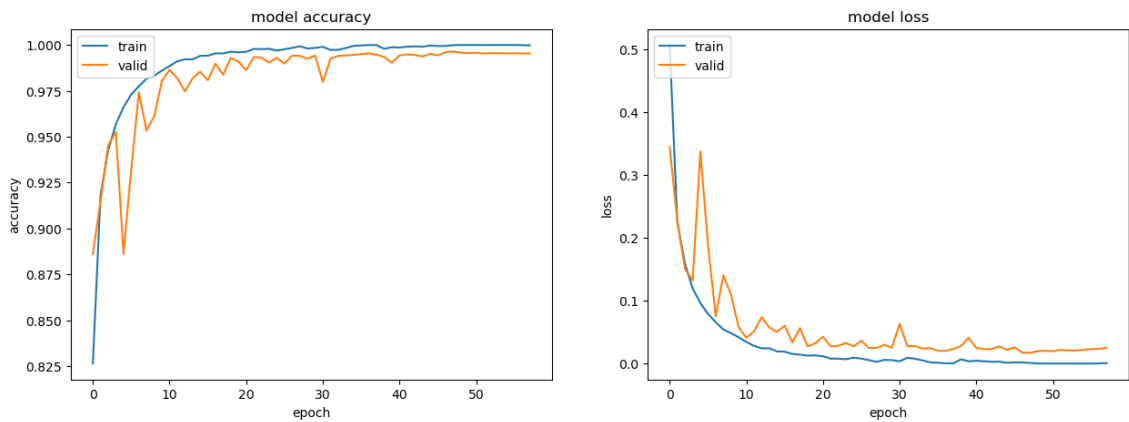


Figure 12: CNN bi-lstm learning curves

Training and testing:

The model achieved an accuracy of 98% on the test set and had high precision, recall, and F1-score for all classes. And here's the classification report to evaluate the performance of the model on the test dataset that includes metrics such as precision, recall, F1-score, and support for each class in your classification problem.

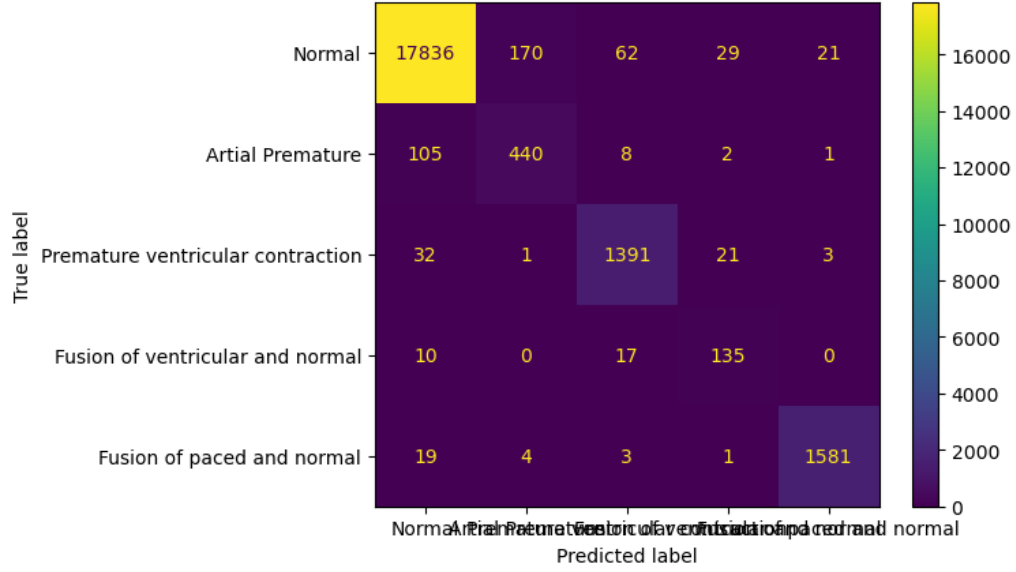


Figure 13: CNN Lstm Classification report

	precision	recall	f1-score	support
Normal	0.99	0.98	0.99	18118
Atrial Premature	0.72	0.79	0.75	556
Premature ventricular contraction	0.94	0.96	0.95	1448
Fusion of ventricular and normal	0.72	0.83	0.77	162
Fusion of paced and normal	0.98	0.98	0.98	1608
accuracy			0.98	21892
macro avg	0.87	0.91	0.89	21892
weighted avg	0.98	0.98	0.98	21892

Figure 14: Classification report cnn-Bilstm

3.3.3 LSTM Autoencoder

An LSTM Autoencoder is an implementation of an autoencoder for sequence data using an Encoder-Decoder LSTM architecture, an encoder-decoder LSTM is configured to read the input sequence, encode it, decode it, and recreate it. The performance of the model is evaluated based on the model's ability to recreate the input sequence

Model Architecture:

The LSTM autoencoder model architecture consisted of multiple LSTM layers, batch normalization, and dropout regularization

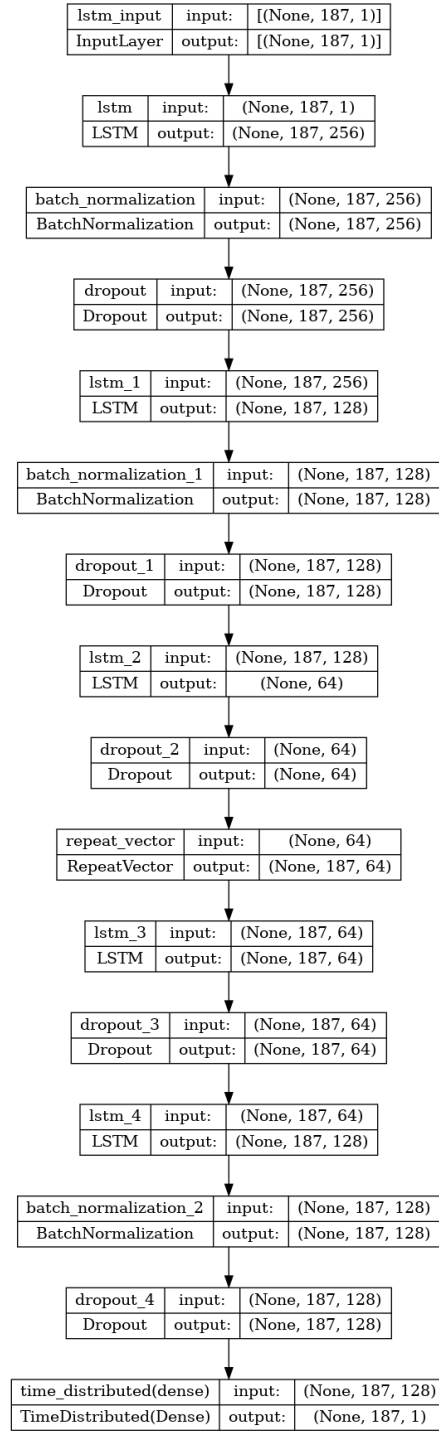


Figure 15: LSTM autoencoder architecture

Training and Testing:

The model was trained using the mean absolute error (MAE) loss function and optimized using the Adam optimizer. Early stopping was used to prevent overfitting. The training data(normal examples) was split into a training set and a validation set, with 10% of the data used for validation. The model was trained for 100 epochs with a batch size of 128.

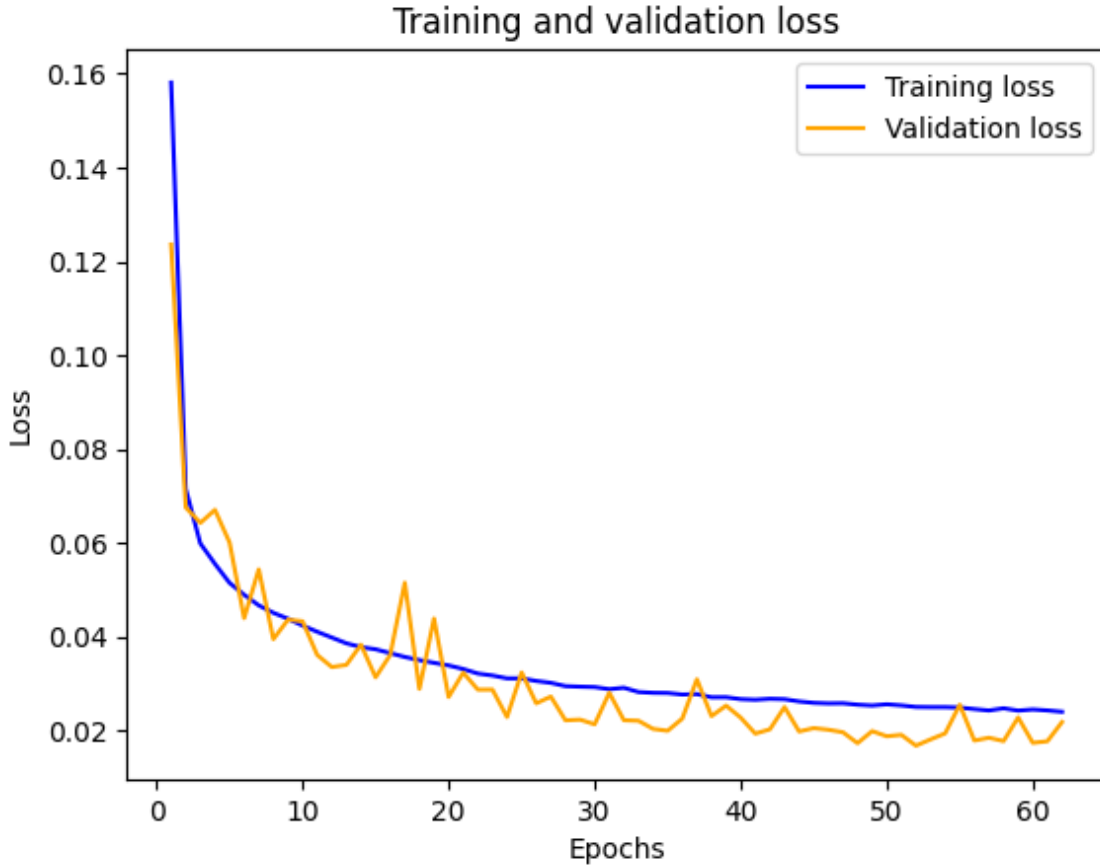


Figure 16: Lstm Auto-encoder results

After training our model, we chose a Threshold by identifying the anomalies by examining the reconstruction error on the training set. The mean and standard deviation of the MAE loss on the training set were used to calculate the threshold for anomaly detection.

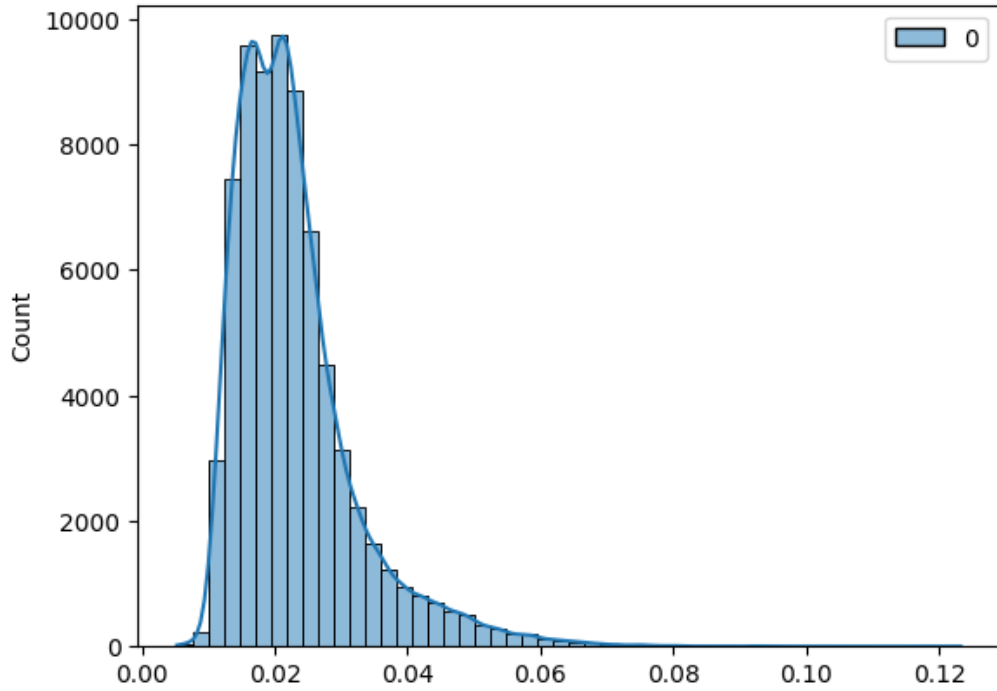


Figure 17: reconstruction error on the training set

Using the threshold, we can turn the problem into a simple binary classification task: If the reconstruction loss for example is below the threshold, we'll classify it as a normal heartbeat

Alternatively, if the loss is higher than the threshold, we'll classify it as an anomaly -to evaluate how well our model performs with normal heartbeats. The test sets containing normal heartbeats will be used.

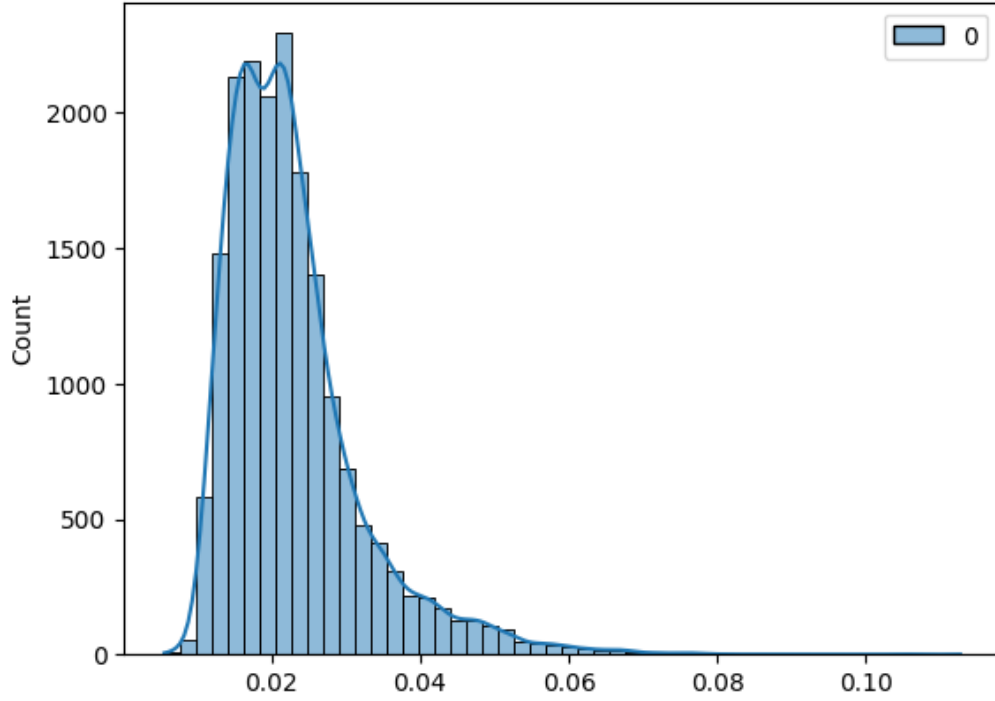


Figure 18: reconstruction error on normal testing set

The model was able to correctly predict 17,809 normal heartbeats out of 18,118 samples resulting in an accuracy of 98.3%. -Similarly, we can look at the anomalies examples The model was also able to correctly identify only 309 anomalies out of 18,118 samples resulting in a true positive rate of 1.7%

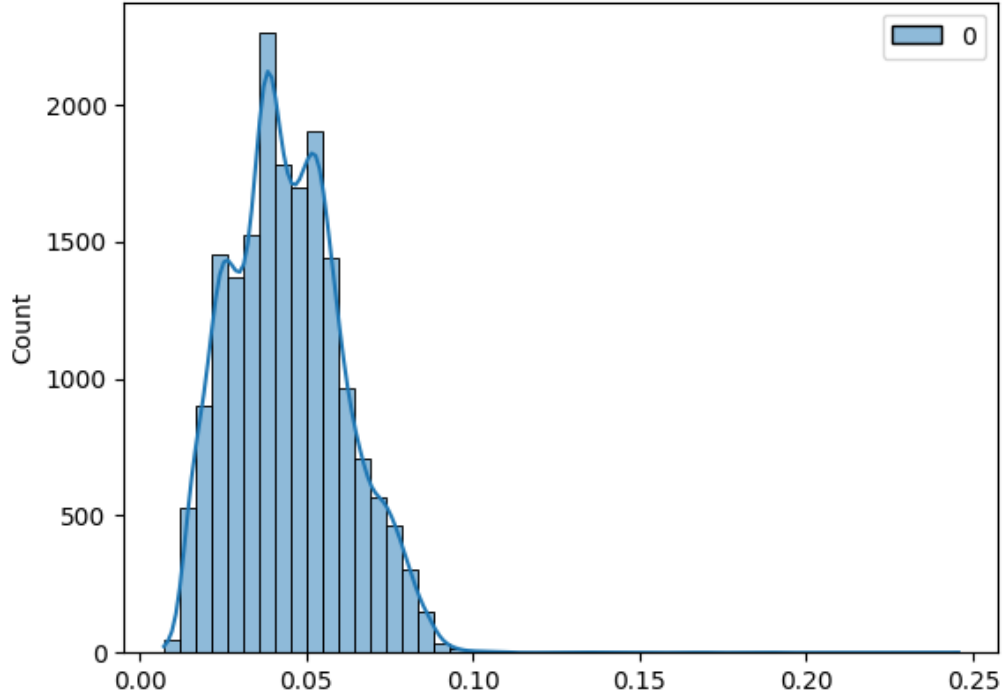


Figure 19: reconstruction error on anomalies testing set

The model could recognize normal heartbeats with high accuracy and detect anomalies with a low false positive rate.

4 Deployment

this application was developed using a combination of reactJs and fastApi. The application contains in total three interfaces

4.1 Home Page:

The home page of our application welcomes users with a visually appealing and intuitive interface. The layout is thoughtfully organized, featuring a top navigation bar that provides easy access to various sections of the application.

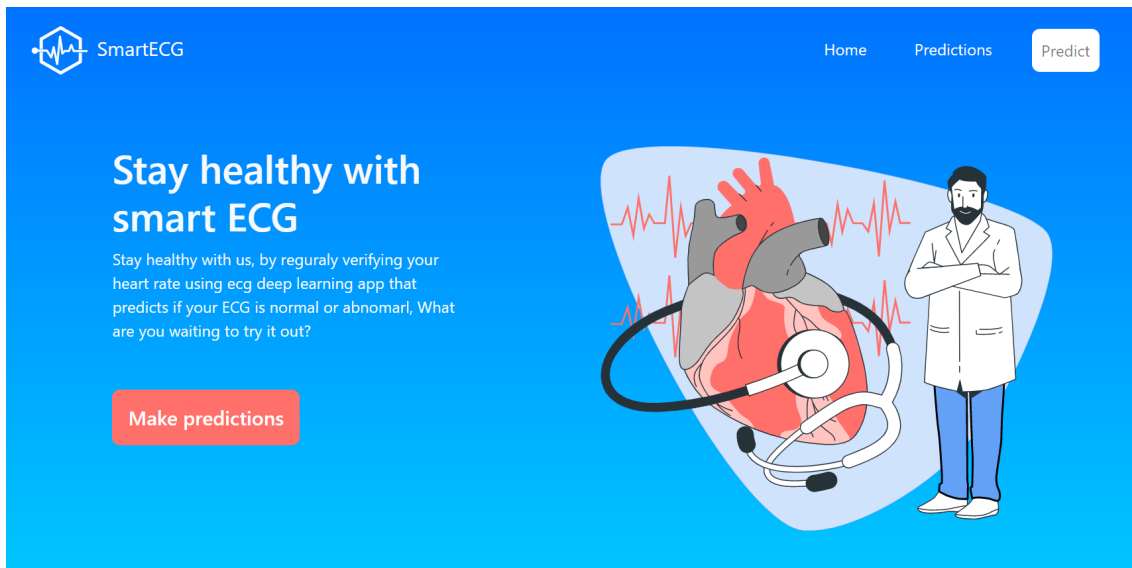


Figure 20: main page

4.2 Prediction Page

The prediction page of our application is where users can unlock the power of advanced algorithms and insightful analytics. Upon reaching this page, users are greeted with a well-structured and data-driven interface that empowers them to make informed decisions. Users can easily upload their datasets and predict the result.

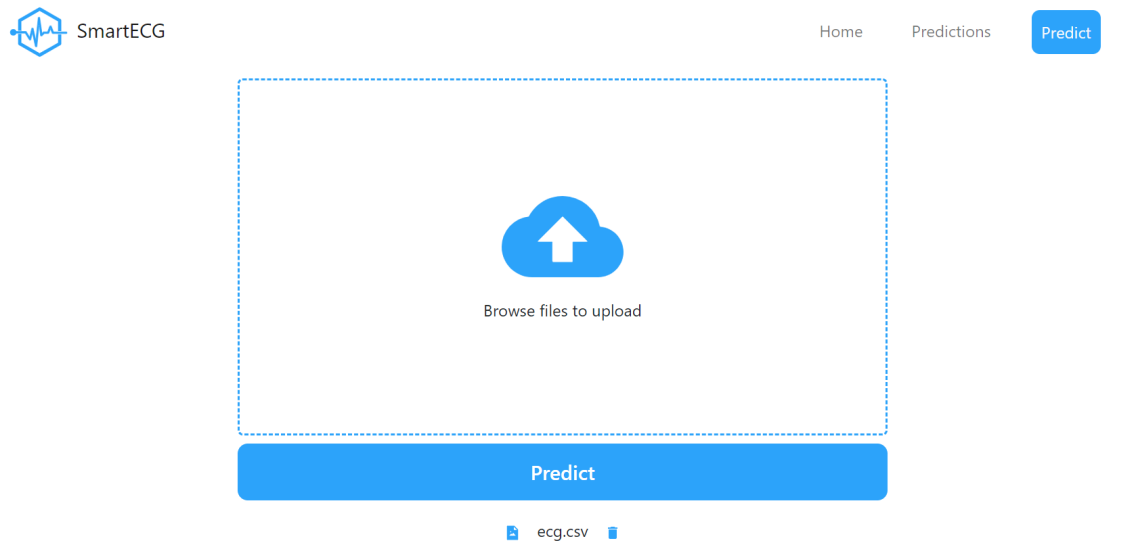


Figure 21: predict page

4.3 Result Page

The result page of our application appears to display the prediction labels of the data that was uploaded on the prediction page. The prediction labels are presented in a table format, which allows users to easily view and compare the different predictions made on their data.

SmartECG

Home Predictions Predict

Patient	Result	ECG
Patient 1	Normal	View
Patient 2	Normal	View
Patient 3	Normal	View
Patient 4	Normal	View
Patient 5	Normal	View

Figure 22: result page

On the result page, there is also an image of the ECG information. Users who want

to visually contrast the predicted labels with the actual data may find this to be useful.

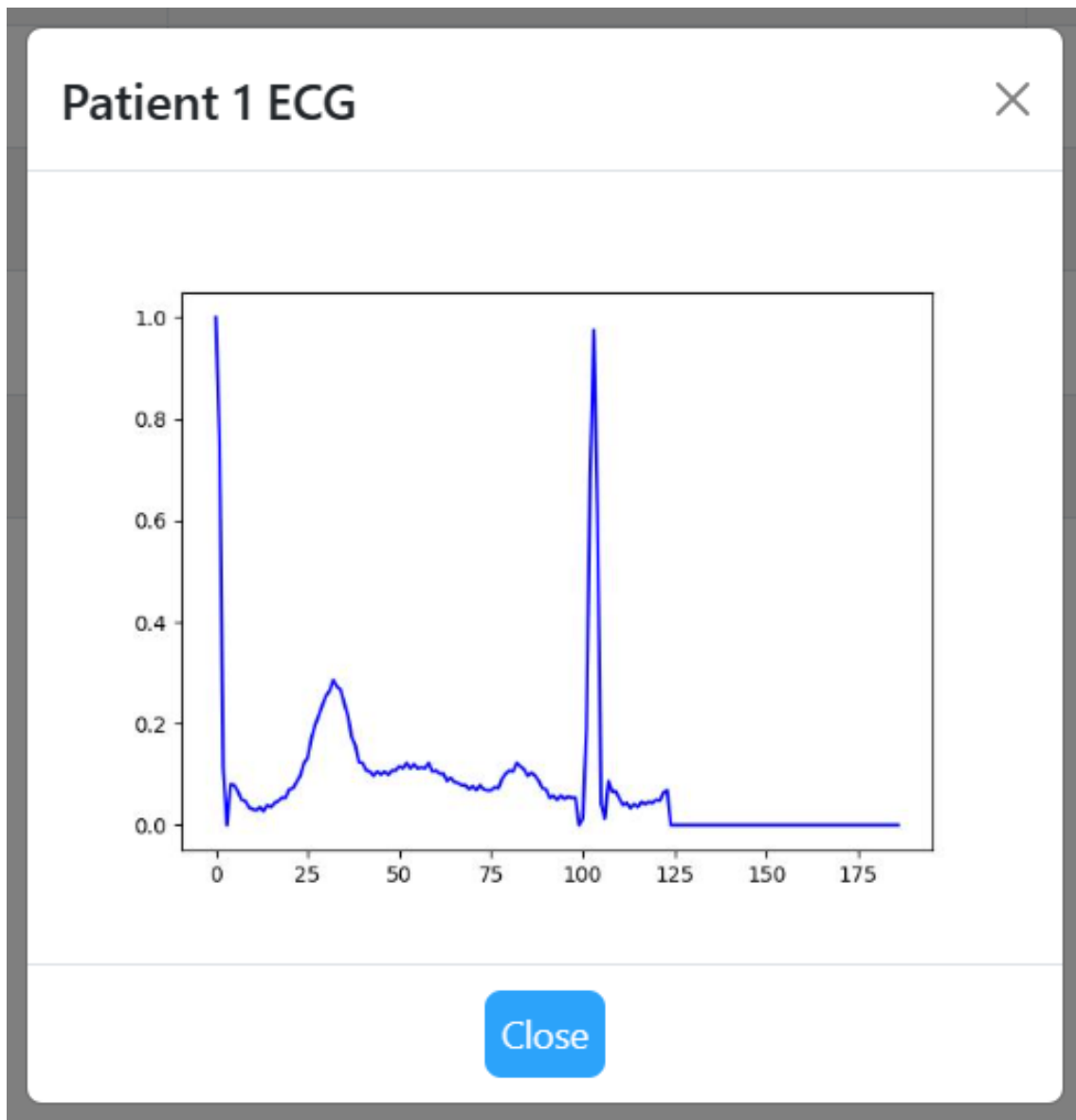


Figure 23: ECG plot window

5 Conclusion

In conclusion, this report has presented a method for ECG signal classification using a transferable representation approach. Through training deep neural networks on the MIT-BIH Arrhythmia Dataset[2], we have demonstrated the effectiveness of our method in accurately classifying ECG heartbeats. The results show that our suggested approach achieves comparable accuracies to the state-of-the-art methods[1] found in the literature. These findings highlight the potential of transferable representations in ECG signal classification and offer promising avenues for further research and development in this field. By leveraging the power of deep learning techniques, we can improve the accuracy and efficiency of ECG analysis, leading to better patient care, early detection of cardiac abnormalities, and more effective treatment strategies.

6 References

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)*, pages 443–444. IEEE, 2018.
- [2] G.B. Moody and R.G. Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.