Arrhythmia classification using ML/DL

Group 7: *Anwesha Samaddar, Hemanth Tirupati, Manpreet Singh and Sharan Rajamanoharan*

Abstract

Cardiac arrhythmias, or irregular heartbeats, occur due to disruptions in the electrical signals to the heart muscles. An electrocardiogram (ECG) is a non-invasive method used to detect and analyze cardiac signals. However, because heart signals are complex, simple algorithms struggle to classify them as normal or abnormal. Moreover, different types of arrhythmia classification require a deeper understanding of ECG signals. Our project aims to implement machine learning and deep learning algorithms to detect and classify arrhythmia into five different classes. We analyze various signal classification methods to comprehend their functionality. After comparing the performance metrics of different methods such as Random Forest, CNN, and LSTM, we demonstrate that while various methods achieve good test accuracy and F1 scores, LSTM models stand out as they are both fast and accurate. Moreover, they are well suited for continuous and remote monitoring, and they can be efficiently deployed on small IoT devices.

I. Introduction

Cardiovascular disease is a major global problem, particularly among the elderly, accounting for almost 31 percent of all deaths each year [1]. Current statistics show a growing elderly population and a corresponding rise in cardiovascular diseases, making it a top public health problem [2]. The electrocardiogram (ECG) is an excellent tools for recording the electrical activities of the heart, allowing medical experts to determine its condition. However, accurately interpreting these electrical signals requires a high level of knowledge from the observer and is prone to errors. Moreover, as the number of heart-related disorders increases and new diseases appear, the task of detection becomes more complex. Thus, the implementation of automated detection and classification systems becomes essential, making current task of manual classification easier, faster and cost effective.

Researchers have used traditional machine learning techniques like SVM , Naive Bayes classifier, and decision trees classifier to tackle this issue. While these methods can provide decent accuracy and improve classification, they have limitations. They work well with small datasets and extract only basic features. In our experiments with these methods, we have found that there is still room for improvement in the results. Better algorithms like Random Forest, which uses a large number of trees, can offer better predictions. However, they may take longer to run or generate results.

Recent research has shifted its focus to deep learning networks, which can extract complex features from data. Models such as Multi-layer Perceptron (MLP), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks have performed well in this area. However, when training on small datasets such as the MIT-BIH dataset, which comprises only 48 ECG recordings, employing a large number of network layers can result in overfitting. In our work, we avoid this problem by distributing the dataset into sizes that accommodate deep learning models. We will standardize the data and merge various beat signals present in the MIT-BIH dataset using the Association for the Advancement of Medical Instrumentation (AAMI) standard. This approach will facilitate better feature extraction when utilizing neural networks. Additionally, we aim to achieve a trade-off between the model parameters of deep learning models and the test accuracy of those models. This can allow the development of lightweight models that achieve high accuracy, which can be integrated into small IoT devices.

Our detection and classification process involves several steps: extracting data from the MIT-BIH dataset, QRS peak detection and data distribution, creating subclasses for classification, and training machine learning and deep learning models. The MIT-BIH dataset is chosen because it is the most commonly used database for arrhythmia classification purposes. Within this dataset, we have 48 ECG recordings, each lasting 30 minutes. Each ECG signal comprises distinct components including the P wave, QRS complex, and T wave (see Fig. 1). Analyzing the variation in these waves enables the diagnosis of numerous cardiac conditions, including arrhythmia. We divide the dataset using QRS peak detection to extract single beats (see Dataset section). Extracting a single beat from a 30-minute dataset will create a large database for classification. Next, for each beat, we divide the QRS complex into five classes for arrhythmia classification . These five classes are: N (normal), F (fusion), Q (unknown), V (ventricular ectopic), and S (supraventricular ectopic). Finally, we use various models like Random Forest, MLPs, CNN, LSTM, etc., to compare the performance metrics (see Results section). We achieve high test accuracy ($\geq$ 95 percent) and a good weighted average F1 score ($\geq$ 0.95) for all the models. However, we observe that LSTM's performance is good even compared with fewer trainable parameters, making it a suitable and lightweight model for small applications.

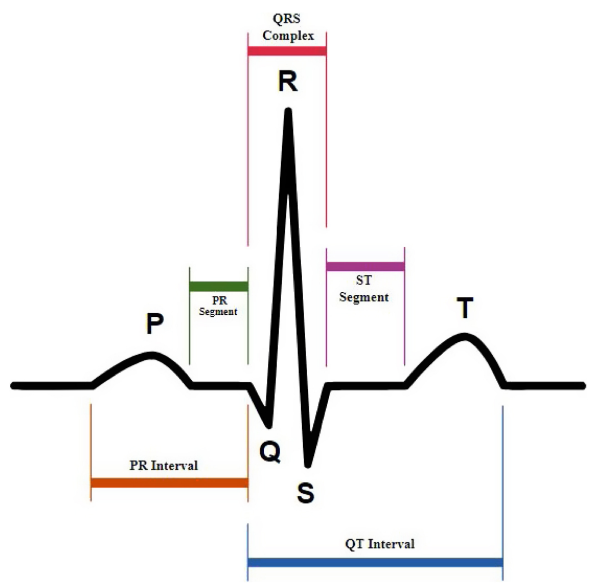


Fig. 1: An PQRST wave present in ECG signal. QRS complex is detected in MIT-BIH for our case. Source [3]

We compare and summarize ML and DL methods that exist in arrhythmia classification. Some of the contributions are:

- **Exploring Various Neural Networks for ECG Interpretation**: We evaluate MLPs, CNNs, LSTMs, and other models to find the best for ECG interpretation.
- **Classifying Arrhythmias into 5 Distinct Classes**: We categorize arrhythmias into N, F, Q, V, and S classes, improving diagnosis and treatment.
- **Comparing ML and DL Methods**: By comparing ML and DL methods, we help researchers choose the most effective techniques.

II. Background

Approximately 1.5% to 5% of the general population has arrhythmias or deviations in the normal heart rhythm [4]. Among the various types of arrythmias Atrial fibrillation (AFib) is the most frequent kind. The acute nature of some arrythmias and due to the fact that some occurrences go undiagnosed, it is challenging to estimate their prevalence. Assessment of arrhythmia typically involves an electrocardiogram. Various monitoring methods, such as ambulatory ECG monitoring and event recorders, help detect sporadic arrhythmias. Arrhythmias can be characterized by heart rate, with bradyarrhythmias (heart rate $<$ 60 bpm) and tachyarrhythmias (heart rate $>$ 100 bpm). Tachyarrhythmias are further classified by their origin (supraventricular or ventricular) and QRS complex duration (narrow or wide). This classification system acts as a guide for selecting relevant treatment approaches.

ECG signal analysis is divided into many stages: pre-processing, peak segmentation, feature extraction, and classification. Previous approaches for feature extraction included features obtained with statistical methods, Independent Component Analysis (ICA), and Discrete Wavelet Transform (DWT), which were then combined with classifiers such as KNN, Decision Trees (DT), Probabilistic Neural Networks (PNN), and Support Vector Machines (SVM) for classification.

Taiyong et. al [5] utilized WPE for feature extraction and Random Forests for arrythmia classification on the MIT-BIH dataset following the AAMI standards. Their technique achieves excellent classification accuracy (94.61%) on the test dataset and effectively captures signal complexity using entropy. One disadvantage is the large dimensionality of the features, which can be addressed with dimensionality reduction approaches such as the PCA. Recent improvements have evolved toward learning-based systems, which use techniques such as deep learning to achieve more sophisticated feature extraction and categorization. This shift increases the amount of information collected from ECG signals, which improves arrhythmia detection accuracy.

Pandey et. al. [6] proposed a novel CNN model for classifying ECG signals from the MIT-BIH dataset according to ANSI-AAMI guidelines. They have combined data preprocessing, segmentation, feature extraction, and classification into a single structure thus eliminating the need for multiple data processing stages. For addressing class imbalance, key additions include an 11-layer CNN architecture and the Synthetic minority oversampling technique (SMOTE). Their proposed method is an end to end structure that achieved superior performance on test dataset (approx 98.30%). However, the drawback lies in utilising SMOTE for oversampling minority class samples. This brings the risk of overfitting as the synthetic data may not accurately capture the minority class distribution.

Fig. 2 illustrates how after applying SMOTE oversampling the classes of S, V, F and Q have a huge number of synthetically generated samples. Using such a huge number of synthetic samples in the training data is bound to cause overfitting. SMOTE generated data may also introduce noise that would negatively impact model performance and interpretability. Fig. 3 shows the architecture proposed by [6], which consists of four convolution and maxpooling layers, followed by 3 fully connected layers.

Heart beat types	Number of heart beats in real dataset	After oversampling, number of heartbeats in training set				
		50%	60%	70%	80%	90%
N	87542	43746	52475	61261	70039	78840
S	2726	43746	52475	61261	70039	78840
V	7221	43746	52475	61261	70039	78840
F	802	43746	52475	61261	70039	78840
Q	3888	43746	52475	61261	70039	78840
Total	102179	218730	262375	306305	350195	394200

Fig. 2: Total number of heartbeats for each category before and after using SMOTE oversampling. Source [6]

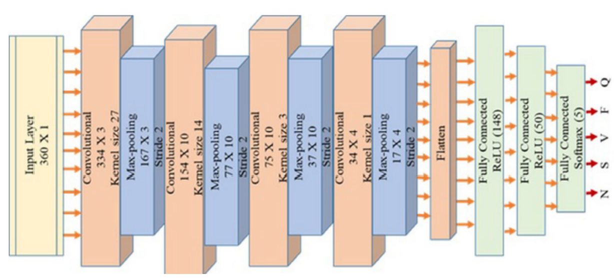


Fig. 3: CNN architecture proposed by Pandey et al. Source [6]

Sowmya et. al. [7] suggested a unique approach by combining a Convolutional Neural Network with an LSTM (Long Short-Term Memory) network for arrythmia classification on MIT-BIH dataset. This hybrid architecture efficiently incorporates the spatial and temporal information present in ECG signals, which improves classification performance. For feature extraction they have used wavelet transform and denoising autoencoders to improve the input features' quality.

Although the hybrid CNN-LSTM model has several advantages like improved accuracy (97%) in identifying ECG signals, and appropriate spatio-temporal feature representation, it can be computationally difficult, having a much higher number of training parameters and increased time for inference. This might limit the usability of CNN-LSTM on low memory devices. Moreover balanced datasets are required for effective training, which is a hindrance in cases of data scarcity.

We perform a comparative analysis of ML algorithms like Logistic Regression, Random Forest and Deep learning approaches like MLP (multi-layer perceptron), CNN, LSTM and CNN-LSTM for arrythmia classification into five categories based on AAMI guidelines. The hyperparameter tuning process includes increasing or reducing the number of hidden layers and nodes, adjusting the activation function, the number of training epochs, and modifying the learning rate. Categorical cross-entropy loss is used for calculation of multi-class classification loss.

Our primary goal is to achieve a trade-off between the number of training parameters and classification performance.The network enhancements were made with the intention of creating a model that generates inference in real time while remaining lightweight enough to run on edge devices.

III. Dataset

The MIT-BIH database [8] was collected through the joint efforts of Boston's Beth Israel Hospital (BIH) and Massachusetts Institute of Technology (MIT) labs from their patients. Since its initial release to public in 1980, it has become a popular dataset for testing the accuracy of arrhythmia detectors. The dataset comprises 48 ECG recordings from 47 patients, including 25 males and 22 females, as referenced in [9]. Male patients range in age from 30 to 90 years old, while females range from 22 to 90. Each recording spans 30 minutes with a sampling frequency of 360 Hz and 11-bit resolution. Typically, ECG data is collected using a 12-lead configuration with electrodes attached to different parts of the body.

Class	Beat annotation
N (Normal)	N (Normal) L (Left bundle branch block beat) R (Right bundle branch block beat) B (Ventricular bigeminy)
S (Supraventricular ectopic beat)	A (Atrial premature beat) a (Aberrated atrial premature beat) J (Nodal (junctional) premature beat) S (Supraventricular premature beat) e (Atrial escape beat) n (Supraventricular escape beat)
V (Ventricular ectopic beat)	V (Premature ventricular contraction) r (R-on-T premature ventricular contraction) E (Ventricular escape beat)
F (Fusion beat)	F (Fusion of ventricular and normal beats)
Q (Unknown beat)	Q (Unclassifiable beat) ? (Beat not classified during learning) / (Paced beat) f (Fusion of paced and normal beat)

Fig. 4: Beat annotation according to AAMI standard. [10]

For arrhythmia detection, the focus is on identifying the PQRST wave, which is captured as the potential difference between the right arm and left leg electrodes, known as lead II signals. The voltage values in this context range between -5 mV to +5 mV. Initially, the dataset had some issues with specific patient recordings, which were excluded during dataset preparation. However, we are utilizing a newer version of the dataset that has addressed these signal problems. While there are numerous ECG datasets available with enhancements in recording procedures, ease of extraction, and larger data sizes, we have chosen the MIT-BIH dataset due to its extensive utilization in literature. By comparing various models used in machine learning and deep learning, we aim to gain a clearer understanding of their performance metrics.

The original data contains a lot of beat signals, which require some filtering to detect arrhythmia. We are going to use the AAMI medical standard to make the data cleaner. In Fig. 4, we can see the five classes and corresponding beat symbols extracted from the MIT-BIH dataset. Plotting the combined values of all classes, we can observe the PQRST wave in Fig. 5. As evident from Fig. 5, we have multiple peaks for a single patient in a 30-minute recording. With a sampling frequency of 360 Hz, we obtain 360 samples in 1 second. We divide our dataset into 200-sample lengths, with each sample centered at QRS-detected waves. The overall process will result in our dataset having a size of 107,400 by 200. Additionally, we store a class label column corresponding to each row, making the data shape 107,400 by 201. This data preparation enables better feature extraction using ML and DL models.

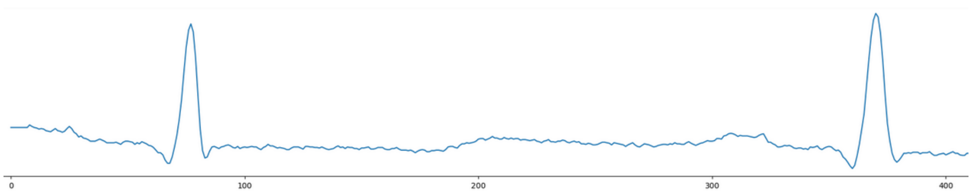


Fig. 5: A PQRST wave generated from single patient.

In Fig. 6, we can observe the presence of 5 different classes within a 200-sample length. Each signal possesses a unique shape, which we can learn using our model for arrhythmia classification.

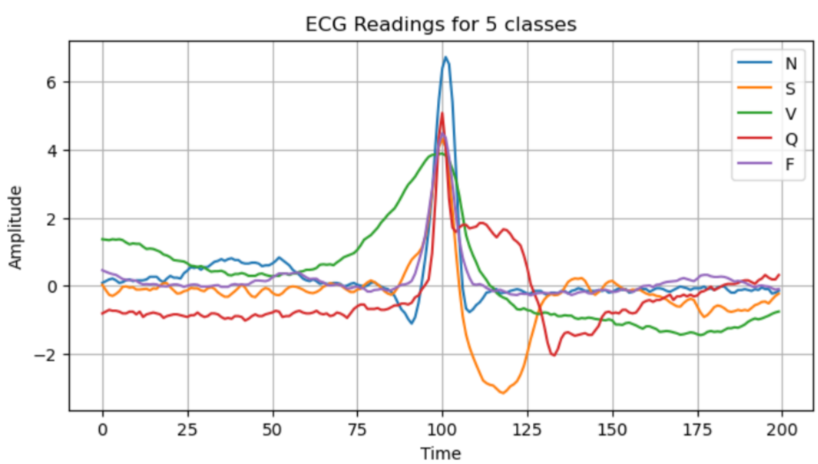


Fig. 6: The 5 ECG classes visualized

IV. Methodology

Fig. 7 depicts the flowchart of our project's workflow. The initial step involves data selection, preprocessing and feature extraction. As indicated in the Dataset section, our dataset is obtained from MIT-BIH. Additionally, in the dataset section, we detail the preprocessing and feature extraction stages, which comprises of data normalization and scaling, QRS detection, and the extraction of 200 samples to the left and right of the peak. This process generates a sizable dataframe with dimensions of 107400 by 200, as explained earlier. With our dataset prepared, next step is to classify arrhythmia using machine learning and deep learning models.

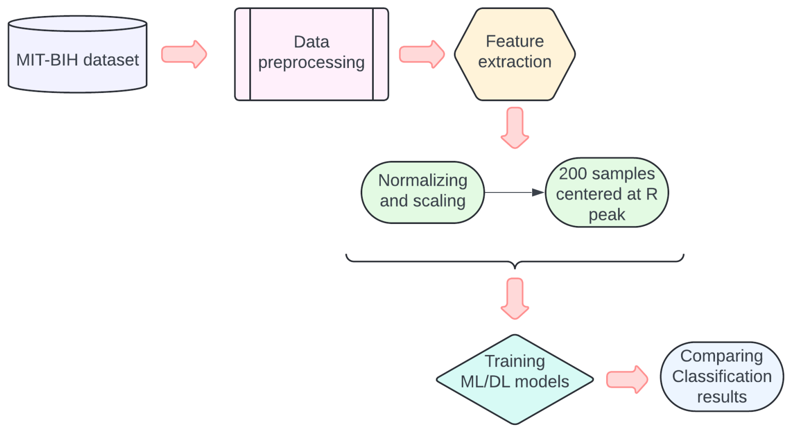


Fig. 7: Proposed approach for data pre-processing, training and comparison of results

Before using machine learning models, we partition the dataset into training, validation, and test sets using a 70-20-10 ratio. Then, we employ Python libraries to train Logistic Regression and Random Forest classifiers on the divided dataset. Finally, we validate and test the results of these models. For logistic regression method, it employed a Bayesian optimization approach via BayesSearchCV, to search for the best combination of hyperparameters such as the regularization strength, the solver algorithm, and the multiclass strategy to maximize the model's performance on the given dataset. The Random Forest Classifier employed the Gini impurity criterion to make decisions at each split of the decision trees. This based method aims to improve predictive accuracy by aggregating the predictions of multiple decision trees trained on random subsets of the data.

For the deep learning model, we maintain the same split as before. A deep learning model consists of multiple layers that act as feature extractors to effectively capture signal characteristics. The models used in our work are MLP, CNN, LSTM, and a combination of CNN-LSTM.The MLP consists of four fully connected layers followed by ReLU activation functions and dropout regularization. For the CNN and LSTM, we draw inspiration from the networks proposed in [11] and [12] respectively.

The CNN model features convolutional layers with ReLU activations, including max-pooling and batch normalization. It also incorporates fully connected layers with dropout regularization before the final output layer for classification.The LSTM model consists of two LSTM layers with 200 input features and 50 hidden units each, followed by a fully connected layer with ReLU activation and dropout regularization. It concludes with a final linear layer for classification. The CNN-LSTM model integrates convolutional and LSTM layers. It processes data through convolution and pooling, then feeds it to an LSTM layer for sequence learning. Finally, it employs fully connected layers with dropout for classification. We can see the summary of model in Fig. 8

In our evaluation of models, the main assessment metric is the confusion matrix. As shown in the results section, we present the confusion matrices for five classes and compute the weighted average F1 score for all the models. The weighted average F1 score is a measure that considers both the precision and recall of a classification model, providing a balanced assessment of its performance across all classes.

Layer (type)	Output Shape	Param #
Linear-1	[-1, 100]	20,100
ReLU-2	[-1, 100]	0
Dropout-3	[-1, 100]	0
Linear-4	[-1, 50]	5,050
ReLU-5	[-1, 50]	0
Dropout-6	[-1, 50]	0
Linear-7	[-1, 25]	1,275
ReLU-8	[-1, 25]	0
Dropout-9	[-1, 25]	0
Linear-10	[-1, 5]	130

Total params: 26,555
Trainable params: 26,555
Non-trainable params: 0

Input size (MB): 0.00
Forward/backward pass size (MB): 0.00
Params size (MB): 0.10
Estimated Total Size (MB): 0.11

(a) MLP summary

Layer (type)	Output Shape	Param #
Conv1d-1	[-1, 16, 198]	64
ReLU-2	[-1, 16, 198]	0
Conv1d-3	[-1, 32, 196]	1,568
ReLU-4	[-1, 32, 196]	0
MaxPool1d-5	[-1, 32, 195]	0
BatchNorm1d-6	[-1, 32, 195]	64
Conv1d-7	[-1, 64, 193]	6,208
ReLU-8	[-1, 64, 193]	0
Conv1d-9	[-1, 128, 191]	24,704
ReLU-10	[-1, 128, 191]	0
MaxPool1d-11	[-1, 128, 190]	0
BatchNorm1d-12	[-1, 128, 190]	256
Flatten-13	[-1, 24320]	0
Linear-14	[-1, 100]	2,432,100
ReLU-15	[-1, 100]	0
Dropout-16	[-1, 100]	0
Linear-17	[-1, 50]	5,050
ReLU-18	[-1, 50]	0
Dropout-19	[-1, 50]	0
Linear-20	[-1, 5]	255

Total params: 2,470,269
Trainable params: 2,470,269
Non-trainable params: 0

(b) CNN summary

Layer (type)	Output Shape	Param #
lstm.weight_ih_l0	[-1,200]	40000
lstm.weight_hh_l0	[-1,50]	10000
lstm.bias_ih_l0	[-1]	200
lstm.bias_hh_l0	[-1]	200
lstm.weight_ih_l1	[-1,50]	10000
lstm.weight_hh_l1	[-1,50]	10000
lstm.bias_ih_l1	[-1]	200
lstm.bias_hh_l1	[-1]	200
fc.1.weight	[-1,50]	1250
fc.1.bias	[-1]	25
final.weight	[-1,25]	125
final.bias	[-1]	5
Total params:		72205

(c) LSTM summary

Layer (type)	Output Shape	Param #
conv1.0.weight	[-1,1,3]	96
conv1.0.bias	[-1]	32
fc1.1.weight	[-1,3168]	633600
fc1.1.bias	[-1]	200
lstm.weight_ih_l0	[-1,200]	40000
lstm.weight_hh_l0	[-1,50]	10000
lstm.bias_ih_l0	[-1]	200
lstm.bias_hh_l0	[-1]	200
lstm.weight_ih_l1	[-1,50]	10000
lstm.weight_hh_l1	[-1,50]	10000
lstm.bias_ih_l1	[-1]	200
lstm.bias_hh_l1	[-1]	200
fc2.1.weight	[-1,50]	1250
fc2.1.bias	[-1]	25
final.weight	[-1,25]	125
final.bias	[-1]	5
Total params:		706133

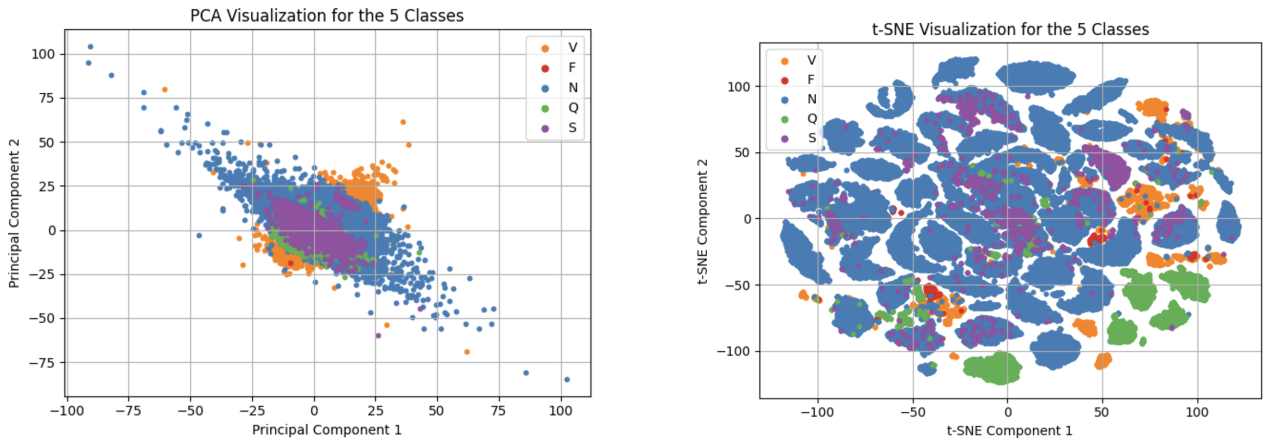
(d) CNN LSTM summary

Fig. 8: The deep learning models

In addition to this, we include test accuracy and validation loss for DL models, along with the number of trainable parameters. Test accuracy reveals how well the model performs on data it hasn't seen before, reflecting its ability to generalize, whereas validation loss measures the degree of error during training, with lower values indicating better performance. The number of trainable parameters offers an understanding of model complexity and resource requirements. This comprehensive evaluation approach considers accuracy, robustness, and efficiency across diverse classes of our models. We have used MSI for all of our computation needs. This entailed using a CUDA-enabled NVIDIA A40 GPU for deep learning algorithms.

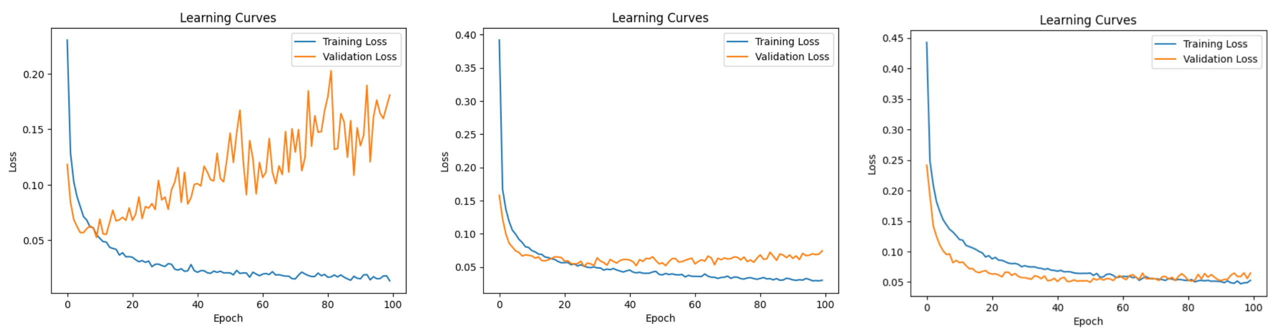
V. Results

Before we can perform classification, we visualize the ECG data. We used PCA and obtained 7 dimensions that explained 90% of variance. We plotted the first 2 principal components in Fig 9. PCA plot indicates the data is not linearly separable. We have also experimented with t-SNE, to visualize non-linear relationships between data points.



(a) PCA (b) t-SNE

Fig. 9: Visualization of ECG Data through PCA and tSNE Projections



(a) CNN (b) LSTM (c) CNN-LSTM

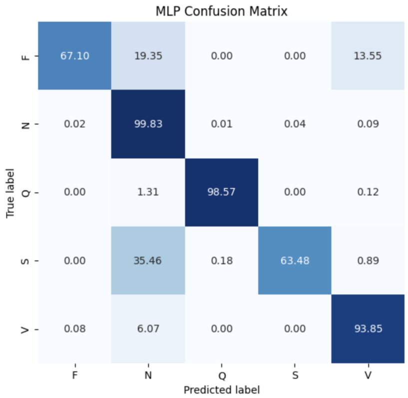
Hyperparameters	
Learning Rate	1.00E-03
Batch Size	100
No. Epochs	100
Optimizer	Adam
Train:Val:Test Split	70:10:20

(d) Hyperparameters

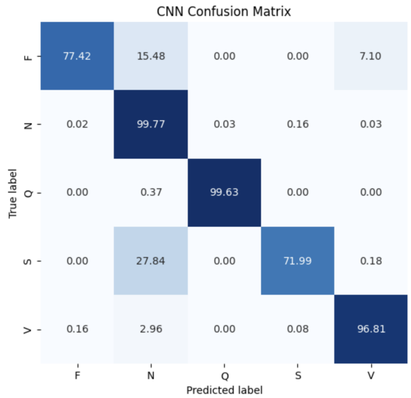
Fig. 10: Learning Curves of MLP, CNN, LSTM and CNN-LSTM Models and hyperparameters used.

We then apply various statistical and deep learning techniques to the ECG data. Fig 11 compares the performance of these techniques through confusion matrices. The hyperparameters we used for deep learning models are given in Fig 10. Further, the figure also depicts learning curves for CNN and LSTM training. We see that initially, both training and validation losses are reduced. However, with more epochs training loss is further reduced while validation loss is increasing depicting overfitting. We choose the model with the lowest validation loss to classify the test set.

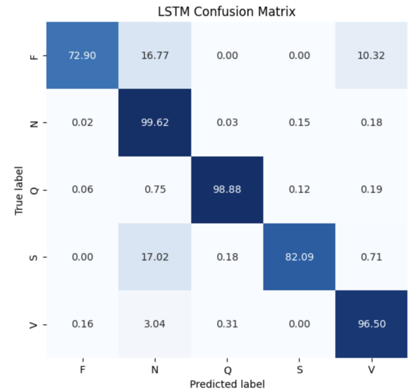
Our interpretation of the models is that linear models like Logistic Regression performed poorly (in comparison) corroborating PCA in Fig 9. CNNs excel in ECG classification because of their ability to capture local patterns and hierarchically learn more complex and global patterns. From Fig 6 we see that each Arrhythmia class has a unique signal pattern and CNNs learn to distinguish these patterns. This kind of ability to capture local and global patterns through a shared kernel is absent in MLP. A MLP looks at the



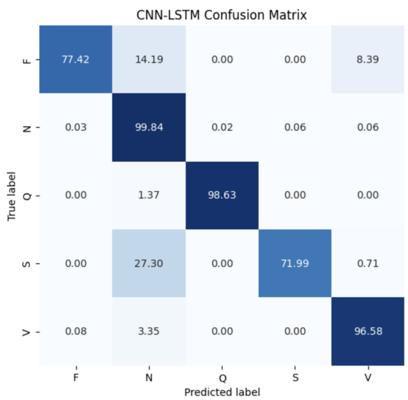
(a) MLP



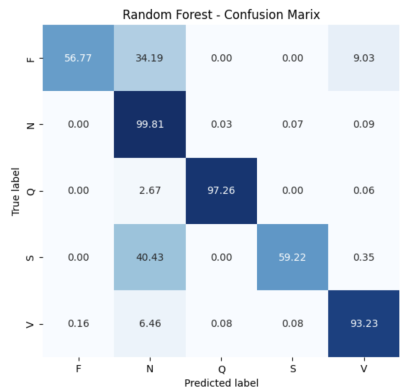
(b) CNN



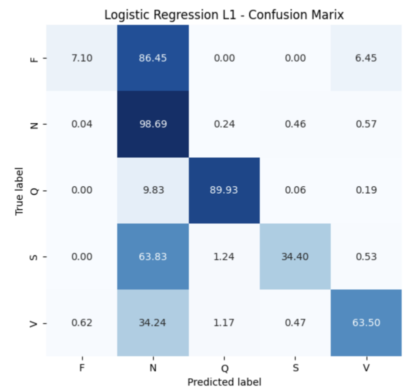
(c) LSTM



(d) CNN-LSTM



(e) Random Forest



(f) Logistic Regression

Fig. 11: Confusion Matrices for MLP, CNN, LSTM, CNN-LSTM, Logistic Regression Models

time-series data as n-dimensional data and tries to fit a non-linear classifier by memorizing information from training data in its parameters. Thus MLP is less generalizable while CNNs which learn underlying patterns that classify these Arrhythmias are more generalizable as demonstrated in higher Test Accuracy for CNNs compared to MLPs in Table I. LSTMs can capture long-term dependencies/global patterns with a lesser number of parameters compared to CNNs. CNNs capture global patterns by hierarchically combining local patterns resulting in greater number of parameters. However, LSTMs capture this efficiently using gating mechanisms that remember long-term dependencies while forgetting irrelevant noise. Thus using LSTM we've achieved high accuracy with very few parameters as depicted in Table I.

VI. Discussion

Our project aims to utilize various machine learning and deep learning algorithms, including Logistic Regression (LR), Random Forest (RF), MLP, CNN, LSTM, and CNN-LSTM, to classify ECG signals into five categories for arrhythmia detection. As evident from the confusion matrices, algorithms like LR are unable to generate highly accurate predictions, as the models do not capture the non-linearity or time dependencies in the data. LR assumes there are linear relationships between the input features and output classes.

We use hyperparameter tuning to obtain the optimal architecture for deep learning networks that have the lowest validation loss and lowest classification error on the test set. Additionally, we want to create networks that can generate inference in lightweight devices with fewer training parameters. This is emphasized due to the fact that a well-generalizable model with little processing and storage needs would be ideal for remote monitoring on IoT devices.

Model Name	Parameters	Test Accuracy	F1 score (weighted Avg)
Logistic Regression	L1, C=10	93.60%	0.9283
Random Forest	100 trees	97.85%	0.9768
MLP	26,555	98.19%	0.9807
CNN	2,470,269	98.69%	0.9863
LSTM	72,205	98.73%	0.9870
CNN-LSTM	706,133	98.66%	0.9860

TABLE I: Comparing performance of various ML/DL Techniques

We compare the test set accuracy, validation loss, confusion matrix and average F1-score for our seven models across five classes. Based on test set accuracy and confusion matrix, LSTM appears to perform the best; CNN-LSTM, CNN, and LSTM all have similar test set accuracies, however, LSTM has a lot less trainable parameters, requires less training time, and has a smaller model size. CNN has the highest number of parameters. For real-time arrhythmia diagnosis through energy-efficient devices, LSTM can be recommended due to its lightweight design and very short inference time.

Future research on arrhythmia detection may involve experimenting with different Neural Network Architectures, such as Bidirectional-LSTM and Transformers. Bi-LSTM has two separate hidden layers that process the input sequence in forward and backward directions, thus capturing data in both past and future contexts. Transformers have a self-attention mechanism that helps extract important features or capture complex temporal dependencies. Irregularities and certain nuances in ECG signals can be better captured by these advanced architectures.

Furthermore, we could try using different datasets like PTB-XL, CinC 2017, and CPSC 2018, collected with modern standards and have better distribution of super classes and sub classes.This can help improve the accuracy and reliability of our models for detecting arrhythmias. By testing our models on these datasets, we can make sure they work well in real life situations, making remote monitoring and quick diagnosis in healthcare more effective.

Code and Data availability statement

- Code for this project can be accessed on our github repo here
- MIT-BIH is publicly available dataset. It can be found here

VII. Conclusion

In summary, our project aims to detect and classify five classes of arrhythmia in ECG signals: N (normal), F (fusion), Q (unknown), V (ventricular ectopic), and S (supraventricular ectopic). We seek accurate and memory efficient results, favoring simple models while maintaining accuracy. While traditional ML models are our first choice, they often lack accuracy and tend to overfit, so we turned to DL models instead.

Among DL models, MLP, CNN, LSTM, and CNN-LSTM demonstrate effectiveness, yet all of them are not memory-efficient. LSTM stands out as a lightweight model with approximately 98.7% percent accuracy, making it a justifiable choice. To further enhance our project, we aim to explore models such as bi-LSTM, LSTM-transformer networks, and their variants. Additionally, using larger ECG datasets like PTB-XL or Cicn 2017 could yield more generalized and robust results. After validating our model on public datasets, we plan to assess its performance on real time data in controlled environments such as clinics and hospitals before considering deployment for common public use.

REFERENCES

[1] E. J. Benjamin, M. J. Blaha, S. E. Chiuve, M. Cushman, S. R. Das, R. Deo, S. D. De Ferranti, J. Floyd, M. Fornage, C. Gillespie *et al.*, "Heart disease and stroke statistics—2017 update: a report from the american heart association," *circulation*, vol. 135, no. 10, pp. e146–e603, 2017.

[2] K. Mc Namara, H. Alzubaidi, and J. K. Jackson, "Cardiovascular disease as a leading cause of death: how are pharmacists getting involved?" *Integrated pharmacy research and practice*, pp. 1–11, 2019.

[3] P. Madona, R. I. Basti, and M. M. Zain, "Pqrst wave detection on ecg signals," *Gaceta Sanitaria*, vol. 35, pp. S364–S369, 2021, the 3rd International Nursing and Health Sciences Students and Health Care Professionals Conference (INHSP). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0213911121002466

[4] D. Desai and S. Hajouli, "Arrhythmias," *StatPearls*, Jun 5 2023, updated.

[5] T. Li and M. Zhou, "Ecg classification using wavelet packet entropy and random forests," *Entropy*, vol. 18, no. 8, 2016. [Online]. Available: https://www.mdpi.com/1099-4300/18/8/285

[6] S. Pandey and R. Janghel, "Automatic detection of arrhythmia from imbalanced ecg database using cnn model with smote," *Australasian Physical & Engineering Sciences in Medicine*, vol. 42, pp. 1129–1139, 2019.

[7] S. Sowmya and D. Jose, "Contemplate on ecg signals and classification of arrhythmia signals using cnn-lstm deep learning model," *Measurement: Sensors*, vol. 24, p. 100558, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665917422001921

[8] G. Moody and R. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, May-June 2001.

[9] H. Ullah, M. B. Bin Heyat, F. Akhtar, Sumbul, A. Y. Muaad, M. S. Islam, Z. Abbas, T. Pan, M. Gao, Y. Lin, and D. Lai, "An end-to-end cardiac arrhythmia recognition method with an effective densenet model on imbalanced datasets using ecg signal," *Computational Intelligence and Neuroscience*, vol. 2022, p. 9475162, 2022.

[10] PhysioNet, "Physionet annotations," https://archive.physionet.org/physiobank/annotations.shtml, Accessed: 2024-05-03.

[11] B. Murugesan, V. Ravichandran, K. Ram, P. S.P., J. Joseph, S. M. Shankaranarayana, and M. Sivaprakasam, "Ecgnet: Deep network for arrhythmia classification," in *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2018, pp. 1–6.

[12] S. A. Mirsalari, N. Nazari, S. A. Ansarmohammadi, S. Sinaei, M. E. Salehi, and M. Daneshtalab, "Elc-ecg: Efficient lstm cell for ecg classification based on quantized architecture," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5.