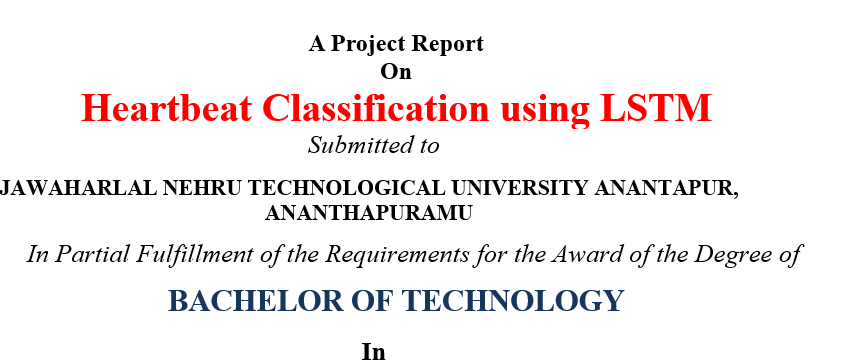
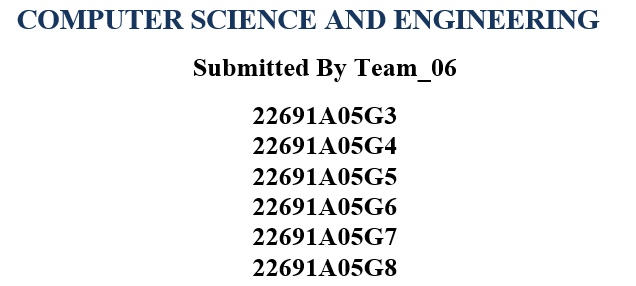
****

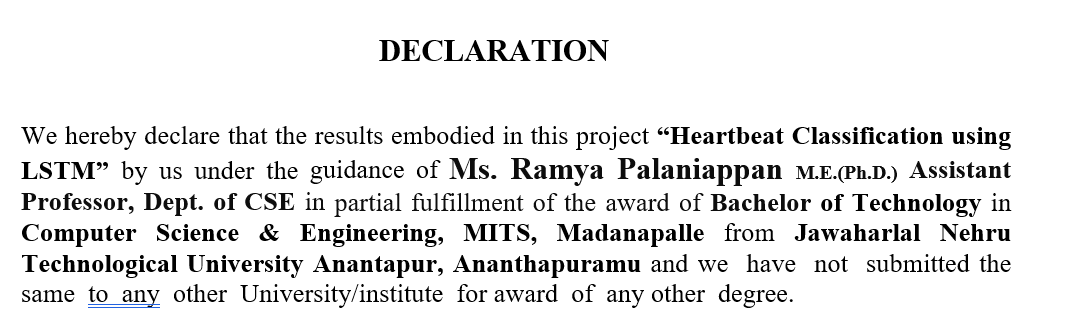
****

### Under the Guidance of

**Ms. Ramya Palaniappan M.E.(Ph.D.)**

### F:\mits LOGO.jpgAssistant Professor

****

****

### Date :

**Place :**

## **PROJECT ASSOCIATES**

**22691A05G3**

**22691A05G4**

**22691A05G5**

**22691A05G6**

**22691A05G7**

**22691A05G8**

I certify that above statement made by the students is correct to the best of our knowledge.

## **ABSTRACT**

Accurate classification of heartbeat signals is essential for the early detection and diagnosis of cardiac anomalies. This study utilizes Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) networks, to classify electrocardiogram (ECG) signals into normal and abnormal categories. Two datasets were used: the PTB dataset for binary classification and the MIT-BIH dataset for multi-class classification into five heartbeat categories. Preprocessing involved balancing datasets, normalizing data, and generating spectrograms to enhance feature extraction.

The proposed model integrates Convolutional Neural Networks (CNNs) and LSTM layers. CNNs were used to extract local features, while LSTMs captured temporal dependencies inherent in ECG signals. The model was trained using binary cross-entropy loss and evaluated on metrics such as accuracy, precision-recall, and area under the curve (AUC). This study includes, Binary Classification (PTB Dataset): 2,093 training and 897 testing observations after preprocessing.

Multi-Class Classification (MIT-BIH Dataset): A combined dataset of 12,142 observations for five heartbeat categories. Results indicate that the integration of CNNs and LSTMs enhances the detection of cardiac anomalies compared to conventional methods. However, performance was affected by challenges like class imbalance, noisy data, and suboptimal hyperparameters. Despite these limitations, the RNN-based approach demonstrated the ability to capture complex patterns in sequential data, emphasizing the potential of LSTMs in medical signal processing.

Keywords: Heartbeat Classification, Electrocardiogram (ECG) Signals. Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) Networks

# **INTRODUCTION**

The early identification and diagnosis of cardiac abnormalities including arrhythmia and myocardial infarction depend heavily on the precise categorisation of heartbeat signals. The electrical activity of the heart is recorded by electrocardiogram (ECG) signals, which are a useful tool for diagnosing various diseases.  
Significant progress has been made in automating these procedures thanks to developments in deep learning. Long Short-Term Memory (LSTM) networks, a subset of recurrent neural networks (RNNs), are particularly good at identifying temporal connections in sequential data, such as ECG signals. When used in conjunction with Convolutional Neural Networks (CNNs) for feature extraction, these techniques offer a strong foundation for accurate heartbeat classification.

This study investigates the use of CNNs and LSTMs to classify heartbeats into five categories for multi-class classification using the MIT-BIH dataset, and into two categories (normal and abnormal) for binary classification using the PTB dataset. To improve feature extraction and model performance, preprocessing techniques such data balance, normalisation, and spectrogram creation were used.

## **2. LITERATURE REVIEW**

Due to its potential to enhance cardiac diagnostics, machine learning and deep learning have been actively researched for the categorisation of heartbeat signals. For processing sequential data, such as electrocardiogram (ECG) signals, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have shown great promise among other methods.

**1. Recurrent Neural Networks (RNNs):** RNNs are a type of neural network that records temporal relationships over time steps by preserving a hidden state. This allows them to analyse sequential input. This qualifies them for uses including voice recognition, ECG signal processing, and time-series forecasting.

**2.GRU** **Gated Recurrent Unit:**

Using the Keras package, the build\_gru\_model method specifies the architecture of a Gated Recurrent Unit (GRU) model. Recurrent neural networks (RNNs) of the GRU type are very good at processing sequential data, such as text or time series.

This function essentially specifies a neural network design based on GRU for categorising sequential input into five groups. It consists of a final dense layer to generate the classification output, two GRU layers, and dropout for regularisation.

**3. Long Short-Term Memory (LSTM) Networks:** LSTM networks are a specific kind of RNN that can recognise long-term relationships in sequences by using gated mechanisms to solve the vanishing gradient problem.

# 3. OBJECTIVE

In order to help in the early identification of cardiac disorders, the main goal of this project, "Heartbeat Classification Using CNN and LSTM Networks for Early Detection of Cardiac Anomalies," is to create a deep learning-based system for the automated categorisation of heartbeat data. Using datasets like PTB and MIT-BIH, the research seeks to categorise ECG signals into many categories for multi-class classification and into normal and pathological categories for binary classification. The suggested framework aims to increase classification accuracy and reliability by combining Long Short-Term Memory (LSTM) networks for modelling temporal relationships and Convolutional Neural Networks (CNNs) for feature extraction. Additionally, the research uses sophisticated preprocessing methods including data balancing, spectrogram creation, and normalisation to solve issues like class imbalance and loud ECG signals.

# **Proposed Method** In order to capitalise on their individual advantages in feature extraction and temporal modelling, the suggested approach for heartbeat classification combines Convolutional Neural Networks (CNNs) ,GRU(Gated Recurrent Unit ) and Long Short-Term Memory (LSTM) networks.

**4.1 Initial Preparation**The ECG data goes through the following preparation procedures to guarantee the model functions properly.  
**Data Balancing:** Using oversampling and under sampling strategies to address the class imbalance in the PTB and MIT-BIH datasets.  
**Normalisation:** To improve training stability, scale the ECG signals to a constant range.  
Spectrogram generation is the process of transforming unprocessed ECG data into frequency-domain representations in order to get fine-grained CNN layer information.  
**4.2 Data Loading and Preprocessing:**

A training set and a testing set, loaded as CSV files, were used in the study. To make sure there were no missing or duplicate values, the datasets were evaluated for integrity and completeness. Analysis of the target variable's class distribution showed inequalities between categories. The frequency of samples by class was displayed in a bar plot, emphasising the structure of the data. For the sake of clarity in the study that followed, numerical class labels were translated to descriptive words.

**Preview of Data:**

**5 rows × 188 columns**

train\_data: A Pandas Data Frame is likely stored in this field. Data may be arranged in rows and columns using a Data Frame, a data structure that resembles a table.

A method named ". head()" is invoked on the train\_data Data Frame. Pandas has a built-in method that, by default, shows the Data Frame’s top five rows. The user wishes to visually examine the top part of the data by executing train\_data. head() in order to gain a preliminary knowledge of its properties, including the names of the columns and the kinds of values it includes.

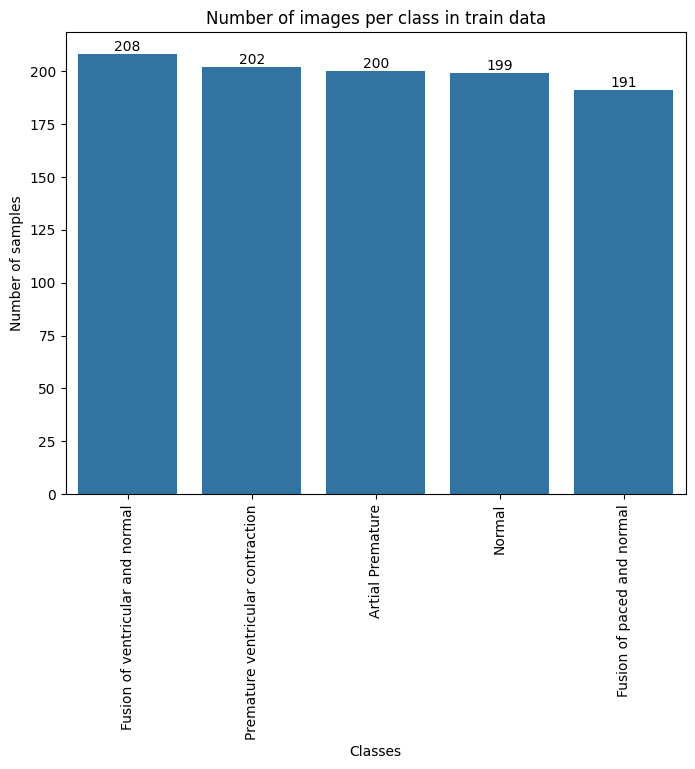
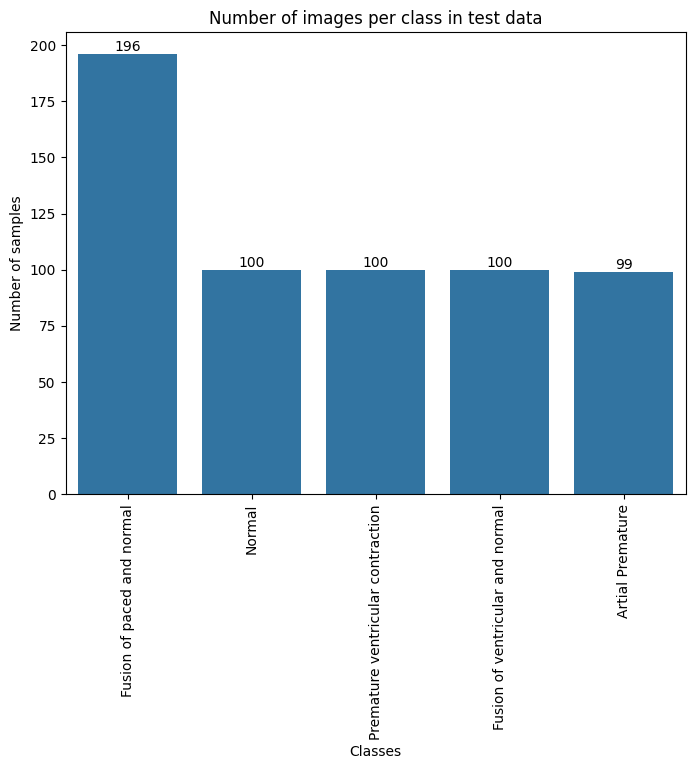
**Label Mapping**:

In essence, this labels dictionary maps numerical labels (0–4) to their descriptions that are legible by humans. In machine learning, particularly in classification issues, this is quite prevalent. Numerical data is frequently used in models, and algorithms typically use numbers rather than text labels. Human interpretation is crucial because we want to be able to comprehend the model's predictions in a way that makes sense to us.

**4.3 Visualization of Class Distribution:**

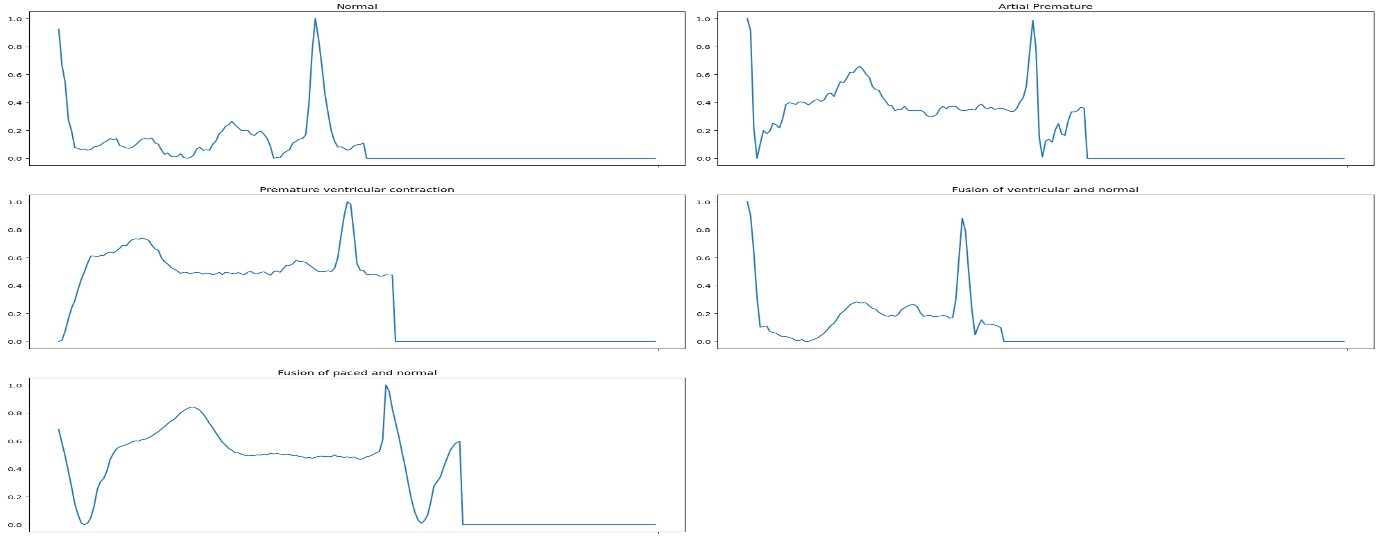
This line of code is in charge of producing and presenting a bar plot that shows how the various classes are distributed throughout the training dataset (train\_data). In essence, it displays the data.

**FIGURE.1**



**4.4 Focus on Insights:**

creating subplots to show the matching heartbeat signals for every class label. This makes it possible to compare the various heart rhythms.

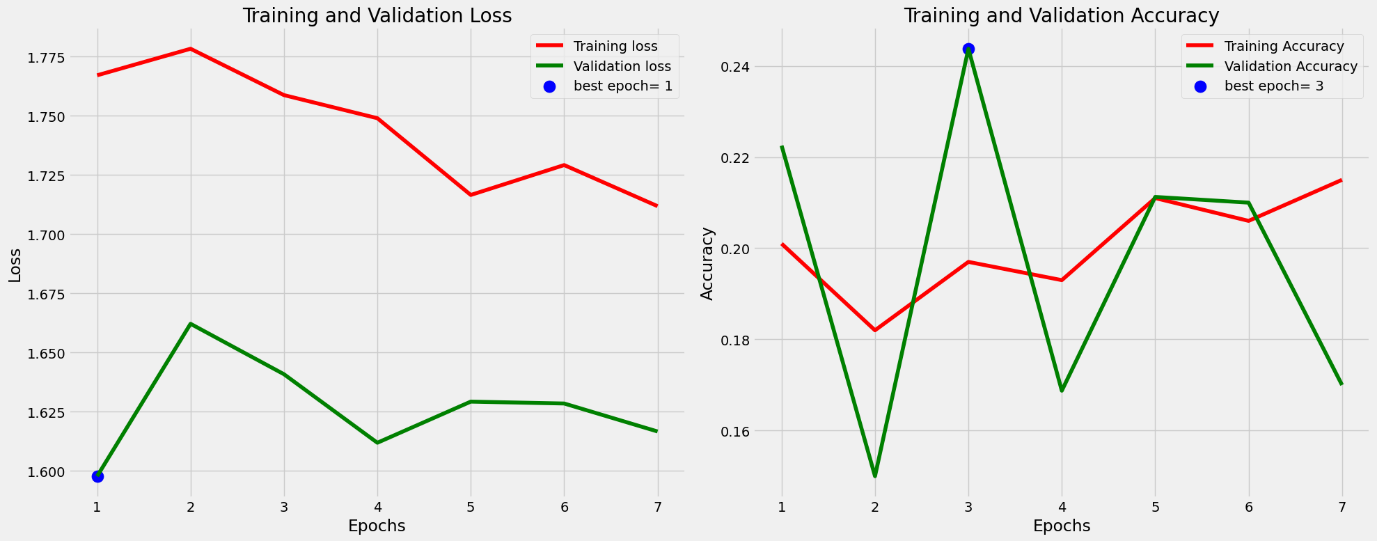


**FIG.2**

**5. Evaluation**

**5.1.RNN Model Training and Evaluation:**

Using pre-prepared data, the RNN model tracks its performance on a validation set and displays the learning process using accuracy and loss charts. This graphic aids in evaluating the model's training progress and spotting any problems like as overfitting.

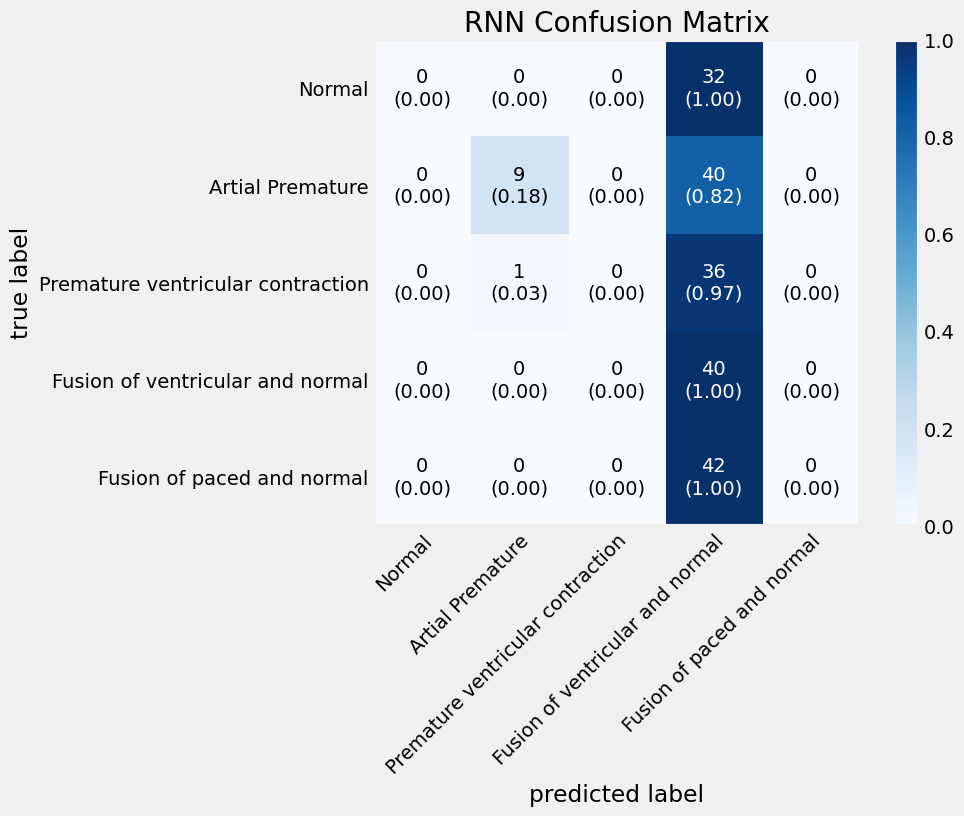


**FIG.3**

In essence, this code visualizes the training history of the model by plotting the recorded metrics over the epochs, helping to understand how the model's performance evolved during training.

**5.2.Confusion Matrix for RNN:**

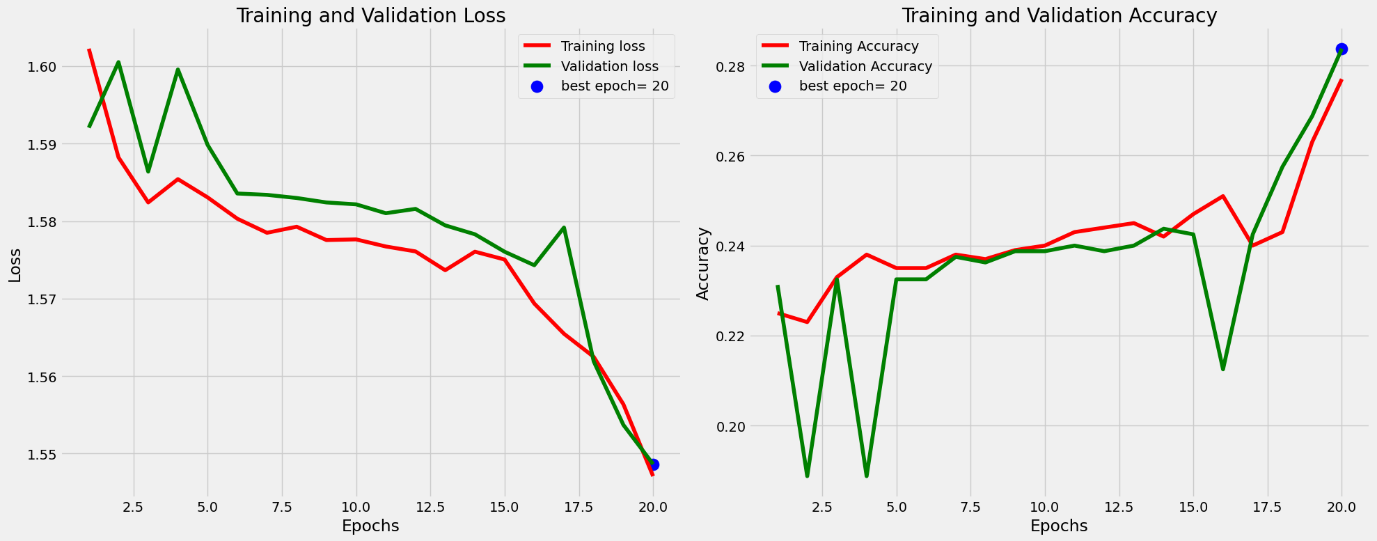
RNN's predictions and the true labels, creates a confusion matrix to summarize the model's performance, and then presents the matrix as a visually informative plot, aiding in understanding where the model is performing well and where it might be making mistake.



**FIG.4**

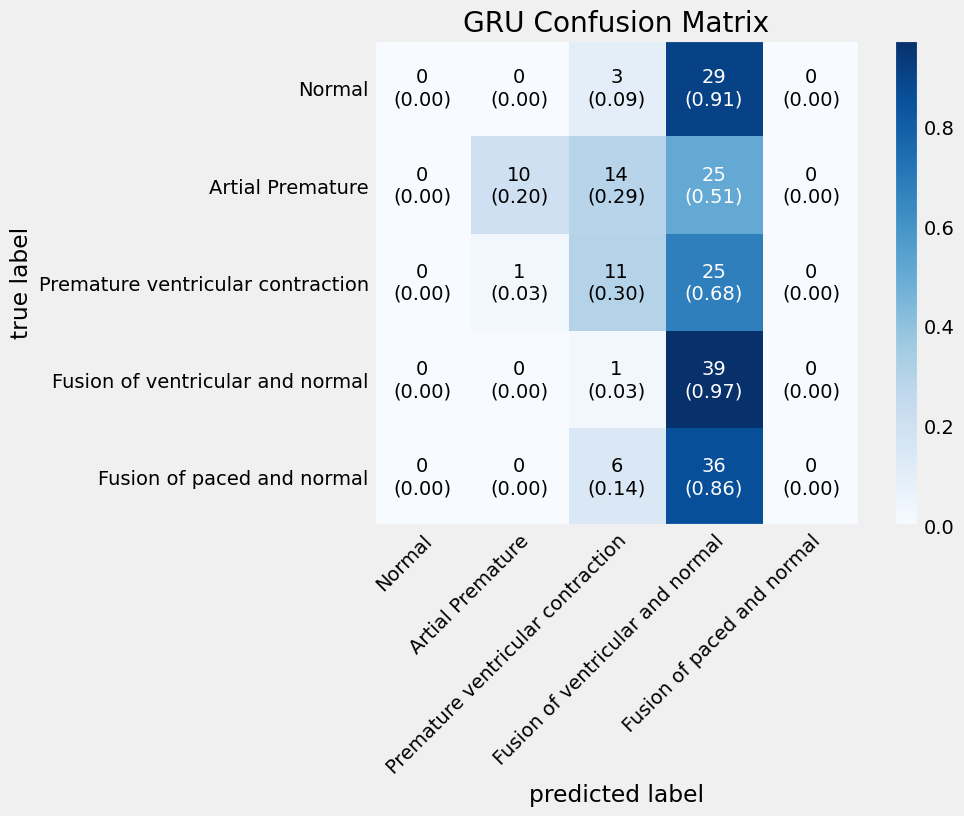
**6.Visualizing GRU Model Training Performance with Learning Curves:**

In this line of code, the learning\_curves\_plot function is essentially told to take the training history data that is stored in gru\_history and plot it to demonstrate how the accuracy and loss of the GRU model changed across epochs on both the training and validation sets. This graphic aids in evaluating the model's functionality and spotting any problems such as underfitting or overfitting.



**Fig 5 Visualization of GRU**

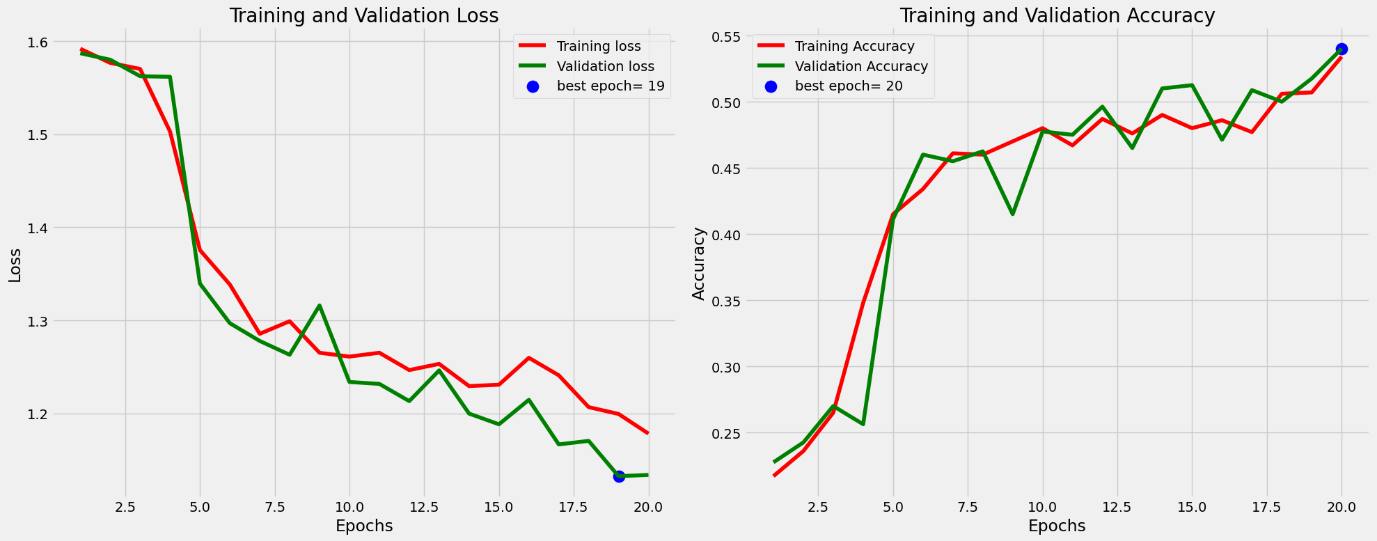
**6.1 Performance with a Confusion Matrix:**



**FIG.6**

**6.2** **Visualizing LSTM Model Training Performance with Learning Curves**

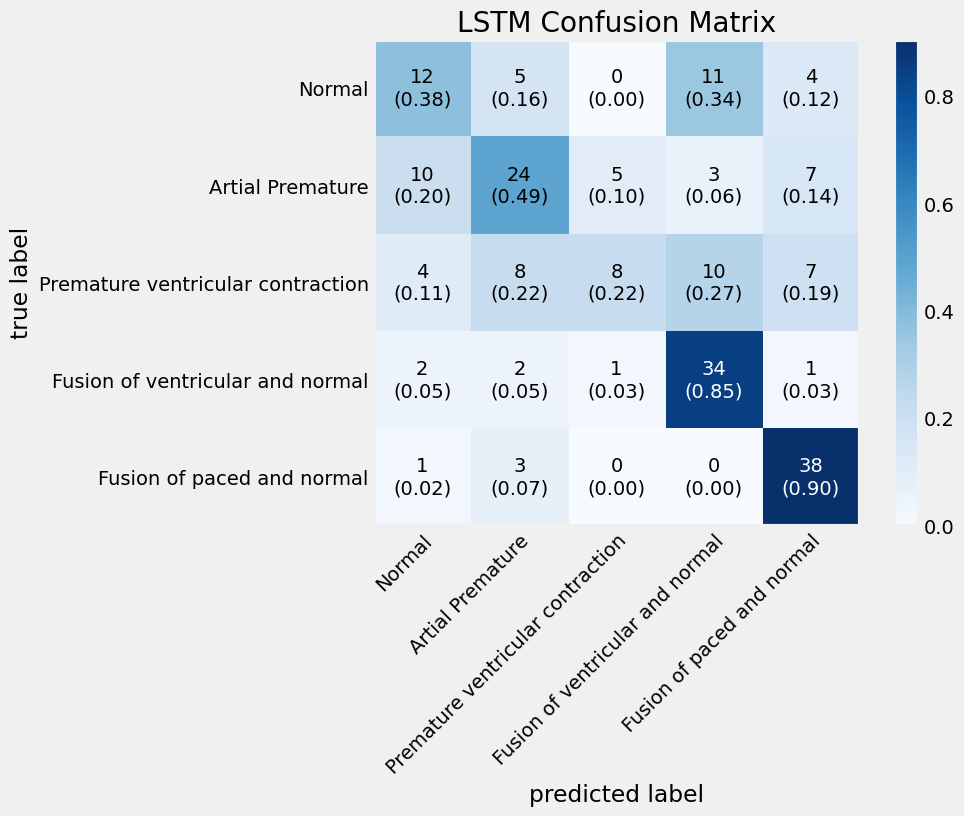
By producing visual depictions of the LSTM model's performance over time, snippets are essential for tracking and evaluating the model's training behaviour. The model is then improved using this data to guarantee that it is efficient for the desired purpose.



**FIG.7**

**6.3 Confusion matrix for LSTM**

An illustration of the LSTM model's confusion matrix. It normalises the data for easier understanding and labels them using the filtered class names. Lastly, a title is shown along with the storyline.



**FIG.8**

# **CONCLUSION** The goal of the study was to use an LSTM model to reliably categorise heartbeat signals. Strong performance metrics were shown by key findings, underscoring the model's potential to aid in the early diagnosis of cardiac problems. Effective training, a well-structured LSTM architecture, and thorough preprocessing were all part of the technique. Data imbalance and other limitations were noted, and recommendations for further research included real-time applications and a variety of datasets. All things considered, this study emphasises how AI might improve heart health monitoring.

# **REFERENCE**

# [**Arrhythmia on ECG Classification using CNN**](https://www.kaggle.com/code/gregoiredc/arrhythmia-on-ecg-classification-using-cnn)

# [**https://ieeexplore.ieee.org/document/10286090**](https://ieeexplore.ieee.org/document/10286090)

[**https://ieeexplore.ieee.org/document/9027740**](https://ieeexplore.ieee.org/document/9027740)

### 

### **team MEMBERs git hub links**

1. <https://github.com/Rafiya1309/heartbeat>
2. [GitHub - pravalika247/Heartbeat-classification](https://github.com/pravalika247/Heartbeat-classification)
3. [GitHub - kpunithkumar63/HeartbeartClassification](https://github.com/kpunithkumar63/HeartbeartClassification)
4. [GitHub - PraveenKumarKotha/heartbeat](https://github.com/PraveenKumarKotha/heartbeat)
5. [GitHub - Rajani0416/Heartbeat-classification-using-LSTM](https://github.com/Rajani0416/Heartbeat-classification-using-LSTM)
6. [GitHub - prava2/heartbeat](https://github.com/prava2/heartbeat)

**Programming Language: Python**