19L620 – INNOVATION PRACTICES

REALTIME ECG MONITORING SYSTEM

AADITYA PRAKAASH J – 22L201
KRISHNAKANTH A – 22L230
MITHUN HARI K – 22L232
MUGUNDAN K – 22L236

Dissertation submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

Branch: ELECTRONICS AND COMMUNICATION ENGINEERING

of Anna University

April 2025

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

REALTIME ECG MONITORING SYSTEM

Bona fide record of work done by

AADITYA PRAKAASH J 22L201
KRISHNAKANTH A 22L230
MITHUN HARI K 22L232
MUGUNDAN K 22L236

Dissertation submitted in partial fulfillment of the requirements for the degree of

Branch: ELECTRONICS AND COMMUNICATION ENGINEERING

of Anna University

April - 2025

……………………………………… …………………………………

Dr. K. Vasanthamani **Dr. V. Krishnaveni**

Faculty guide Head of the Department

Certified that the candidate was examined in the viva-voce examination held on 11/04/2025

……………………….. ………………………………..

(Internal Examiner) (External Examiner)

ACKNOWLEDGEMENT

We would like to sincerely express our heartfelt gratitude to our Principal, **Dr. K. PRAKASAN**, for providing us with the opportunity and necessary facilities to successfully carry out this project.

We also wish to express our sincere thanks to **Dr. V. KRISHNAVENI**, Professor and Head, Department of Electronics and Communication Engineering, for the encouragement and support that she extended towards this project work.

We are grateful to our Programme Coordinator, **Dr. M. SANTHANALAKSHMI**, Associate Professor, Department of ECE, Tutor **Dr. K. THIYAGARAJAN**, Associate Professor, Department of ECE and committee members. **Dr. K. VASANTHAMANI**, Assistant Professor (Sr. Gr) and **Dr. S. MOHANDASS**, Assistant Professor for always being approachable and for the advice and constructive feedback regarding various aspects of the project. Finally, we would like to thank all my department seniors and student colleagues who have helped us in completing this project without any hassles.

CONTENTS

CHAPTER No. **Page**

LIST OF FIGURES

SYNOPSIS

The growing prevalence of cardiovascular diseases underscores the urgent need for accessible and real-time heart health monitoring solutions. Traditional electrocardiogram (ECG) machines, while effective, are often limited to clinical settings, making continuous cardiac health assessment impractical for everyday use. The proposed project, Realtime ECG Monitoring System, addresses this gap by presenting an IoT-enabled, intelligent, and portable ECG monitoring framework capable of providing real-time heart data along with vital health metrics like heart rate (BPM), oxygen saturation (SpO_2), and blood pressure estimation.

The system integrates multiple technologies including microcontroller-based data acquisition, cloud data storage using Firebase, machine learning-based arrhythmia classification, and an intuitive front-end interface developed using Streamlit. A MAX30102 sensor is used for non-invasive heart rate and SpO_2 monitoring, while analog ECG signals are acquired via dedicated biomedical circuitry and transmitted from the ESP32 microcontroller. The acquired ECG signals are sampled, buffered, and uploaded to a real-time Firebase database.

A trained deep learning model, built using TensorFlow and hosted within the Streamlit dashboard, classifies the ECG data into distinct arrhythmia categories such as APC, PVC, LBBB, RBBB, and Normal. The model processes and normalizes ECG waveforms, enabling accurate classification even with limited or noisy data. A sampling rate of 360 Hz ensures clinical-grade resolution of the ECG signal, while automatic detection of lead-off conditions prevents false readings.

The novelty of this system lies in its end-to-end real-time data pipeline—from sensor-based data acquisition to cloud synchronization and on-the-fly ECG classification, all accessible via a simple web interface. The front-end enables users to visualize live ECG plots, review health metrics, and download reports in PDF format, facilitating remote health tracking. To enhance usability and reliability, the system includes a Telegram bot that provides remote notifications and alerts, including hardware connection failures or user-triggered status requests. A voice assistant module is also integrated, capable of detecting keywords like "help" or "SOS" and automatically sending an emergency alert to caregivers via Telegram. Additionally, it provides medication and hydration reminders using text-to-speech output.

This end-to-end system demonstrates a low-cost, portable, and intelligent solution for real-time cardiovascular health monitoring, offering both proactive assistance and automated classification for early diagnosis and rapid intervention.Furthermore, the system automatically manages data storage by clearing outdated entries, optimizing Firebase resource usage.

CHAPTER 1
INTRODUCTION

Cardiovascular diseases (CVDs) are a leading cause of death worldwide,[World Health Organization (WHO) [1]] highlighting the need for early detection and continuous monitoring. ECG is a non-invasive tool for detecting heart abnormalities like arrhythmias. With advancements in low-cost sensors, embedded systems, and AI, real-time health monitoring is now possible outside clinical settings. This project focuses on developing a compact, affordable ECG monitoring system that captures signals using wearable sensors, classifies them using a pretrained deep learning model, and displays real-time results via a web dashboard, leveraging embedded and cloud technologies.

1.1 BACKGROUND

Cardiovascular diseases (CVDs) are a major cause of premature deaths globally, making early detection and continuous monitoring essential. Traditional ECG machines are bulky and clinic-bound, but recent advances in ML, edge AI, and IoT have enabled compact, wearable systems for real-time heart monitoring. This project uses the AD8232 for ECG, MAX30102 for SpO_2 and pulse rate, ESP32 for processing, Firebase for cloud storage, and a CNN + Transformer model for arrhythmia classification—visualized through a Streamlit-based web dashboard.

1.2 PROBLEM STATEMENT

Cardiovascular diseases are the leading cause of death globally, and many of them can be prevented if detected early through continuous ECG monitoring. Existing ECG devices are either too costly or lack the intelligence to classify heart conditions in real-time. This leads to missed opportunities in early diagnosis and treatment of arrhythmias. The main problem addressed in this project is the absence of an affordable and portable real-time system that can acquire ECG signals, process them, classify different types of arrhythmias using deep learning models, and present the data in a user-friendly dashboard. We aim to fill this gap by using low-cost microcontrollers, a machine learning pipeline, and cloud integration.

1.3 OBJECTIVE

The primary objectives of this project include acquiring clean ECG signals using the AD8232 module and processing SpO_2 and pulse rate data via the MAX30102 sensor. The ESP32 is used to interface with the sensors and perform digital signal sampling. Captured data is transmitted to Firebase for temporary cloud storage. An encoder- based Transformer model is developed and trained using Jupyter Lab to classify arrhythmias. The pretrained model is used for real-time ECG signal classification, with results visualized on a Streamlit-based web interface that also plots live ECG waveforms.

1.4 SCOPE

This project enables real-time monitoring and classification of 3-lead ECG signals, with live cloud synchronization via Firebase and basic data clearing features. It is designed for applications in remote health monitoring, personal fitness, and clinical support tools. The system focuses on detecting five arrhythmia classes—Normal, LBBB(Left Bundle Branch Block), RBBB(Right Bundle Branch Block), PVC(Premature Ventricular Contraction), and APC(Atrial Premature contraction)—based on ECG morphology

CHAPTER 2
LITERATURE SURVEY

With the rapid advancements in biomedical signal processing, researchers have explored various methods for classifying ECG signals—ranging from traditional machine learning algorithms to deep learning and hybrid ensemble approaches. These techniques aim to improve the accuracy, robustness, and real-time performance of ECG-based arrhythmia detection systems.

2.1 LITERATURE REVIEW ON ECG CLASSIFICATION USING DEEP LEARNING TECHNIQUES AND ENSEMBLE TECHNIQUES

This chapter presents a detailed review of key research studies that have influenced the design and development of our system. The focus is on models that utilize CNNs, Transformers, XGBoost, and domain adaptation strategies for ECG signal classification. Each study highlights different strengths and limitations, which serve as valuable references for selecting and optimizing our project's methodology.

2.1.1 ECG Classification with Dual Models: XGBoost Voting and Deep Learning with Attention

- Methodology: This work proposes a hybrid model combining XGBoost with a deep learning classifier enhanced by an attention mechanism. It utilizes XGBoost for interpretability and the deep learning module for better feature extraction from ECG signals.
- Results: The dual-model system outperformed both standalone XGBoost and standalone neural networks.
- Limitations: The study lacks real-time deployment experiments and detailed performance benchmarks on embedded hardware.

2.1.2 Electrocardiogram Signals Classification Using Deep Learning Approach Based on LSTM and CNN

- Methodology: Combines LSTM (for temporal features) with CNN (for spatial features) and uses FFT for frequency-domain feature extraction.
- Results: Achieved high classification accuracy on both MIT-BIH and BIDMC ECG datasets.
- Limitations: Does not cover hardware constraints or explain model optimization for microcontrollers.

2.1.3 A Deep Learning Approach for ECG-Based Heartbeat Classification for Arrhythmia Detection

- Methodology: Focuses on ECG-based arrhythmia classification using deep learning architectures trained on the MIT-BIH dataset.
- Results: Demonstrated that deep neural networks can match cardiologist-level performance.
- Limitations: Model complexity may limit use in real-time embedded systems.

2.1.4 Enhancing Electrocardiogram Classification with Multiple Datasets Using Distant Transfer Learning

- Methodology: Uses distant transfer learning and GAN-based domain adaptation to improve generalization across different ECG datasets.
- Results: Classification accuracy improved by ~4% when applying the proposed method across multiple datasets.

- Limitations: Computationally intensive and not optimized for real-time use.

2.1.5 Single and Multi-Lead ECG Heartbeat Classification Using XGBoost on Pre-trained CNN Feature Layer.

- Methodology: Utilizes features extracted from a pre-trained Convolutional Neural Network (CNN), which are then classified using the XGBoost algorithm for both single and multi-lead ECG data.
- Results: Achieved high classification accuracy by effectively combining deep feature extraction and gradient-boosting classification.
- Limitations: The study is a preprint and may lack peer-reviewed validation. It also does not explore model generalizability across different datasets.

2.1.6 Cardiac Arrhythmia Detection and Classification From ECG Signals Using XGBoost Classifier.

- Methodology: Proposes a classification approach using XGBoost, leveraging robust feature extraction techniques to improve the detection and classification of cardiac arrhythmias.
- Results: Showed superior accuracy and performance over traditional classifiers in arrhythmia detection.
- Limitations: Does not address the interpretability of the model and may require significant computational resources.

2.1.7 A Novel Hybrid CNN-Transformer Model for Arrhythmia Detection.

- Methodology: Combines CNN and Transformer models using Stockwell transform without R-peak detection for ECG classification.
- Results: Achieved 97.8% accuracy on Icentia11k and 99.58% on MIT-BIH datasets.
- Limitations: Lacks discussion on deployment for real-time or resource-constrained environments.

2.1.8 DCETEN: A Lightweight ECG Automatic Classification Network.

- Methodology: Uses 1D-CNN with Efficient Channel Attention and Transformer layers, along with model pruning for IoT suitability.
- Results: Achieved 99.84% accuracy and 99.67% F1 score on MIT-BIH dataset.
- Limitations: Tested only on MIT-BIH; real-world generalizability unverified.

2.1.9 Constrained Transformer Network for ECG Classification.

- Methodology: Integrates CNN and Transformer with a link constraint loss to improve performance on imbalanced data.
- Results: Achieved 78.6% F1 score, improving performance on minority classes.
- Limitations: Lower accuracy for rare arrhythmia types due to class imbalance.

2.1.10 ***Unsupervised Transformer-Based Anomaly Detection in ECG Signals*.**

- Methodology: Uses a transformer encoder trained only on normal ECGs; anomalies are detected via reconstruction error.
- Results: Achieved 99% accuracy on ECG5000 and 89.5% on MIT-BIH; outperformed existing models.
- Limitations: Real-world performance and deployment on low-power devices not evaluated.

CHAPTER 3
PROPOSED METHODOLOGY

The proposed system (Fig 3.1) aims to deliver a real-time health monitoring platform that can detect, classify, and visualize ECG signals and vital parameters such as BPM and SpO_2. The system is built using a modular approach involving biomedical sensors, edge computing, cloud storage, and web technologies. This chapter explains the architecture and design decisions taken for both hardware and software components.

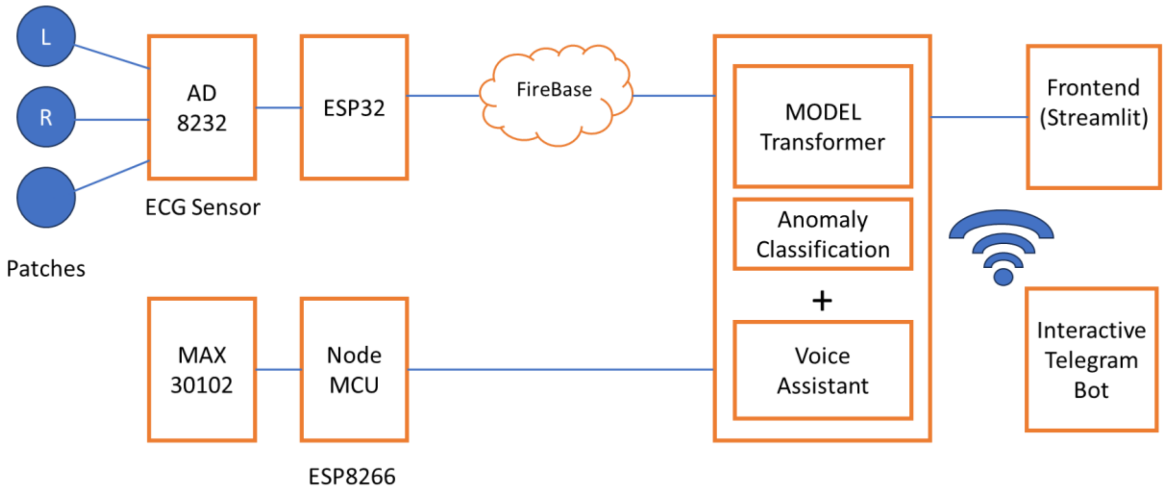


Figure 3.1 General block diagram

3.1 OVERALL SYSTEM ARCHITECTURE

The system comprises three key layers: data acquisition, processing, and visualization. In the acquisition layer, ECG signals are obtained using AD8232 and heart rate/oxygen saturation are measured using the MAX30102 sensor. The ESP32 and NodeMCU ESP8266 handle signal digitization and communication. The processed data is uploaded to Firebase Realtime Database for temporary storage. A web application developed using Streamlit fetches this data, applies the pre-trained Transformer model for classification, and plots real-time ECG signals for end users. This architecture ensures low latency and high availability, making the system suitable for continuous health monitoring and early detection of cardiac anomalies.

3.2 HARDWARE COMPONENTS

3.2.1 ECG Patch and AD8232

The AD8232 analog front-end module is used to acquire ECG signals from electrode patches placed on the subject's chest. It is designed specifically for biopotential measurements and provides built-in signal conditioning including amplification and filtering. It delivers a clean analog ECG waveform, which is fed into the ADC pin of the ESP32. The electrodes used are reusable ECG pads connected to the AD8232 through 3-pin wires (RA, LA, and RL).

3.2.2 ESP32 Microcontroller

The ESP32 microcontroller is utilized for digitizing ECG data from the AD8232 module. With its 12-bit ADC, it samples the analog ECG waveform and temporarily buffers the signal. It ensures accurate real-time conversion and communication with Firebase using Wi-Fi. The ESP32 is also responsible for sampling control, handling lead-off detection, and ensuring signal quality before uploading.

3.2.3 NodeMCU and MAX30102

The MAX30102 is an integrated optical sensor used for measuring SpO_2 (oxygen saturation) and pulse rate via photoplethysmography (PPG). It works in conjunction with the NodeMCU ESP8266, which reads raw IR and red-light data from the sensor and processes it using the open-source SpO_2 algorithm. The NodeMCU then transmits the BPM and SpO_2 readings via serial to the Streamlit dashboard for real-time display

3.3 SOFTWARE STACK

3.3.1 Jupyter Lab (Model Training)

The model training and evaluation were carried out in Jupyter Lab. ECG segments were loaded, preprocessed, and normalized. Custom scripts were used to extract features from the P, Q, R, S, and T waves. The labeled dataset was split into training and testing sets before being passed to a Transformer-based model built with TensorFlow.

3.3.2 TensorFlow (CNN Transformer)

The core of the classification model is based on a CNN-Transformer encoder. The Transformer layers capture long-term dependencies in ECG signals while CNN layers extract local features like wave morphology. TensorFlow was used to build, train, and export the model. After achieving satisfactory accuracy and loss metrics, the trained model was saved in "*.keras*" format for use in the deployment stage.

3.3.3 Streamlit (Web Server)

Streamlit was used to develop a lightweight, interactive web dashboard to monitor patient vitals. The interface connects to Firebase, fetches the latest ECG samples, applies the trained model, and displays the predicted arrhythmia class. It also shows BPM, SpO_2, and calculated BP (SBP/DBP) values

3.4 SIGNAL EXTARCTION AND SIGNAL PROCESSING

3.4.1 ECG SIGNAL CHARACTERISTICS

An ECG (Electrocardiogram) signal reflects the electrical activity of the heart and consists of repetitive cycles of five major waveform components: P, Q, R, S, and T. These components represent different phases of cardiac depolarization and repolarization. The R-wave is typically the most prominent and is used as a reference for beat segmentation. Key parameters such as amplitude, interval, and duration of these waves help identify arrhythmic patterns. ECG signals are typically non-stationary and sensitive to noise, requiring careful processing before classification.

3.4.2 NOISE REMOVAL USING ANALOG FILTERS

To ensure high signal fidelity before digitization, analog filters were employed at the hardware level. A combination of high-pass and low-pass filtering was implemented using the AD8232 module's built-in filtering circuitry. These filters suppress baseline drift, muscle noise, and high-frequency interference from power lines. Additionally, the system uses lead-off detection pins to identify when electrodes are improperly connected, preventing false readings. This preconditioning stage ensures that only physiologically relevant signal content is passed to the ADC.

3.4.3 PQRST FEATURE EXTRACTION

Once the ECG signal is digitized, each beat is isolated using R-peak detection algorithms. From each segmented beat, temporal and amplitude-based features are extracted from the P, Q, R, S, and T points.

These include:
- ***P-wave amplitude and duration***
- ***QRS complex width and R-peak height***
- ***ST segment slope***
- ***T-wave polarity and duration***
- ***RR interval***

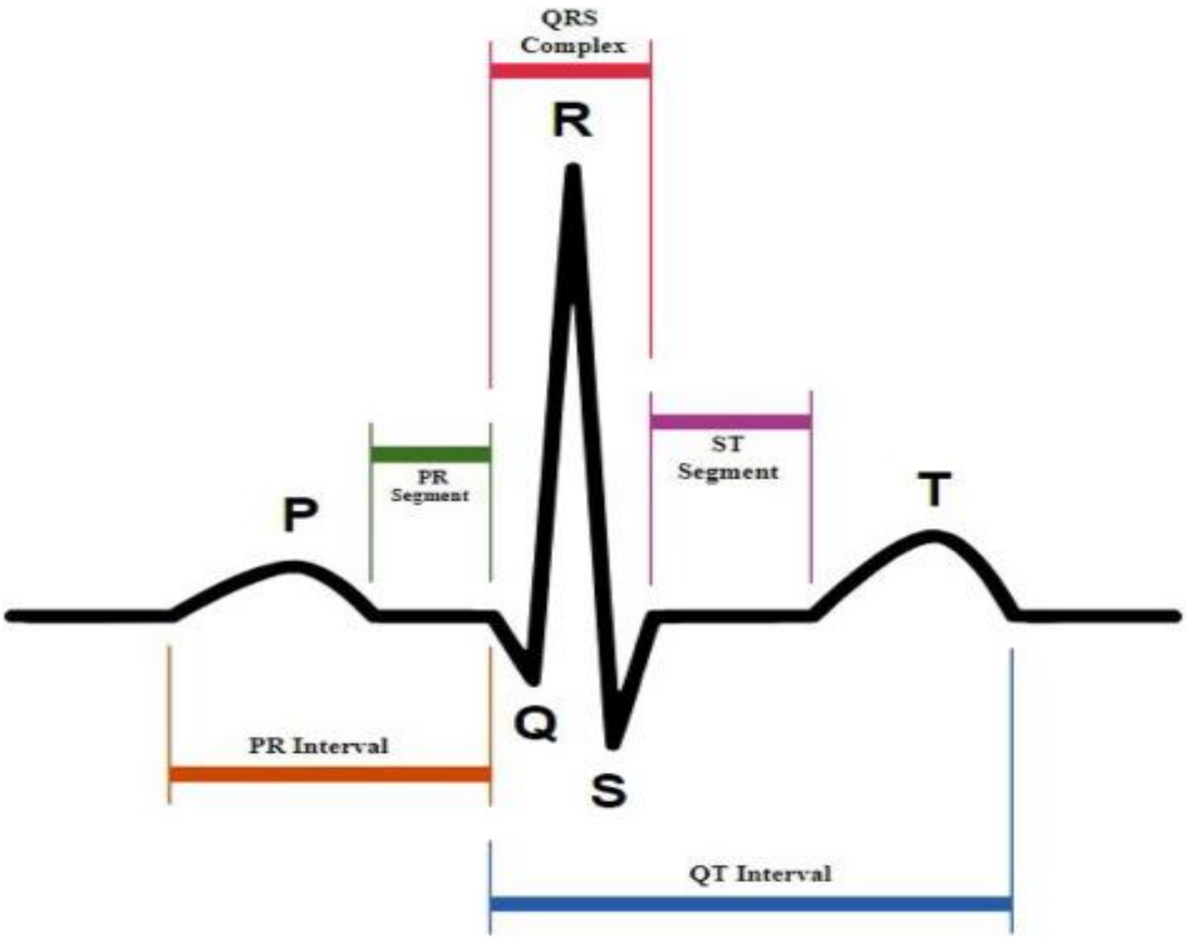


Fig 3.2 ECG signal

These features (as shown in Fig 3.2) are medically relevant and play a crucial role in identifying arrhythmias such as LBBB, RBBB, PVC, and APC. The extracted values are structured as feature vectors, serving as input to the model.

3.4.4 DATASET PREPROCESSING AND NORMALIZATION

Before training, all feature vectors and raw ECG signals are normalized to ensure consistency across samples. Z-score and Min-Max normalization techniques are applied to scale signals to a standard range (typically [-1, 1]). This prevents the model from being biased towards larger amplitude components. Additionally, ECG sequences are zero-padded or truncated to a fixed length of 100-time steps to maintain input dimensionality. This preprocessing improves convergence during model training and reduces variance in prediction.

3.4.5 INTERFACING MAX30102 USING NODEMCU

The MAX30102 sensor is used for measuring heart rate and blood oxygen saturation (SpO_2) through photoplethysmography (PPG). It operates using infrared (IR) and red LEDs, which emit light through the skin and detect changes in blood volume. The output obtained is shown in Fig 3.3. These changes correlate with the heartbeat and oxygen concentration. For this project, the MAX30102 module was connected to a NodeMCU ESP8266 board via the I²C interface. The SDA and SCL lines of the sensor were wired to GPIO4 (D2) and GPIO5 (D1), respectively. The sensor was initialized and configured using the official MAX30105 Arduino library.

A fixed-size buffer was used to store PPG readings, which were processed using the open-source SpO_2 and BPM extraction algorithm provided by Maxim Integrated. The results were printed in serial format (BPM and SpO_2) and read by the Streamlit web interface running on the laptop. This serial link enabled real-time visualization of pulse and oxygen saturation alongside ECG data.

This modular approach ensured that both electrical and optical signals were processed in parallel, providing a more complete picture of the user’s cardiovascular health.

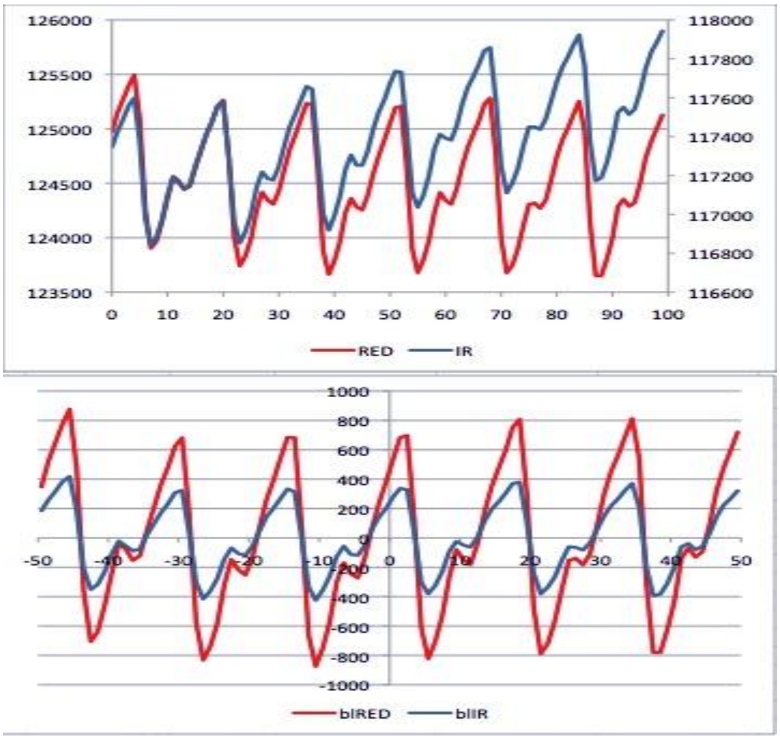


Fig 3.3 Raw and Baseline-Removed RED and IR Signals

3.5 MACHINE LEARNING MODEL

This chapter describes the design and implementation of the deep learning model used for ECG signal classification. The approach leverages a Transformer encoder architecture to effectively learn both local and long-range dependencies in ECG waveforms. The model is trained and validated using a real-world dataset from the WFDB (Waveform Database), which contains annotated ECG records. The chapter also presents evaluation metrics such as accuracy and loss, which were used to assess model performance.

3.5.1 ENCODER TRANSFORMER MODEL ARCHITECTURE

The Transformer encoder architecture is adopted for its capability to model temporal relationships without relying on sequential recurrence. The model accepts ECG signals of fixed length and processes them through an embedding layer followed by stacked multi-head self-attention blocks. These blocks allow the model to learn how different parts of the waveform relate to each other — even across distant time steps. Each attention block is followed by layer normalization and residual connections to maintain gradient stability. After attention, the output passes through feed-forward dense layers with dropout for regularization. The final classification layer uses a SoftMax activation to output the predicted arrhythmia class from five predefined categories: Normal, LBBB, RBBB, PVC, and APC.

The model architecture is designed to be lightweight enough for deployment while preserving sufficient complexity to capture the subtle variations in ECG morphology.

3.5.2 TRAINING AND VALIDATION IN JUPYTER LAB

Model development and experimentation were performed in the Jupyter Lab environment. ECG signals were loaded from the WFDB waveform database using the wfdb Python library. R-peak detection and beat segmentation were applied directly to the waveform, followed by length standardization to ensure uniform input. During training, the dataset was split into training and validation sets using an 80:20 ratio. The model was trained using categorical cross-entropy as the loss function and Adam optimizer with a learning rate of 0.001. Data augmentation techniques such as noise injection and random shifting were applied to increase variability and prevent overfitting. Model checkpoints and early stopping were implemented to optimize training time and preserve the best-performing version of the model.

5.3 ACCURACY AND LOSS METRICS

Model performance was evaluated using two key metrics: ***accuracy*** and ***categorical cross-entropy loss***. Accuracy measures the percentage of correctly classified heartbeats, while the loss quantifies how well the predicted class probabilities match the true labels. Over the training epochs, a consistent decrease in loss and an increase in validation accuracy were observed, indicating stable learning. The final model achieved high classification accuracy on unseen validation data, with strong generalization across arrhythmia classes. Confusion matrix analysis further revealed low misclassification rates between structurally similar beats like LBBB and RBBB.

These results validate the effectiveness of the encoder-based Transformer architecture for ECG signal classification in real-time healthcare applications.

CHAPTER 4
IMPLEMENTATION AND RESULTS

Although the final machine learning model was not deployed on the microcontroller itself, the ESP32 plays a critical role in the system by acquiring, digitizing, and uploading ECG data in real-time. This chapter focuses on how the ESP32 interfaces with the sensor and Firebase, and how the laptop handles the actual model inference by retrieving data from the cloud. This architecture offloads computational load from the microcontroller, enabling the use of powerful Transformer models without sacrificing responsiveness.

4.1 REALTIME ECG ACQUISITION USING ESP32

The ESP32 was used to collect ECG signals from the AD8232 module. It reads analog values using its 12-bit ADC at a sampling rate of approximately 360 Hz. The digitized ECG samples are temporarily stored in a buffer, and once the buffer is filled (e.g., 360 samples per batch), the data is uploaded to Firebase Realtime Database under a user-specific path. Lead-off detection logic is included to ensure reliable signal acquisition and to prevent data corruption due to sensor disconnection. The connections given are shown in Fig 4.1

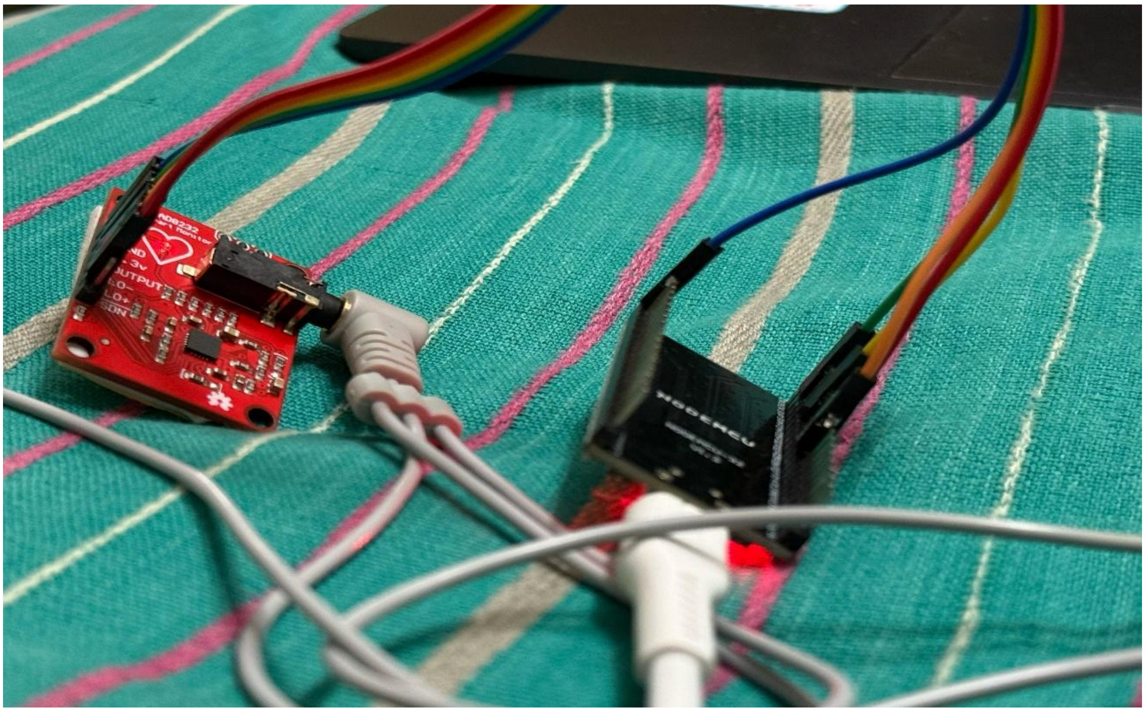


Fig 4.1 Interfacing diagram of ESP32 with AD8232

4.2 FIREBASE COMMUNICATION PROTOCOL

The ESP32 communicates with Firebase via Wi-Fi. Firebase credentials and tokens are configured during device initialization. Once authenticated, the ESP32 pushes ECG batches into the "*/UsersData/<UID>/ecgReadings*" path as JSON arrays. Each batch is timestamped using *millis()* to maintain order. Additionally, the system periodically checks and deletes older data if the sample count exceeds a threshold, thereby preventing database overflow and improving real-time access performance.

4.3 LAPTOP BASED MODEL INTERFACE

In this architecture, the actual inference process is performed on a laptop using Python and Streamlit. The pre-trained Transformer model, saved in ".keras"

format, is loaded at runtime. The Streamlit app continuously polls Firebase for the latest ECG batch, preprocesses the signal (normalization and padding), and feeds it into the model for classification. This approach allows the use of a complex deep learning model without exceeding the resource limits of the ESP32.

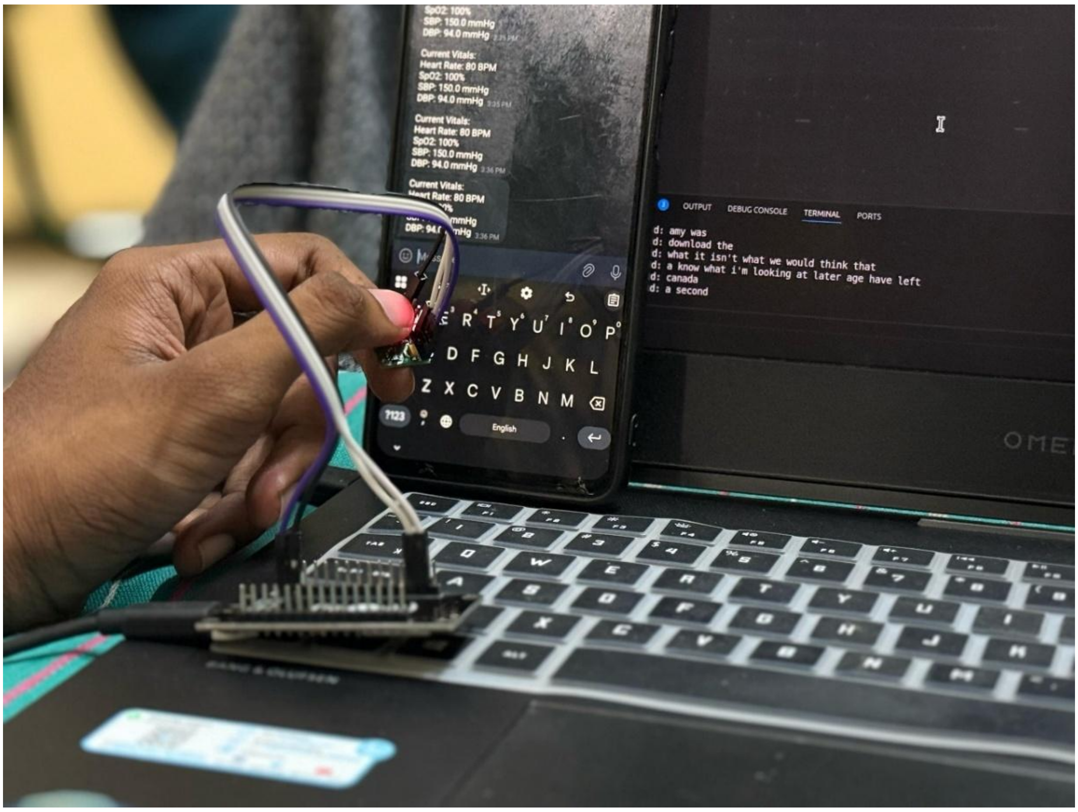


Fig 4.2 Interfacing diagram of MAX30102 with NodeMCU

4.4 INTEGRATION OF TELEGRAM BOT

In this project, a Telegram bot was integrated to enhance the monitoring capabilities of the system. The bot acts as a communication bridge between the user and the real-time health monitoring system, offering two key functionalities:

1. Error Notification:
   Upon failure to establish a connection with either the NodeMCU (ESP32) or the Firebase database, the bot automatically sends a warning message to a predefined Telegram chat. This ensures that the user is immediately alerted to any hardware or connectivity issues.
2. On-Demand Status Updates:
   Users can request the latest vital signs by sending the message "status" to the bot. The bot then fetches the latest values for Heart Rate (BPM), SpO_2, SBP, and DBP from the Firebase database and replies with the current readings.

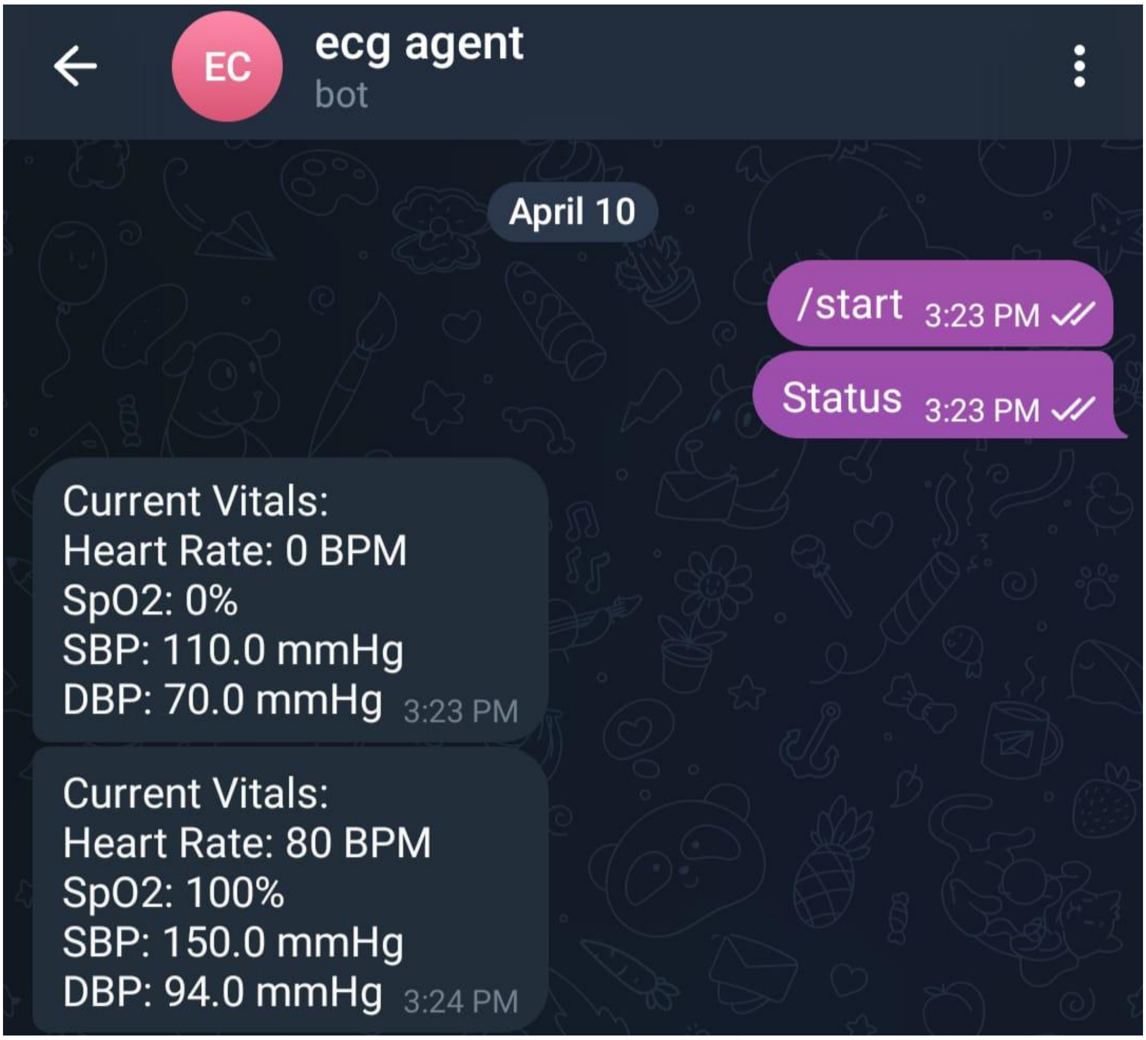


Fig 4.3 Output of Heart rate, SpO2, SBP, DBP as telegram message

The integration was built using Python's requests library for interacting with the Telegram Bot API and Streamlit for the user interface. The bot operates in parallel with ECG signal classification and visualization. This offers a compact and efficient way to notify users without needing them to open the application dashboard.

Key Features:

- Monitors and displays Heart Rate (BPM), SpO_2, SBP, and DBP.
- Fetches ECG signal from Firebase and classifies it using a pre-trained Transformer model.
- Sends alerts in case of:
  - Serial port connection failures.
  - Firebase communication issues.
- Automatically updates the dashboard using streamlit_autorefresh.

Sample Output:

- *"Heart Rate: 78, SpO_2: 97%, SBP: 207 mmHg, DBP: 99 mmHg"*
- *"Warning: Sensor or NodeMCU is not connected properly!"*

This added layer of interaction not only improves accessibility but also enhances the system's reliability by acting as an early-warning mechanism for technical issues.

4.5 VOICE ASSISTANT FOR EMERGENCY ALERTS AND REMAINDERS

To further improve user interaction and safety, a voice assistant module was integrated into the system. The assistant is capable of recognizing keywords such as **"help"** or **"SOS"** from the surrounding environment. When these keywords are detected through voice input, the system automatically triggers an emergency message via the Telegram bot, instantly notifying caretakers or medical staff. The assistant uses the *VOSK offline speech recognition engine* and processes real-time audio using the *PyAudio/sounddevice* module. The keyword detection runs continuously in a parallel thread, ensuring it doesn't interfere with the primary signal classification pipeline. When triggered, it sends the alert using Telegram's Bot API, similar to other error notification messages in the system.

In addition to emergency alerts, the assistant also handles scheduled voice-based reminders using Google Text-to-Speech (`gTTS`) and `pygame` for playback. These reminders are pre-scheduled using the `schedule` library and include:

- Medication reminders at 9 AM, 1 PM, and 9 PM
- Hydration reminders every 30 minutes

These features are especially useful for elderly or disabled patients who may require external prompting for regular health routines.

4.6 RESULT AND ANALYSIS

The real-time ECG monitoring system was successfully designed and implemented with full integration of hardware, cloud, and AI components. The ECG signal acquired using AD8232 and digitized via ESP32 was consistently uploaded to Firebase with minimal latency. The pre-trained Transformer model achieved accurate classification of arrhythmia types and responded reliably when fed with real-time data. The addition of SpO_2 and BPM monitoring using the MAX30102 module expanded the system's capability to track multiple vital signs simultaneously. Live plotting and predictions were visualized using a responsive Streamlit web interface. The system also included Telegram bot notifications, which alerted users to hardware disconnections, connection issues, and live vital signs on request.

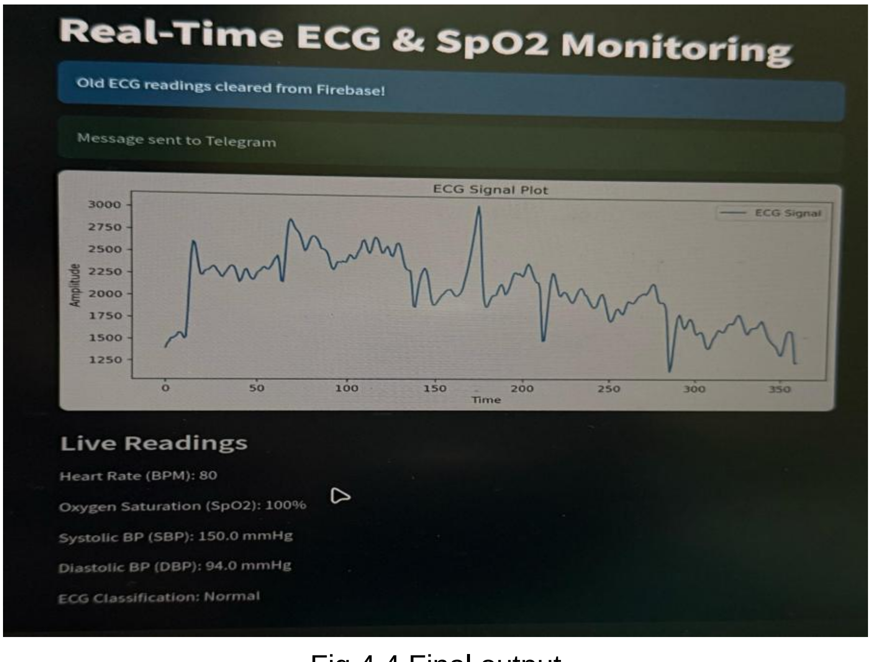


Fig 4.4 Final output

To further improve user safety, a voice assistant feature was integrated, capable of recognizing emergency keywords and sending SOS alerts autonomously. It also provided periodic voice reminders for medication and hydration, improving the system's usability for elderly or assisted-living patients.

CHAPTER 5
CONCLUSION AND FUTURE WORKS

5.1 CONCLUSION

This project journey transformed a simple hardware setup into a smart health monitoring system capable of reacting, learning, and speaking for the user. It wasn't just about measuring signals — it was about making those signals meaningful, accessible, and timely. By combining embedded systems, cloud storage, machine learning, and voice interaction, the final prototype evolved into more than a device — it became a responsive companion for real-time cardiac health support. Each module worked in harmony to offer not only accurate predictions but also convenience, safety, and user comfort.

5.2 FUTURE WORKS

The future works can be done to enhance its capabilities by expanding Firebase storage to allow longer ECG data retention and incorporating additional health metrics such as temperature, respiration rate, and stress levels. The frontend will be upgraded using HTML, CSS, and JavaScript to improve user experience. Additionally, a dedicated application will be developed for doctors to monitor patients' ECG signals in real time, along with features for history tracking and automated report generation at regular intervals.

BIBLIOGRAPHY

1. G. B. Moody and R. G. Mark, “The impact of the MIT-BIH Arrhythmia Database,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.
2. T. Ince, S. Kiranyaz, and M. Gabbouj, “A generic and robust system for automated patient-specific classification of ECG signals,” *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 5, pp. 1415–1426, 2009.
3. O. Yildirim, U. Baloglu, R. Tan, and A. Acharya, “A deep learning model for automated ECG classification,” *Expert Systems with Applications*, vol. 120, pp. 103–111, 2019.
4. H. Rajpurkar et al., “Cardiologist-level arrhythmia detection with convolutional neural networks,” *arXiv preprint arXiv:1707.01836*, 2017.
5. M. Schirrmeister et al., “Deep learning with convolutional neural networks for EEG decoding and visualization,” *Human Brain Mapping, vol. 39, no. 2, pp. 539–554, 2018.*
6. A. Daoud and M. Bayoumi, "Efficient ECG classification using deep learning," *in 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2019, pp. 731-735.*
7. P. de Chazal et al., "Automatic classification of heartbeats using ECG morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering, vol. 51, no. 7, pp. 1196-1206, 2004.*
8. S. Hong et al., "ECG arrhythmia classification using a 2-D convolutional neural network," *IEEE Access, vol. 7, pp. 92871-92880, 2019.*
9. J. Yao and R. W. Smith, "ECG signal classification using deep learning and a stacked LSTM model," *in 2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS), 2021, pp. 568-573.*
10. S. M. A. Basha, K. J. Kiran, and B. Rajasekaran, "Deep learning-based approach for ECG classification," *in 2020 IEEE International Conference on Advances in Computing, Communication, and Control (ICAC3), 2020.*
11. M. S. Rahman et al. , "A Novel Hybrid CNN-Transformer Model for Arrhythmia Detection," Scientific Reports , vol. 14, no. 1, 2024
12. K. Liu, X. Zhang et al. , "DCETEN: A Lightweight ECG Automatic Classification Network Based on Transformer," Computers in Biology and Medicine , vol. 169, 2024.
13. N. V. K. Parcha et al. , "Constrained Transformer Network for ECG Signal Processing and Arrhythmia Classification," BMC Medical Informatics and Decision Making , vol. 21, no. 1, 2021.
14. K. Palanisamy et al. , "Unsupervised Transformer-Based Anomaly Detection in ECG Signals," Algorithms , vol. 16, no. 3, pp. 152, 2023.
15. M. Abdelbaky et al. , "A Tiny Transformer for Low-Power Arrhythmia Classification on Microcontrollers," arXiv preprint , arXiv:2402.10748, 2024.
16. L. Moroni et al. , "Reliable ECG Anomaly Detection on Edge Devices for Internet of Medical Things Applications," Sensors , vol. 25, no. 8, pp. 2496, 2023.
17. How2Electronics, "IoT-Based ECG Monitoring with AD8232 ECG Sensor & ESP32," 2023.
18. A. Aby, "IoT-Based Low-Cost ECG and Heart Monitoring System with ESP32," GitHub , 2023.

19. M. Faust, U. Ray, and E. Eskreis-Winkler, "Automated detection of arrhythmias using machine learning on ECG signals: A review," *Journal of Electrocardiology*, vol. 69, pp. 33–39, 2021.
20. P. Zarei, A. Aslani, and M. Khandan, "Edge computing for real-time ECG signal monitoring and classification: Challenges and solutions," *IEEE Access*, vol. 11, pp. 12456–12467, 2023.
21. A. Kachuee, M. M. Fazeli, and M. Sarrafzadeh, "ECG heartbeat classification: A deep transferable representation," in *IEEE International Conference on Healthcare Informatics (ICHI)*, 2018, pp. 443–444.
22. Y. Xie, L. Jin, and Y. Zhang, "Real-time ECG signal processing and classification based on a hybrid deep neural network," *Biomedical Signal Processing and Control*, vol. 68, p. 102743, 2021.
23. A. S. Wahid, A. Afzal, and N. Ahmad, "IoT-based intelligent system for ECG monitoring and arrhythmia detection using lightweight deep learning models," *Sensors*, vol. 23, no. 4, p. 1674, 2023.
24. T. Ma, Y. Li, and Z. Qian, "Transformer-based ECG classification with attention-guided heartbeats," in *Proceedings of the IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, 2022, pp. 275–278.
25. R. Chauhan and P. Vig, "Anomaly detection in ECG using lightweight CNN on ESP32: An IoT-based implementation," *Internet of Things*, vol. 19, p. 100595, 2022.

APPENDIX-A

DATASET USED FOR TRAINING THE MODEL:

For training and validating the model the MIT-BIH arrythmia dataset was used. The MIT-BIH Arrhythmia Dataset is a widely used benchmark in ECG and arrhythmia detection research. It was created by the Beth Israel Hospital in collaboration with MIT.

Key Features:

- Records: 48 half-hour ECG recordings from 47 subjects
- Sampling Rate: 360 Hz
- Leads: Primarily Lead II and V1
- Annotations: Each beat is manually labeled with arrhythmia classes (e.g., normal, premature ventricular contraction, atrial premature beat)
- Classes: Includes both normal and abnormal heartbeats (e.g., PVC, LBBB, RBBB)

SPECIFICATIONS OF HARDWARE USED:

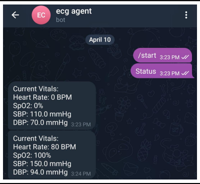
- **ESP32 :**
  - The **ESP32** is a low-cost, dual-core microcontroller with Wi-Fi and Bluetooth v4.2. It runs up to 240 MHz, has 520 KB SRAM, and supports GPIO, ADC, DAC, SPI, I2C, and UART. Ideal for IoT and automation, it supports Arduino, ESP-IDF, and FreeRTOS for flexible development.
- **NodeMCU:**
  - The **NodeMCU** is a low-cost, open-source development board based on the ESP8266 Wi-Fi SoC. It features a 32-bit processor, built-in Wi-Fi, ADC, PWM, I2C, SPI, and GPIOs. Supporting Lua and Arduino IDE, it’s ideal for IoT, home automation, and rapid prototyping with USB-based programming.
- **AD8232 SENSOR**:
  - The **AD8232** is a compact, low-power ECG sensor for monitoring heart activity. It amplifies and filters biopotential signals, providing clean analog ECG output. Operating at 3.3V–5V, it's ideal for wearable and fitness devices, and easily interfaces with microcontrollers like Arduino and ESP32 for real-time heart monitoring.
- **MAX30102:**
  - The MAX30102 is an integrated pulse oximeter and heart-rate sensor. It combines red and IR LEDs, a photodetector, and analog/digital signal processing in one module. It communicates via I2C and is widely used for heart rate and SpO_2 monitoring.

APPENDIX-B

Poster:

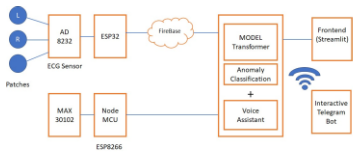
	19L620 Innovation Practices - Project Expo	Date: 11/04/2025 Venue: PSGCT	

REALTIME ECG SIGNAL CLASSIFICATION



Project Objectives:

The primary objectives of this project include acquiring clean ECG signals using the AD8232 module and processing SpO_2 and pulse rate data via the MAX30102 sensor. Captured data is transmitted to Firebase for temporary cloud storage. An encoder-based Transformer model is developed and trained using Jupyter Lab to classify arrhythmias. The pretrained model is used for real-time ECG signal classification, with results visualized on a Streamlit-based web interface that also plots live ECG waveforms.



TEAM MEMBERS:

- *AADITYA PRAKASH J (22L201)*
- *KRISHNAKANTH A (22L230)*
- *MITHUN HARI K (22L232)*
- *MUGUNDAN K(22L236)*