IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁂ Peer-reviewed & Refereed journal ⁂ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15470

IoT and Machine Learning Based Smart Health Monitoring System

Sidagam Geethika[1], P. Bose Babu[2]

B.Tech Student, Department of Electronics and Communication Engineering,

Andhra Loyola Institute of Engineering and Technology (ALIET), Vijayawada, India[1]

Assistant Professor, Department of Electronics and Communication Engineering,

Andhra Loyola Institute of Engineering and Technology (ALIET), Vijayawada, India[2]

Abstract: The rapid advancement of Internet of Things (IoT) and Machine Learning (ML) technologies has opened new avenues for developing intelligent and cost-effective healthcare monitoring systems. This paper presents the design and implementation of an IoT and Machine Learning based Smart Health Monitoring System capable of continuously monitoring multiple physiological and environmental parameters in real time. The proposed system integrates the MAX30102 sensor for heart rate and blood oxygen saturation (SpO_2) measurement, the DS18B20 sensor for body temperature monitoring, and the DHT11 sensor for ambient temperature and humidity sensing. An ESP32 microcontroller acts as the central processing unit, collecting sensor data and transmitting it to the ThingSpeak cloud platform via Wi-Fi for remote access and visualization. A 16×2 I2C LCD display provides immediate local readout of health parameters. A Random Forest machine learning algorithm, deployed through a Streamlit-based Python application, classifies the patient's health condition as Normal or Critical. Additionally, a GSM module (SIM900A) sends automated SMS alerts and voice calls to caregivers when critical conditions are detected. Experimental results demonstrate that the system achieves reliable real-time monitoring, accurate ML-based health classification, and effective emergency notification. The system is low cost, portable, and scalable, making it highly suitable for home healthcare, elderly monitoring, and remote medical applications.

Keywords: Internet of Things (IoT), Machine Learning, Health Monitoring System, ESP32, MAX30102, ThingSpeak, Random Forest, Remote Patient Monitoring, GSM Alerts.

I. INTRODUCTION

The increasing prevalence of chronic diseases and sudden cardiac events has underscored the critical need for continuous, real-time patient health monitoring systems. Traditional healthcare relies heavily on in-hospital observation, which is not always feasible for elderly patients, individuals with chronic conditions, or those residing in remote regions. The convergence of Internet of Things (IoT) and Machine Learning (ML) technologies offers a transformative solution by enabling intelligent, remote, and autonomous health monitoring outside clinical settings.

The Internet of Things refers to a network of interconnected physical devices embedded with sensors, software, and communication capabilities that collect and exchange data over the internet [1]. IoT-based health monitoring systems leverage low-cost biomedical sensors to capture vital physiological parameters such as heart rate, blood oxygen saturation (SpO_2), and body temperature. These parameters are then transmitted to cloud platforms for storage, visualization, and remote access by healthcare professionals. With the integration of Machine Learning algorithms, such systems gain the ability to intelligently analyze health data, identify abnormal patterns, and predict potential health risks, thereby enabling early medical intervention.

Heart diseases remain one of the leading causes of mortality worldwide. Conditions such as coronary artery disease, arrhythmias, and heart failure require continuous monitoring for effective management. A smart health monitoring system that integrates multiple sensors with ML-based analysis can assist doctors and caregivers in tracking patient health trends and responding promptly to emergencies.

The system proposed in this paper integrates the ESP32 microcontroller, the MAX30102 pulse oximetry sensor, the DS18B20 body temperature sensor, and the DHT11 ambient sensor to form a comprehensive health monitoring platform. The collected data is visualized locally on an LCD display and transmitted to the ThingSpeak cloud for remote monitoring. A Random Forest ML classifier, implemented in Python using the Streamlit framework, provides automated health condition classification. In critical cases, the integrated GSM module dispatches SMS alerts and voice calls to designated emergency contacts, ensuring timely medical response.

This paper is organized as follows: Section II presents a literature survey of related work. Section III describes the system scope and design objectives. Section IV details the hardware and software methodology. Section V presents the experimental results and discussion. Section VI provides the conclusion, and Section VII outlines future work.

II. LITERATURE SURVEY

A growing body of research has explored the integration of IoT and machine learning technologies in healthcare monitoring systems. The following summarizes key relevant works.

Gupta et al. [1] developed an IoT-based smart healthcare monitoring system utilizing wearable sensors to continuously monitor vital signs, demonstrating that IoT-enabled wearables can effectively support remote patient monitoring and early detection of health anomalies. Khan et al. [2] proposed a real-time IoT system using cloud computing for patient health monitoring, illustrating how cloud-based architectures facilitate data storage, real-time alerts, and accessibility for healthcare providers.

Patel and Shah [3] combined IoT technology with machine learning algorithms to build an intelligent healthcare monitoring system, proving that ML-based analysis significantly enhances disease detection accuracy compared to fixed threshold-based methods. Chatterjee et al. [4] introduced an AI-enabled IoT framework for remote healthcare monitoring published in IEEE Access, emphasizing that artificial intelligence can process large health datasets to provide accurate early diagnoses. Ramesh and Prakash [5] explored IoT healthcare monitoring with edge computing to reduce communication latency and enable faster abnormality detection, extending the applicability of IoT health systems to resource-constrained environments.

The present work advances beyond these contributions by integrating multiple sensor modalities (pulse oximetry, body temperature, ambient monitoring) with a Random Forest ML classifier and GSM-based emergency alerting into a unified, low-cost, portable system.

III. METHODOLOGY

A. System Architecture and Block Diagram

The proposed smart health monitoring system is organized into five functional layers: (1) Sensor Layer, (2) Processing Layer, (3) Local Display Layer, (4) Cloud Communication Layer, and (5) Machine Learning and Alert Layer. The ESP32 microcontroller acts as the hub integrating all layers. Fig. 1 illustrates the overall block diagram of the system.

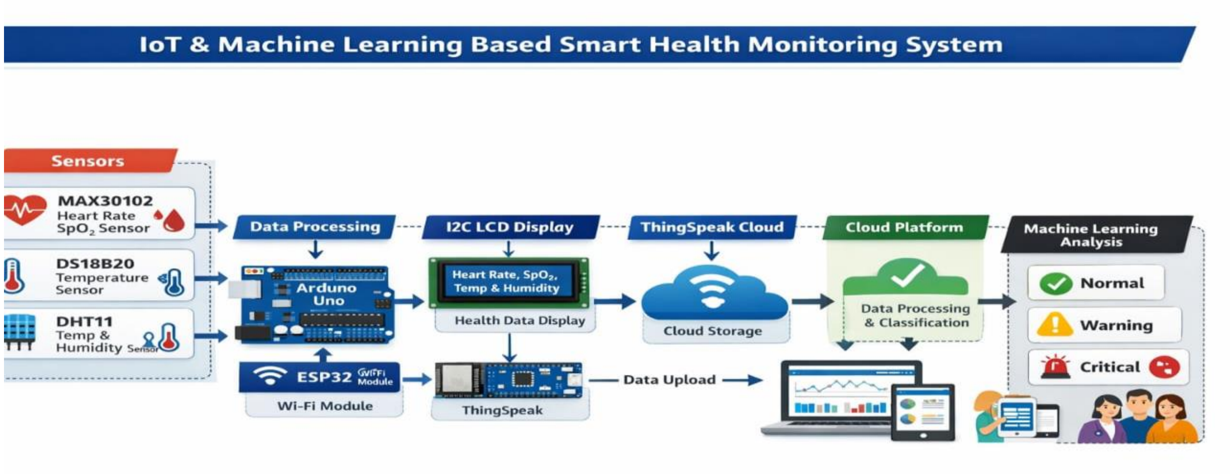


Fig. 1: Block Diagram of the IoT and ML Based Smart Health Monitoring System

The MAX30102, DS18B20, and DHT11 sensors connect to the ESP32 via I2C and GPIO interfaces. The ESP32 processes the sensor data, displays it on the I2C LCD, and transmits it to the ThingSpeak cloud using Wi-Fi. The Python-based Streamlit application fetches data from ThingSpeak via an API and runs the trained Random Forest model to classify the patient's health status. In critical conditions, the GSM module (SIM900A) sends SMS alerts and makes voice calls.

B. Hardware Components and Pin Specifications

The hardware platform consists of the following components:

1) ESP32 Microcontroller: The ESP32 is a dual-core, 240 MHz microcontroller with built-in Wi-Fi and Bluetooth, developed by Espressif Systems. It serves as the central processing unit of the system, interfacing with all sensors and communication modules. Key pin assignments include: 3.3V for sensor supply, GND for common ground, GPIO21 (SDA) and GPIO22 (SCL) for I2C communication, GPIO4 for the DHT11 data line, GPIO5 for the DS18B20 one-wire data line, GPIO16 (RX) connected to GSM TX, and GPIO17 (TX) connected to GSM RX.

DOI: 10.17148/IJARCCE.2026.15470

2) MAX30102 (Pulse Oximetry and Heart Rate Sensor): The MAX30102 is an integrated biosensor module that measures heart rate and blood oxygen saturation (SpO_2) using infrared and red LED photoplethysmography. It communicates with the ESP32 via the I2C protocol (SDA: GPIO21, SCL: GPIO22) and operates at 3.3 V.

3) DS18B20 (Body Temperature Sensor): The DS18B20 is a digital one-wire temperature sensor with a measurement range of -55°C to +125°C and ±0.5°C accuracy, suitable for body temperature monitoring. It connects to GPIO5 and requires a 4.7 kΩ pull-up resistor between the data line and VCC (3.3 V).

4) DHT11 (Ambient Temperature and Humidity Sensor): The DHT11 monitors room temperature (0°C to 50°C) and relative humidity (20% to 90% RH). It uses a single-wire digital communication protocol connected to GPIO4 and is powered at 3.3 V. Ambient monitoring complements physiological measurements and enhances ML-based health classification.

5) 16×2 I2C LCD Display: The LCD display (with an I2C backpack module at address 0x27) provides real-time local readout of health parameters. Using I2C reduces the wiring to four pins (VCC: 5 V, GND, SDA: GPIO21, SCL: GPIO22), sharing the I2C bus with the MAX30102 sensor.

6) SIM900A GSM/GPRS Module: The SIM900A GSM module enables cellular network communication for SMS alerts and voice calls in emergency conditions. It operates via UART serial communication (TX to GPIO17, RX to GPIO16) using AT commands. The module requires a stable external power supply capable of delivering up to 2 A.

TABLE I
Sensor Measurement Ranges and Critical Thresholds

Sensor	Parameter	Measurement Range	Normal Range	Critical Condition
MAX30102	Heart Rate	30 – 220 bpm	60 – 100 bpm	< 50 or > 110 bpm
MAX30102	SpO_2	0 – 100 %	95 – 100 %	< 90 %
DS18B20	Body Temperature	-55°C to 125°C	36 – 37.5°C	> 38.5°C or < 35°C
DHT11	Room Temperature	0°C – 50°C	20 – 30°C	< 15°C or > 35°C
DHT11	Humidity	20% – 90% RH	40% – 70% RH	< 30% or > 80% RH

C. Circuit Diagram

The circuit integrates all hardware components around the ESP32 microcontroller. The power supply provides a regulated 5 V DC output distributed to the LCD display, DHT11 sensor, and ESP32 board, while the ESP32 internally generates 3.3 V for the MAX30102 and DS18B20 sensors. I2C lines (SDA: GPIO21, SCL: GPIO22) are shared between the MAX30102 and LCD display, with each device assigned a unique I2C address. The DS18B20 communicates on a single one-wire bus (GPIO5) with a 4.7kΩ pull-up resistor. The SIM900A GSM module connects via UART and requires dedicated ground and power connections. Fig. 2 shows the complete circuit diagram.

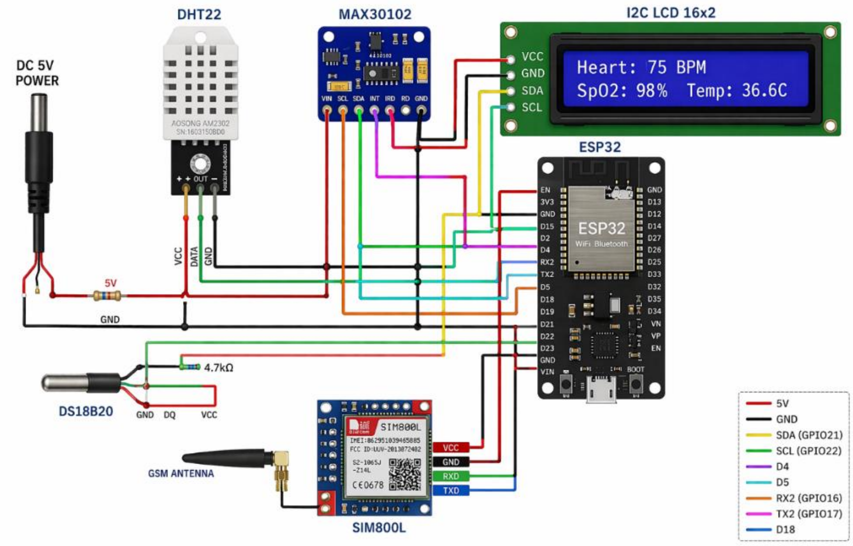


Fig. 2: Circuit Diagram of the Smart Health Monitoring System

D. Software Design and Implementation

The software architecture of the system comprises two principal components: the embedded firmware running on the ESP32 and the Python-based machine learning application.

Embedded Firmware (Arduino IDE): The ESP32 is programmed using the Arduino IDE with libraries including WiFi HTTPClient.h, Wire.h, LiquidCrystal_I2C.h, DallasTemperature.h, DHT.h, MAX30105.h, and spo2_algorithm.h. On startup, the firmware initializes all sensors, establishes a Wi-Fi connection, and initializes the LCD and GSM modules. In the main loop, the system continuously reads sensor values, displays them on the LCD, and periodically transmits the data to ThingSpeak via an HTTP POST request. When abnormal health thresholds are exceeded, the firmware triggers the GSM module via AT commands (AT+CMGF=1 for SMS text mode, AT+CMGS for message dispatch) to send emergency alerts.

Machine Learning Application (Python / Streamlit): A Python application built with the Streamlit framework provides an interactive web-based user interface for health condition classification. The application operates in two modes: (i) Auto Mode, in which real-time sensor data is fetched from the ThingSpeak API (JSON format) using the Python requests library, and (ii) Manual Mode, in which users enter health parameter values directly through input widgets. The application loads a pre-trained Random Forest classifier (scikit-learn) to predict the patient's health condition. The Random Forest algorithm consists of multiple decision trees that independently analyze the input data, with the final prediction determined by majority voting, providing high accuracy and robustness against overfitting. The input feature vector includes heart rate, SpO_2, body temperature, and room temperature. A classification output of 1 indicates a Normal condition, while 0 indicates a Critical condition. The interface also generates an AI-based textual explanation summarizing the predicted health status.

E. System Workflow

Fig. 3 presents the system flowchart. The system begins with initialization of Wi-Fi, sensors, LCD, and GSM modules. The user selects Auto Mode (live IoT data from ThingSpeak) or Manual Mode (direct input). The data is forwarded to the ML model, which predicts the health condition. If the condition is Normal, the system displays a 'NORMAL' status, generates an AI explanation, and updates the ThingSpeak dashboard. If the condition is Critical, the system displays a 'CRITICAL' alert, generates an AI explanation, and immediately triggers the GSM module to send an SMS alert and initiate a voice call to the caregiver or medical professional. The monitoring loop then repeats continuously.

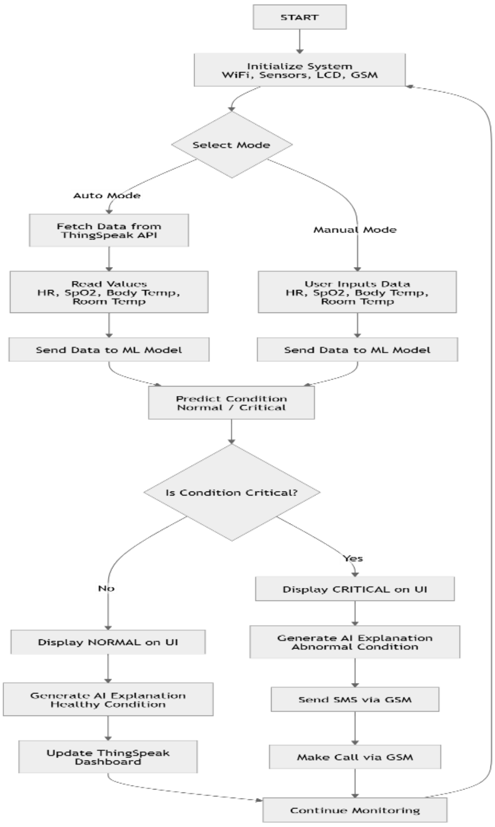


Fig. 3: System Flowchart

IV. RESULTS AND DISCUSSION

A. LCD-Based Real-Time Display

The system was tested by placing the MAX30102 sensor on the fingertip, the DS18B20 probe in contact with the skin, and the DHT11 sensor in the monitoring environment. Real-time health parameters including heart rate, SpO_2, body temperature, and room temperature were successfully displayed on the 16×2 I2C LCD. The LCD output was updated at consistent intervals, confirming stable and continuous local monitoring. Fig. 4 shows the hardware setup and the corresponding LCD output values during testing.

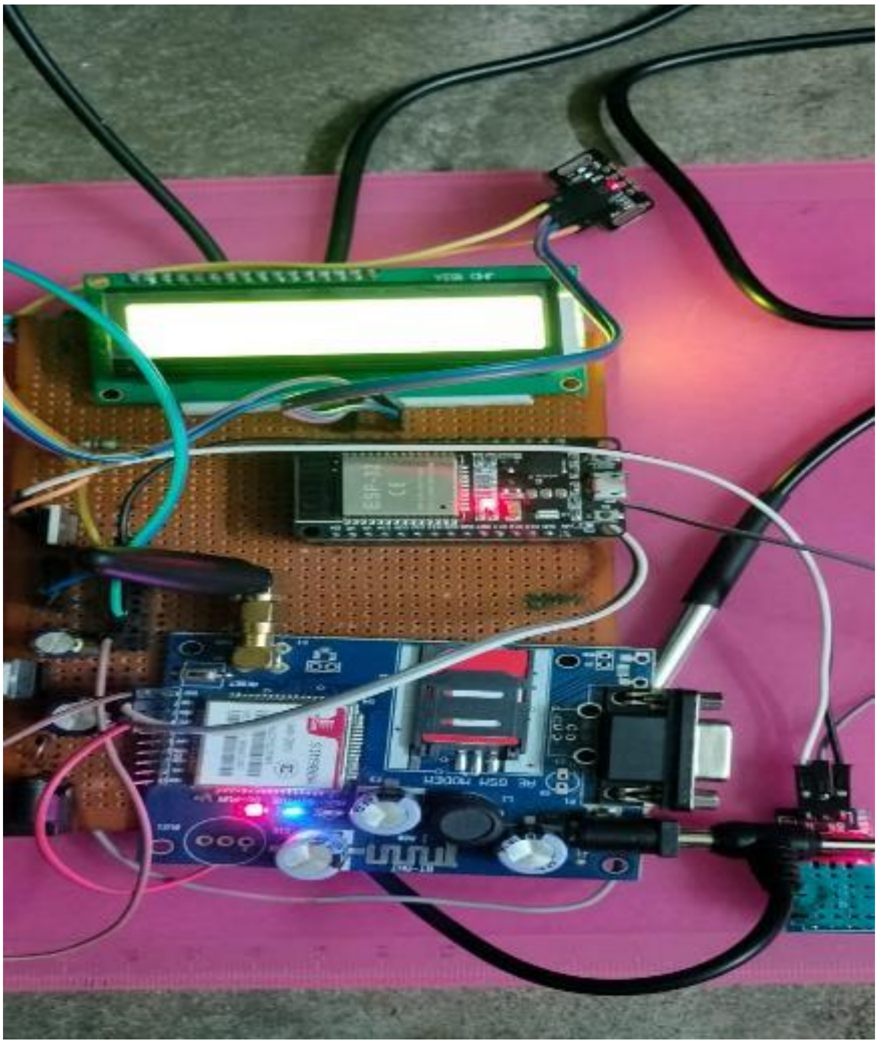


Fig. 4: Hardware Setup and LCD Output During Testing

```
 !!! ALERT !!!
BODY TEMP HIGH
```

Fig. 5: Hardware LCD Output alert

Fig. 6: Hardware LCD Output

B. ThingSpeak Cloud Visualization

The sensor data was successfully uploaded to the ThingSpeak cloud platform through the ESP32 Wi-Fi module. ThingSpeak organized the data into four dedicated fields: Field 1 (Heart Rate), Field 2 (SpO_2), Field 3 (Body Temperature), and Field 4 (Room Temperature). Each field generated a real-time graphical plot, enabling trend analysis over time. Fig. 5 illustrates the ThingSpeak output, including plots for heart rate, SpO_2, and ambient humidity monitoring. The cloud storage with timestamps supports longitudinal health tracking and enables remote access by healthcare providers.

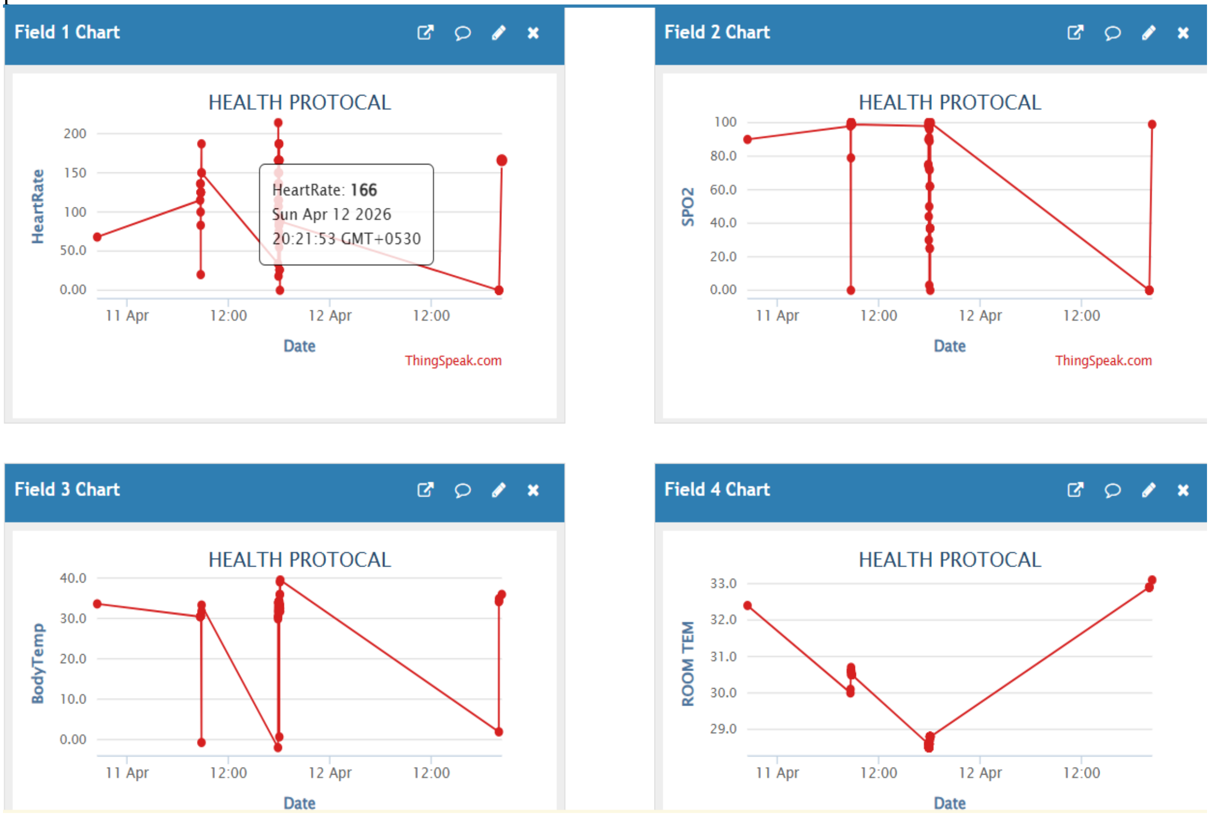
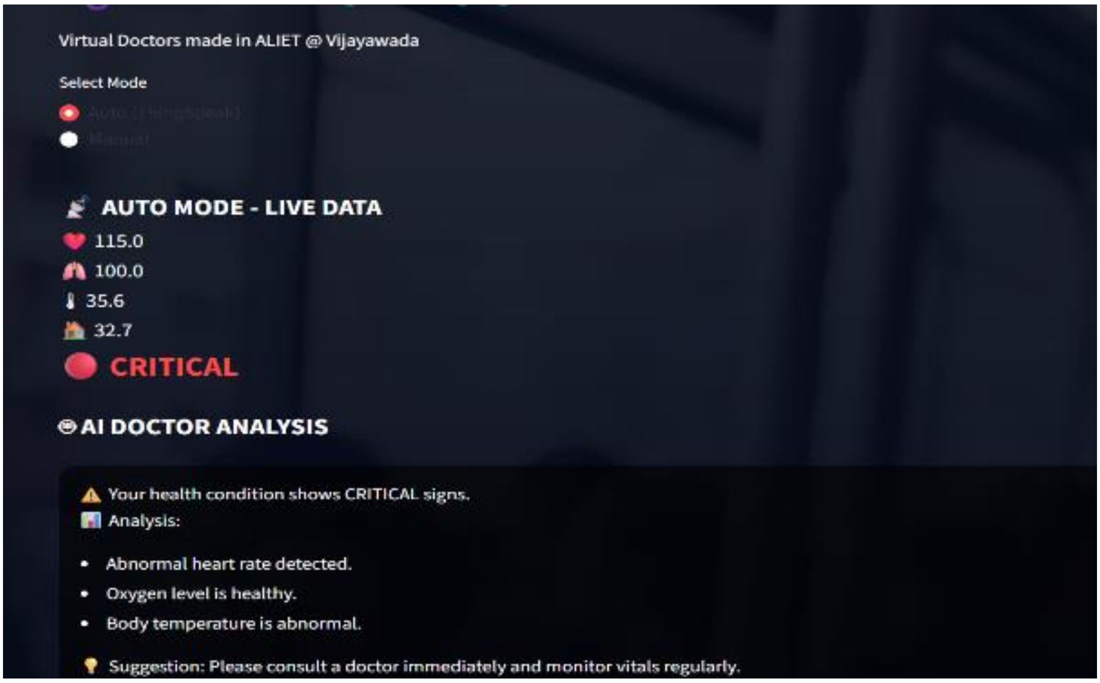


Fig. 7: ThingSpeak Cloud Output – Health Parameter Graphs

C. Machine Learning Health Classification

The Python Streamlit application was tested using both Auto Mode (live ThingSpeak API data) and Manual Mode (direct user inputs). The trained Random Forest classifier accurately classified health conditions based on the four input parameters. The model demonstrated reliable performance, correctly distinguishing Normal and Critical health states. In Auto Mode, the application fetched sensor values in JSON format from the ThingSpeak API and passed them directly to the model without manual intervention, enabling fully automated real-time health classification. Fig. 6 shows the ML model interface during patient condition checking.



Virtual Doctors made in ALIET @ Vijayawada

Select Mode

● Auto (ThingSpeak)

○ Manual

AUTO MODE - LIVE DATA

115.0

100.0

35.6

32.7

CRITICAL

AI DOCTOR ANALYSIS

⚠ Your health condition shows CRITICAL signs.

Analysis:

- Abnormal heart rate detected.
- Oxygen level is healthy.
- Body temperature is abnormal.

Suggestion: Please consult a doctor immediately and monitor vitals regularly.

Fig. 8: ML Model Interface – Patient Condition Classification

D. GSM-Based Emergency Alert

When the ML model detected a Critical health condition, the system automatically triggered the GSM module to send an SMS alert to the pre-configured caregiver phone number. The SMS message included a notification of the abnormal health status, prompting immediate medical attention. The GSM module successfully dispatched alerts within a few seconds of critical condition detection, confirming the responsiveness of the emergency notification mechanism. Fig. 7 illustrates a sample alert SMS received during system testing.

P13:ALERT!
BT:39.04
HR:65
SpO2:37

P13:ALERT!
BT:39.51
HR:88
SpO2:100

10:38 pm

P13:ALERT!
BT:44.19
HR:187
SpO2:69

P13:ALERT!
BT:50.58
HR:0
SpO2:0

10:42 pm

Fig. 9: Sample Emergency Alert SMS Generated by the System

E. System Advantages

The following key advantages were observed during system evaluation: (1) Real-time continuous monitoring of heart rate, SpO_2, body temperature, and room temperature; (2) Remote patient monitoring via ThingSpeak cloud, accessible from any location; (3) Intelligent health classification using a Random Forest ML model, eliminating reliance on fixed thresholds alone; (4) Automated emergency alerting through GSM SMS and voice calls for timely medical intervention; (5) Local real-time display on the LCD for immediate on-site observation; (6) Low cost and portable design suitable for home healthcare and elderly monitoring; (7) Scalability to accommodate additional sensors or advanced ML models.

F. Applications

The proposed system is applicable in a wide range of healthcare scenarios, including: home monitoring for cardiac patients, post-surgery care and hospital ward monitoring, elderly care facilities, rural and remote healthcare delivery, cardiac rehabilitation programs, integration with telemedicine platforms, and clinical research and trial data collection.

V. CONCLUSION

This paper presented the design, implementation, and experimental validation of an IoT and Machine Learning based Smart Health Monitoring System. The system integrates the MAX30102, DS18B20, and DHT11 sensors with an ESP32 microcontroller to continuously monitor heart rate, blood oxygen saturation, body temperature, and ambient conditions. Real-time data is displayed locally on an I2C LCD and transmitted to the ThingSpeak cloud platform for remote monitoring and trend visualization.

A Random Forest ML classifier, deployed through a Python Streamlit application, provides automated health condition classification with high reliability. The GSM-based alert mechanism ensures immediate notification to caregivers during critical health events, significantly enhancing patient safety. The system achieves its core objectives of real-time monitoring, intelligent analysis, remote accessibility, and emergency response in a single low-cost and portable platform.

The experimental results confirm that the integration of IoT and Machine Learning technologies yields a more intelligent and responsive health monitoring system compared to conventional threshold-based approaches. The proposed system offers a strong foundation for next-generation smart healthcare solutions and holds significant potential for deployment in home healthcare, elderly care, and rural medical settings.

VI. FUTURE WORK

Several avenues exist for enhancing and extending the capabilities of the proposed system. First, the ML model can be upgraded to advanced deep learning architectures such as LSTM or CNN-based classifiers to improve prediction accuracy for complex health patterns. Second, a dedicated mobile application can be developed to provide caregivers and doctors with an intuitive interface for monitoring patient health parameters from smartphones. Third, additional biomedical sensors such as ECG sensors, blood pressure sensors, and blood glucose monitoring modules can be integrated to provide a more comprehensive health assessment. Fourth, edge computing capabilities can be incorporated to enable local ML inference on the ESP32 or a companion edge device, reducing dependence on cloud connectivity and lowering response latency. Fifth, the system can be extended to support multi-patient monitoring through a centralized hospital dashboard, enabling healthcare facilities to monitor multiple patients simultaneously.

ACKNOWLEDGMENT

The authors express sincere gratitude to Mr. P. Bose Babu, M.Tech., (Ph.D.), Associate Professor, Department of Electronics and Communication Engineering, Andhra Loyola Institute of Engineering and Technology, for his invaluable guidance and constant encouragement throughout this work. Special thanks are extended to Dr. T. Lakshmi Narayana, Head of the Department of ECE, for his technical and moral support. The authors also acknowledge Dr. O. Mahesh, Principal of ALIET, and Rev. Fr. Dr. B. Joji Reddy S.J., Director, for providing the excellent facilities and infrastructure necessary for this research. Finally, heartfelt thanks are due to all family members for their unwavering support and encouragement.

REFERENCES

[1] A. Gupta, R. Singh, and P. Kumar, "IoT-Based Smart Healthcare Monitoring System Using Wearable Sensors," in *Proc. 2023 IEEE Int. Conf. on Smart Computing and Electronic Enterprise (ICSCEE)*, IEEE, 2023.

[2] M. Khan, S. Ahmed, and T. Ali, "Real-Time IoT-Based Patient Health Monitoring System Using Cloud Computing," in *Proc. 2023 IEEE Int. Conf. on Internet of Things and Intelligence System (IoTaIS)*, IEEE, 2023.

[3] D. Patel and H. Shah, "Smart Health Monitoring System Using IoT and Machine Learning," in *Proc. 2024 IEEE Int. Conf. on Artificial Intelligence and Smart Systems (ICAISS)*, IEEE, 2024.

[4] S. Chatterjee, A. Roy, and S. Mukherjee, "AI-Enabled IoT Framework for Remote Healthcare Monitoring," *IEEE Access*, vol. 12, pp. 45520–45535, 2024. DOI: 10.1109/ACCESS.2024.3364512.

[5] S. Ramesh and V. Prakash, "Intelligent IoT-Based Healthcare Monitoring System Using Edge Computing," in *Proc. 2025 IEEE Int. Conf. on Smart Internet of Things (SmartIoT)*, IEEE, 2025.

[6] G. Y. Jeong, K. H. Yu, and N. G. Kim, "Continuous Blood Pressure Monitoring Using Pulse Wave Transit Time," in *Proc. Int. Conf. on Control, Automation and Systems (ICCAS)*, 2005.

[7] K. Hung, Y. T. Zhang, and B. Tai, "Wearable Medical Devices for Tele-Home Healthcare," in *Proc. 26th Annual Int. Conf. IEEE EMBS*, 2004.

[8] W. M. Jubadi and S. F. A. M. Sahak, "Heartbeat Monitoring Alert via SMS," *Proc. 2009 IEEE Symp. on Industrial Electronics and Applications*, 2009.

[9] O. Apostu, B. Hagiu, and S. Pasca, "Wireless ECG Monitoring and Alarm System Using ZigBee," in *Proc. Int. Symp. on Advanced Topics in Electrical Engineering (ATEE)*, 2011.

[10] X. Fang et al., "An Extensible Embedded Terminal Platform for Wireless Telemonitoring," in *Proc. 2012 Int. Conf. on Information and Automation (ICIA)*, pp. 668–673. DOI: 10.1109/ICInfA.2012.6246761.

[11] G. Motika and A. Prusty, "Wireless Fetal Heartbeat Monitoring System Using ZigBee & IEEE 802.15.4 Standard," in *Proc. 2nd Int. Conf. on Emerging Applications of Information Technology*, 2011, DOI: 10.1109/EAIT.2011.89.