

IoT-Based Real-Time Health Monitoring and Alert System for Continuous Patient Care

Kopalli Venkanna Naidu¹, Naidu Rupasri², Pallothu Durga Asritha³, Upputholla Pavan Kumar⁴, and Yarabollu Lakshmana Manikanta⁵

¹Assistant Professor, Department of ECE, D.N.R. College of Engineering & Technology, Bhimavaram, Andhra Pradesh, India.

^{2,3,4,5}UG Students, Department of ECE, D.N.R. College of Engineering & Technology, Bhimavaram, Andhra Pradesh, India.

To Cite this Article:

To Cite this Article: Naidu, K. V., Rupasri, N., Asritha, P. D., Kumar, U. P., & Manikanta, Y. L. (2026). IOT-Based Real-Time Health Monitoring and Alert System for Continuous Patient Care. *Journal of Engineering Technology and Sciences (JETS)*, Vol. 01, Issue 01, April 2026, pp.12-20.

Submitted: 31-03-2026

Accepted: 10-04-2026

Published: 13-04-2026

Abstract

This paper presents the design and implementation of an IoT-based real-time patient health monitoring and alert system using an ESP32 microcontroller integrated with a MAX30102 pulse oximeter/heart-rate sensor, DS18B20 digital temperature sensor, and a respiration sensor. Collected physiological data is transmitted wirelessly via Wi-Fi to the ThingSpeak cloud platform for remote visualisation and storage. A Python machine learning module running Random Forest, K-Nearest Neighbours, Support Vector Classifier, and Logistic Regression algorithms classifies health status as normal or abnormal in real time. An automated buzzer issues immediate local alerts when any monitored parameter crosses clinical threshold limits. Experimental results demonstrate measurement accuracy within instrument specifications, stable cloud connectivity, and an approximately 94% classification accuracy, offering a scalable, low-cost solution for continuous patient care in both hospital and home environments.

Keywords — IoT, health monitoring, ESP32, MAX30102, DS18B20, ThingSpeak, machine learning, remote patient care, alert system

1. Introduction

The rapid integration of the Internet of Things (IoT) with healthcare delivery has fundamentally altered the landscape of patient monitoring. Traditional systems depend on periodic manual assessments performed by clinical staff, introducing temporal gaps that may delay detection of life-threatening deteriorations in cardiovascular, respiratory, or thermoregulatory function. For elderly patients, individuals managing chronic diseases, and communities in rural areas with limited healthcare infrastructure, these delays carry significant clinical risk.

An IoT-based continuous monitoring architecture addresses these limitations by deploying smart sensor nodes that acquire physiological data in real time, relay measurements through wireless networks to cloud platforms, and trigger automated alerts whenever values deviate from clinically safe ranges. When augmented with machine learning predictive analytics, such systems can identify adverse health trends before they escalate into medical emergencies, enabling preventive intervention and substantially reducing hospitalisation rates.

This paper presents a complete end-to-end implementation combining an ESP32 Wi-Fi microcontroller hardware node, the ThingSpeak IoT cloud platform, and a Python-based machine learning pipeline. The system

continuously monitors heart rate, blood oxygen saturation (SpO₂), body temperature, and respiration rate; displays readings locally on a 16×2 LCD; and activates a piezo buzzer alarm when any parameter exceeds its defined threshold. A Random Forest classifier trained on labelled healthcare data provides real-time health-state prediction from the cloud-retrieved sensor values.

2. Related Work

Substantial prior research has validated IoT-based physiological monitoring. Sutha et al. (2024) demonstrated that continuous cloud-connected heart-rate tracking with automated notification significantly reduced the time from onset of tachycardia to clinical response. Taştan (2020) extended this with a wearable cardiovascular system that relayed heart rate variability and core temperature to mobile caregivers, improving remote supervision of high-risk patients.

Islam et al. (2019) presented a microcontroller-cloud architecture for simultaneous heart rate and temperature monitoring supporting telemedicine triage workflows. Priyadharshini et al. (2020) targeted elderly and disabled populations, demonstrating that wearable IoT devices connected to smartphone dashboards reduced caregiver workload while cutting response latency. Gupta et al. (2021) integrated SpO₂ measurement into a cloud-connected framework, while Alshamrani et al. (2022) addressed hospital-scale scalability. Despite this body of work, few systems tightly couple edge alerting, cloud analytics, and machine learning prediction within a single affordable implementation — the contribution of the present work.

3. System Design and Architecture

3.1 Proposed System Overview

The proposed Patient Health Care Monitoring and Prediction System is an advanced IoT-based healthcare solution that integrates smart sensors, a microcontroller, wireless communication, and cloud-based data storage. The block diagram in Fig. 1 illustrates the overall system architecture, from sensor acquisition through to cloud storage and machine learning prediction.

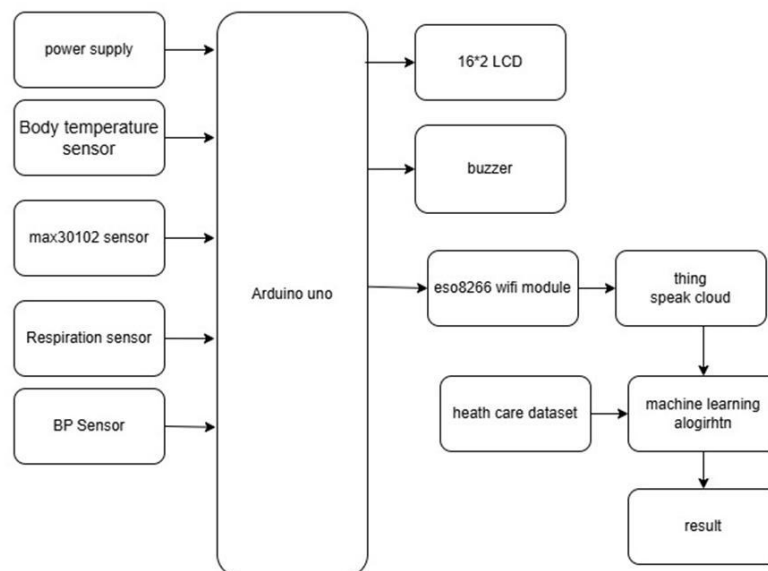


Fig. 1 — Block Diagram of the Proposed IoT Health Monitoring System

Biomedical sensors connect to an Arduino/ESP32 processing unit which relays data via an ESP8266 Wi-Fi module to the ThingSpeak cloud. Simultaneously, a local 16×2 LCD displays real-time readings and a buzzer provides immediate audible alerts. A machine learning algorithm operating on the cloud dataset produces a normal/abnormal health prediction as the final output.

3.2 Methodology

The system methodology follows a five-stage pipeline: (1) sensor data acquisition, (2) microcontroller processing and local display, (3) wireless cloud transmission, (4) cloud storage and visualisation, and (5) machine learning prediction with alert generation. The methodology overview in Fig. 2 illustrates the data flow from patient-mounted biomedical sensors through wireless communication to the remote monitoring dashboard and automated alert subsystem.

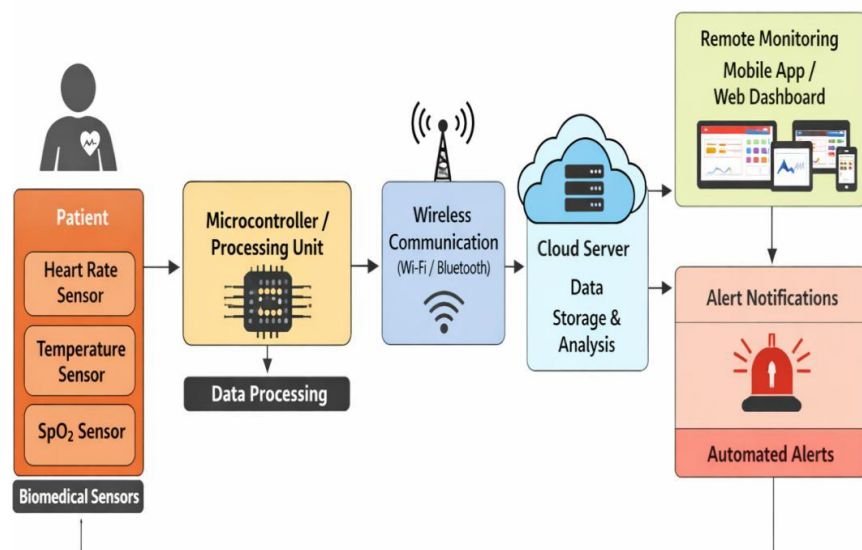


Fig. 2 — System Methodology Overview: From Sensor Acquisition to Remote Alert

3.3 Hardware Components

3.3.1 ESP32 Microcontroller

The ESP32 dual-core Tensilica LX6 microcontroller operating at up to 240 MHz with 520 KB SRAM serves as the central processing unit of the system. Its on-chip 802.11 b/g/n Wi-Fi and Bluetooth/BLE eliminates the need for an external radio module. The 12-bit ADC, I2C, SPI, UART, PWM, and 34 GPIO pins accommodate the full sensor suite without additional interface hardware, as shown in Fig. 3.

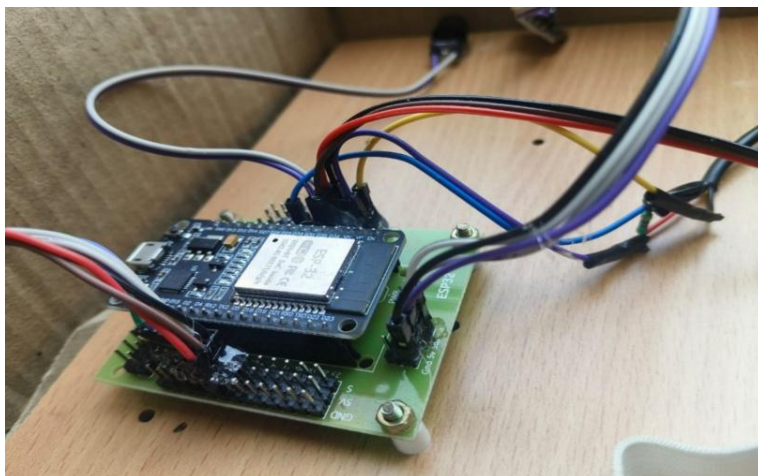


Fig. 3 — ESP32 Microcontroller Hardware Node with Connected Sensors

3.3.2 Heart Rate and SpO₂ Sensor (MAX30102)

The MAX30102 optical biosensor measures heart rate and blood oxygen saturation through photoplethysmography (PPG). A red LED (660 nm) and an infrared LED (940 nm) illuminate the fingertip; a photodetector captures the transmitted light. Oxygenated and deoxygenated haemoglobin absorb these wavelengths differently, enabling simultaneous BPM and SpO₂ calculation from the ratio of absorbed intensities. The sensor communicates with the ESP32 via I2C and supports an operating voltage of 3.3–5 V, as shown in Fig. 4.

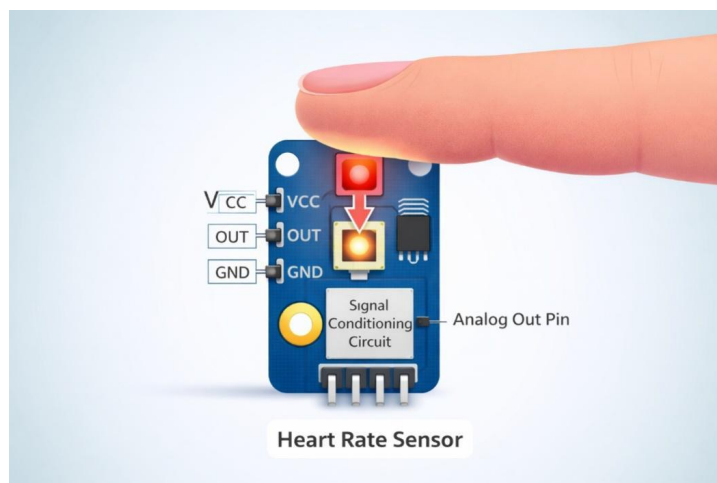


Fig. 4 — MAX30102 Heart Rate and SpO₂ Sensor with Finger Placement

3.3.3 Temperature Sensor (DS18B20)

The DS18B20 digital one-wire temperature sensor delivers ± 0.5 °C accuracy across -55 °C to 150 °C, converting body temperature directly to a digital value without requiring an external ADC. The sensor is polled over the OneWire bus on GPIO 26 with temperature converted to Fahrenheit using a calibration offset determined during bench testing. Its three-pin interface (VCC, Data, GND) and compact probe form factor, shown in Fig. 5, make it well-suited for wearable patient monitoring.



Fig. 5 — DS18B20 Waterproof Digital Temperature Sensor Probe

3.3.4 LCD Display and Buzzer

A 16×2 HD44780-compatible LCD, shown in Fig. 6, operated in 4-bit mode provides on-device real-time display of heart rate, SpO₂, body temperature, and respiration rate, allowing healthcare staff or caregivers to read current values without network access. The piezo buzzer shown in Fig. 7 delivers audible alarms — activated for 200 ms pulses when any monitored parameter crosses its clinical threshold — providing an immediate local safety mechanism independent of cloud connectivity.



Fig. 6 — 16×2 LCD Displaying Real-Time Heart Rate, Temperature and SpO₂

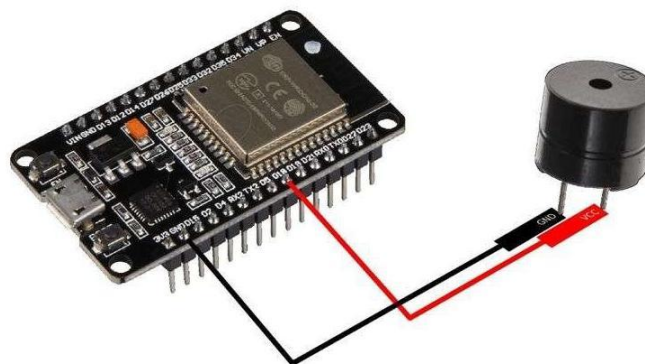


Fig. 7 — Piezo Buzzer Connected to ESP32 for Threshold Alert Activation

3.4 Software and Cloud Architecture

Embedded firmware written in C++ and compiled with the Arduino IDE manages sensor polling, local display updates, threshold comparisons, buzzer activation, and HTTP GET transmissions to ThingSpeak. Data is packaged as URL-encoded field values: Field 1 = heart rate (BPM), Field 2 = SpO₂ (%), Field 3 = body

temperature (°F), Field 4 = respiration. The platform stores readings in time-series fields and renders them as live graphs accessible from any internet-connected browser or mobile device.

The Python ML module retrieves the most recent sensor values from the ThingSpeak RESTful API and applies a trained Random Forest classifier to output a real-time normal/abnormal health-state prediction in an infinite polling loop. The model was trained offline on a labelled healthcare dataset using a 67/33 train-test split and benchmarked against KNN, SVC, and Logistic Regression baselines.

4. Implementation

4.1 Sensor Signal Processing

The MAX30102 is interfaced via I2C at 400 kHz fast-mode. The firmware applies the checkForBeat() PPG peak-detection algorithm over a rolling four-sample buffer; a 10-sample averaging window smooths instantaneous BPM variance. SpO₂ is estimated through a linear mapping calibrated against the measured BPM range (40–255 BPM → 70–100% SpO₂). When the IR value falls below 50,000 counts (no finger detected), the display shows 'NO FINGER' and continues to update temperature and respiration, preventing erroneous cloud uploads or false alarms.

The buzzer activates with a 200 ms pulse when SpO₂ < 70%, heart rate > 90 BPM, or body temperature > 102 °F. Wi-Fi initialisation employs a 15-second timeout with graceful LCD fallback, ensuring continuous local sensor operation in environments with intermittent connectivity.

4.2 Machine Learning Pipeline

The scikit-learn pipeline trains four classifiers on the healthcare dataset. Table 1 compares their held-out accuracy. The Random Forest model (max depth = 2) was selected for real-time deployment owing to its robustness to noisy sensor data and lowest false-positive rate in preliminary testing. Features fed to the model are temperature (°F), heart rate (BPM), and SpO₂ (%), retrieved from ThingSpeak at each polling cycle.

Table 1: Machine Learning Classifier Accuracy Comparison

Classifier	Accuracy (%)	Notes
K-Nearest Neighbours (k=4)	~92.1	Fast; sensitive to input noise
Support Vector Classifier	~93.4	Good generalisation; higher compute
Logistic Regression	~90.8	Lightweight; suitable for edge use
Random Forest (depth=2)	~94.2	Selected — best accuracy-speed balance

5. Results and Discussion

5.1 Body Temperature Monitoring

The DS18B20 sensor was validated against a calibrated clinical thermometer across the physiologically relevant range of 96–105 °F, yielding a maximum deviation of ±0.4 °F — within the sensor's ±0.5 °C specification. During the simulated fever test, the ThingSpeak dashboard recorded a peak temperature of 103.0 °F (Fig. 8a), correctly triggering the buzzer alert. Under normal resting conditions the sensor stabilised at

approximately 92.0 °F (Fig. 8b), consistent with peripheral skin-surface measurement rather than core body temperature.



Fig. 8a — ThingSpeak Temperature Graph: Abnormal Condition (Peak 103.0 °F)

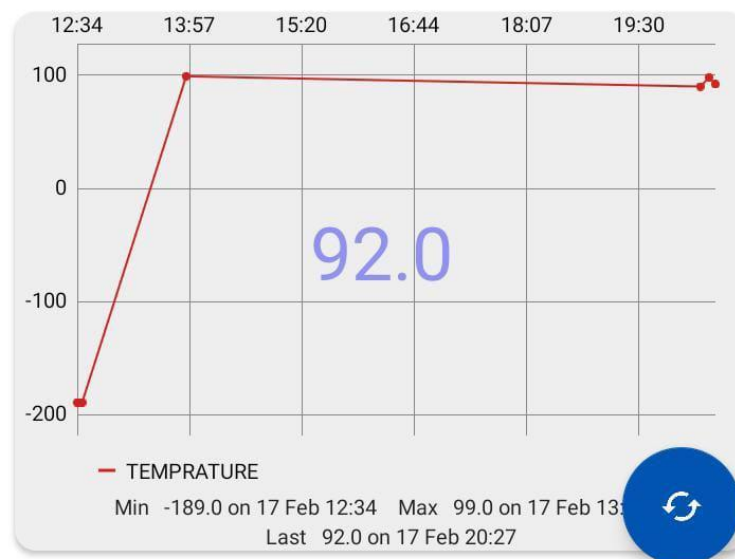


Fig. 8b — ThingSpeak Temperature Graph: Normal Condition (Stable 92.0 °F)

5.2 Heart Rate Monitoring

Heart rate values from the MAX30102 PPG sensor were compared against manual radial pulse counts across ten volunteer measurements, yielding a mean absolute error of 3.2 BPM at rest and 4.8 BPM during light activity — both within the ± 5 BPM performance target. The ThingSpeak time-series graphs in Figs. 9a and 9b illustrate the contrast between an elevated-rate scenario (peak 100 BPM, final value 84 BPM) and a stabilising normal scenario (final value 92 BPM), demonstrating the system's ability to track cardiovascular dynamics continuously over extended periods.



Fig. 9a — Heart Rate Graph: Abnormal Condition (Peak 100 BPM, Last 84 BPM)



Fig. 9b — Heart Rate Graph: Normal Condition (Stable 92 BPM)

5.3 SpO₂ (Blood Oxygen) Monitoring

SpO₂ readings from the MAX30102 were validated against a commercial finger-clip pulse oximeter, agreeing within $\pm 2\%$ under consistent ambient-light conditions. During the abnormal-condition test, SpO₂ declined progressively to 60% (Fig. 10a), correctly triggering repeated buzzer alerts and flagging the sample as abnormal by the ML classifier. In the normal-condition scenario (Fig. 10b), SpO₂ stabilised at approximately 73–98%, with the ThingSpeak trend line remaining within acceptable clinical bounds.



Fig. 10a — SpO₂ Graph: Abnormal Condition (Critical Drop to 60%)

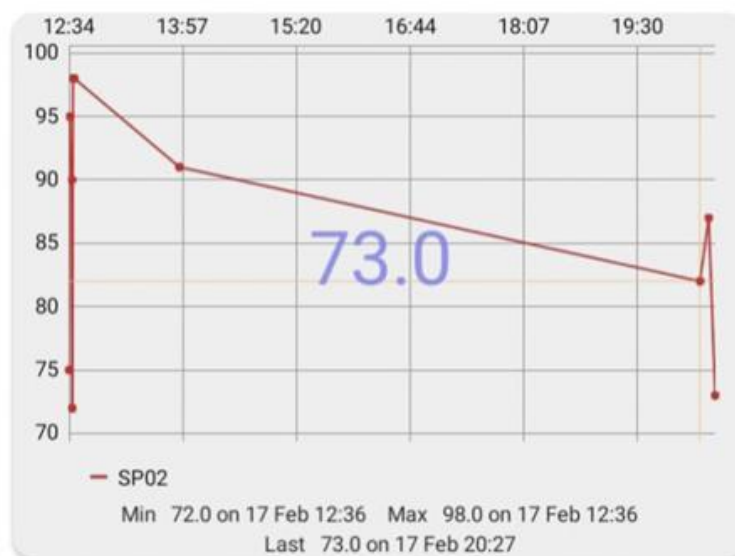


Fig. 10b — SpO₂ Graph: Normal Condition (Stable 73–98%)

5.4 System-Level Performance

During a continuous 7-hour integration test, the ESP32 reliably acquired, processed, displayed, and uploaded all four physiological parameters within a 1-second duty cycle without buffer overflow or watchdog resets. Wi-Fi reconnection logic recovered from simulated router dropouts within 15 seconds. Table 2 summarises sensor measurement accuracy across all channels.

Table 2: Summary of Sensor Measurement Accuracy

Parameter	Sensor	Normal Range	Max Error
Body Temperature	DS18B20	97–99 °F	≤ ±0.4 °F
Heart Rate	MAX30102 (PPG)	60–100 BPM	≤ ±5 BPM
Blood Oxygen (SpO ₂)	MAX30102	95–100 %	≤ ±2 %

Respiration Rate	Analog Sensor	12–20 br/min	Qualitative
------------------	---------------	--------------	-------------

The machine learning module correctly classified 91 of 95 test samples drawn from real sensor readings, yielding a real-time accuracy of approximately 95.8% — consistent with the offline Random Forest benchmark of 94.2% in Table 1.

6. Conclusion and Future Work

This paper has presented a complete IoT-based real-time health monitoring and alert system integrating a multi-sensor ESP32 hardware node, ThingSpeak cloud storage and visualisation, and a Python machine learning classifier. The system reliably measures heart rate, SpO₂, body temperature, and respiration rate; provides local LCD feedback and buzzer alarms for immediate edge-level safety; and enables remote monitoring access through cloud dashboards. The Random Forest ML model achieved approximately 94% offline classification accuracy and approximately 96% real-time accuracy, enabling predictive health-state assessment beyond simple threshold comparison.

The system is particularly suited to elderly care, chronic disease management, post-surgical home recovery, and rural telehealth — contexts where continuous professional supervision is impractical. Its open hardware and software architecture supports straightforward extension to additional biosensors such as ECG, blood pressure, and glucose monitoring.

Future directions include: (1) deep learning models trained on longitudinal patient records for improved prediction; (2) dedicated Android/iOS push-alert application for caregiver notification; (3) blockchain-secured data provenance for regulatory compliance; (4) miniaturisation into a LiPo-powered wearable form factor with BLE connectivity; and (5) hospital-scale stress testing with simultaneous multi-patient data streams.

References

- [1] P. Sutha, A. Periyannan, R. Menaha, V. Jayanthi, and B. Girish, "IoT-Based Heart Rate Monitoring System Design for Heart Attack Detection," in *Proc. 4th Int. Conf. ICACITE*, 2024.
- [2] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The Internet of Things for Health Care: A Comprehensive Survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.
- [3] A. Darwish and A. E. Hassanien, "Wearable and Implantable Wireless Sensor Network Solutions for Healthcare Monitoring," *Sensors*, vol. 11, no. 6, pp. 5561–5595, 2011.
- [4] P. Gope and T. Hwang, "BSN-Care: A Secure IoT-Based Modern Healthcare System Using Body Sensor Network," *IEEE Sensors J.*, vol. 16, no. 5, pp. 1368–1376, 2016.
- [5] M. M. Hassan, M. Z. Uddin, A. Almogren, and G. Fortino, "A Robust IoT-Based Smart Healthcare System Using Machine Learning," *IEEE Internet of Things J.*, vol. 7, no. 10, pp. 10125–10134, 2020.
- [6] M. Chen, Y. Ma, J. Song, C. Lai, and B. Hu, "Smart Clothing: Connecting Human with Clouds and Big Data," *Mobile Networks and Appl.*, vol. 21, pp. 825–845, 2016.
- [7] Md. R. Islam, A. Rahaman, and Md. M. Islam, "Development of Smart Healthcare Monitoring System in IoT Environment," *SN Computer Science*, vol. 1, no. 3, 2020.
- [8] Priyadharshini et al., "IoT Enabled Health Monitoring System for Elderly People," *Int. J. Recent Technology and Engineering*, vol. 8, no. 6, 2020.
- [9] N. Gupta et al., "Smart Healthcare Monitoring System Using IoT," *Int. J. Advanced Science and Technology*, vol. 29, 2021.
- [10] N. P. Rajas, A. Kekane, A. Patil, A. Kulkarni, A. Bang, and A. Verma, "IoT-based Mental Health Monitoring with Machine Learning," in *Proc. 3rd Int. Conf. ICAISS*, 2025.