IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834, p- ISSN: 2278-8735.Volume 21, Issue 6, Ser. 1 (Nov. – Dec. 2026), PP 08-17
www.iosrjournals.org

Integrated Iot-Based Platform for Health Monitoring and Diagnostics with Machine Learning and Cloud Support

S.Surekha[1], M.Govind[2], and R. Durga Gopal[3]
[1]*PG Scholar, Department of ECE, Joginpally B. R. Engineering College, India*
[2]*Associate Professor, Department of ECE, Joginpally B. R. Engineering College, India*
[3]*Associate Professor, Department of ECE, Joginpally B. R. Engineering College, India*

Abstract: *The rapid advancement of the Internet of Things (IoT), machine learning (ML), and cloud computing has paved the way for innovative healthcare monitoring and early disease detection. Traditional healthcare systems often rely on periodic clinical examinations, manual data entry, and hospital-based diagnostic procedures, which can delay the identification of abnormal health conditions. This study introduces a Smart IoT-Enabled Health Monitoring and Diagnostic Platform that utilizes machine learning and cloud connectivity for continuous acquisition, analysis, storage, and visualization of patient health parameters. The platform integrates IoT-enabled physiological sensors with an embedded processing unit to collect data such as heart rate, blood oxygen saturation, body temperature, blood pressure, and other relevant health indicators. The collected data is securely transmitted to a cloud platform for real-time monitoring and historical analysis. A machine learning module analyzes physiological data to identify abnormal patterns and generate preliminary health-risk predictions. A cloud-based dashboard enables healthcare professionals and authorized users to remotely access patient information, observe trends, and receive alerts when critical thresholds or anomalous patterns are detected. The proposed architecture combines real-time sensing, edge processing, cloud connectivity, machine learning, and remote healthcare services into a unified platform. The system is intended to support early intervention, remote patient monitoring, and data-driven clinical decision support while reducing dependence on continuous physical hospital visits. Experimental evaluation can be performed using accuracy, precision, recall, F1-score, latency, network utilization, and system reliability to assess the effectiveness of the proposed framework.*
Key Word*: Internet of Things, Smart Healthcare, Machine Learning, Cloud Computing, Remote Patient Monitoring, Health Monitoring, Disease Prediction, IoT Sensors, Edge Computing.*

Date of Submission: 02-11-2026 Date of Acceptance: 13-11-2026

I. Introduction

Healthcare systems are experiencing substantial transformation driven by the increasing integration of digital technologies. The advancement of the Internet of Things (IoT), artificial intelligence (AI), machine learning (ML), wireless communication, edge computing, and cloud platforms has facilitated the transition of healthcare services from traditional hospital-centered models. These technologies enable the continuous monitoring of physiological parameters and offer opportunities for the early detection of potentially abnormal health conditions [1]. Traditional health monitoring typically necessitates patient visits to healthcare facilities for periodic examinations. While conventional clinical procedures yield reliable medical assessments, they may not be optimal for continuous monitoring, especially for elderly individuals, patients with chronic conditions, those residing in remote areas, and patients requiring post-treatment observation [2]. Additionally, manually recorded health information may be fragmented and challenging to analyze over extended periods [3].

The Internet of Things (IoT) offers an efficient framework for overcoming existing limitations by interconnecting physiological sensors, embedded systems, communication networks, cloud platforms, and healthcare applications. These sensors continuously gather physiological data, which is then transmitted to processing units. Subsequently, the data is stored within cloud infrastructure, ensuring accessibility for authorized healthcare professionals. Nonetheless, the mere collection and display of physiological data do not suffice for contemporary healthcare applications [4]. The substantial volume of sensor data necessitates automated analysis to detect abnormal patterns. Machine learning techniques can discern relationships within historical health data, aiding in the classification of health states or the estimation of potential risks [5]. Therefore, the integration of machine learning with IoT-based monitoring can convert a traditional sensing system into an intelligent health-monitoring and diagnostic-support platform [6]. This study introduces a smart healthcare architecture that incorporates IoT sensing, edge processing, machine learning, cloud connectivity, remote visualization, and alert generation. The proposed platform continuously monitors selected physiological parameters, transmits

measurements to a cloud environment, analyzes the data using machine learning algorithms, and provides preliminary risk information to authorized users.

The major contributions of this study are as follows:

1. Development of an integrated IoT architecture for continuous physiological monitoring.
2. Integration of machine learning for automated health-status classification and risk prediction.
3. Cloud-based storage and visualization of real-time and historical health information.
4. Real-time abnormal-condition detection and alert generation.
5. Design of a scalable architecture capable of supporting multiple patients and healthcare devices.

Provision of a foundation for remote healthcare monitoring and preliminary diagnostic decision support.

II. Material And Methods

The system consists of six major layers: **Sensing, Edge/Device Processing, Communication, Cloud, and Machine Learning Layer.** **F**igure 1 shows the data flow architecture for IoT based health monitoring system. Figure 2 shows the architecture of wearable devices connected to the data management system integrated through the cloud.

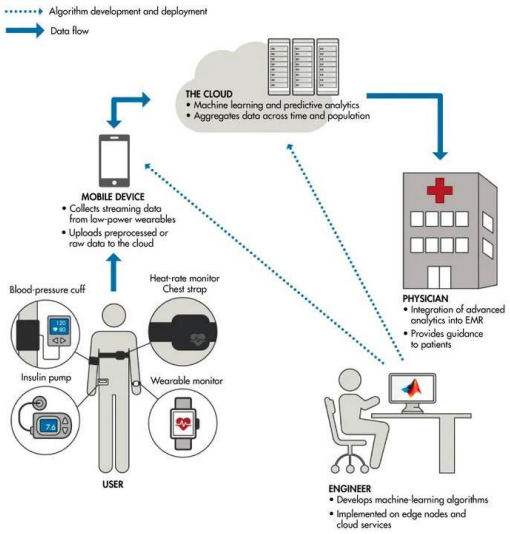

Figure 1: Architecture of IoT-Based Cloud-Integrated Smart Healthcare Monitoring System

Sensing Layer

The sensing layer consists of biomedical sensors used to measure physiological parameters. Depending on the application, the platform can incorporate a heart-rate sensor, SpO_2 sensor, body-temperature sensor, blood-pressure sensor, ECG sensor, respiratory-rate sensor, glucose sensor/interface, and activity or motion sensor[7]. The sensors continuously acquire measurements from the patient. The selection of sensors can be customized according to the intended healthcare application [8].

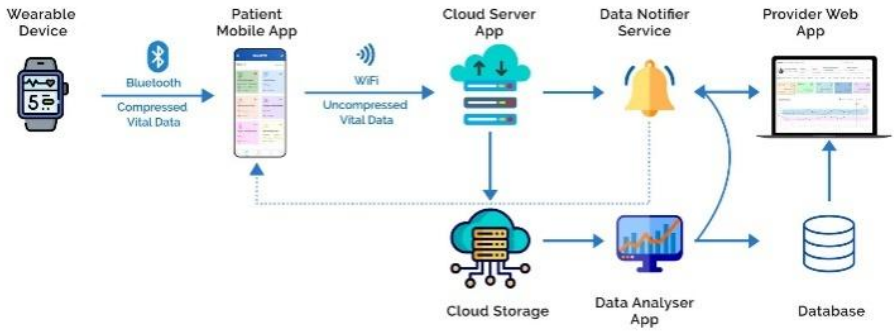

Figure 2: Architecture of Wearable IoT-Based Patient Health Monitoring and Cloud Data Management System

Edge Processing Layer

An embedded controller such as an ESP32, Raspberry Pi, or comparable IoT processing unit receives the sensor measurements. The edge layer performs preliminary operations such as sensor-data acquisition, noise filtering,

data normalization, missing-value detection, timestamp generation, preliminary threshold checking and data packet formation. Edge processing reduces unnecessary communication and allows basic emergency conditions to be detected locally [2].

Communication Layer

The processed measurements are transmitted through wireless communication technologies such as Wi-Fi, Bluetooth/BLE, LoRaWAN, 4G/5G and MQTT-based communication. MQTT can be used as a lightweight communication protocol between IoT devices and cloud services [9].

Cloud Layer

The cloud platform acts as the central data-management environment. It provides patient-data storage, sensor-data management, real-time data ingestion, historical data analysis, user authentication, API services, dashboard connectivity, machine-learning integration, and alert management [10]. Cloud storage allows authorized healthcare professionals to access patient information remotely.

Machine Learning Layer

The ML layer processes collected physiological measurements and determines whether the patient's current health state is normal or potentially abnormal. Possible classification models include logistic regression, decision tree, random forest, support vector machine, K-Nearest Neighbors, gradient boosting, and Artificial Neural Network. For structured physiological datasets, random forest can serve as a strong baseline because it can model nonlinear relationships and provide feature-importance information [11].

Application Layer

The application layer provides interfaces for patients, doctors, healthcare professionals and administrators [12]. The dashboard displays current readings, historical trends, ML predictions, patient profiles, and warning notifications.

System Architecture

The overall workflow of the proposed platform is as follows:

Patient → Sensors → IoT Controller → Edge Processing → Wireless Network → Cloud Platform → Data Processing → Machine Learning → Prediction → Dashboard/Alert

The system can be represented conceptually as shown in figure 3 below.

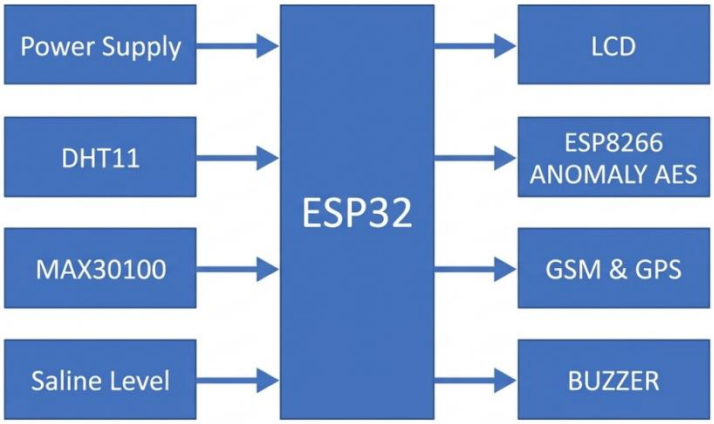

Figure 3: Block diagram of the system architecture for the proposed system

Procedure methodology:

Data Acquisition

The system periodically samples physiological measurements from connected sensors [13]. Each measurement is associated with a timestamp and patient identifier.

A sample record can be represented as:

$$D_i = \{H, R_i, S, p, O, 2_i Temp_i BP_i ECG_i T_i\}$$

where:

- HR= heart rate
- SpO_2= oxygen saturation
- $Temp$= body temperature
- BP= blood pressure
- ECG= electrocardiographic information
- T= timestamp

Data Preprocessing

Raw sensor measurements may contain noise, missing values, outliers, and inconsistent sampling intervals. Therefore, preprocessing is performed before ML analysis. The preprocessing pipeline consists of:

1. Data validation
2. Missing-value handling
3. Noise filtering
4. Outlier detection
5. Feature extraction
6. Normalization
7. Dataset splitting

Min-Max scaling can be applied for normalization:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

This transforms numerical features into a standardized range.

Feature Engineering

Relevant physiological features are extracted from the collected measurements. Possible features include average, maximum and minimum heart rate. The Heart-rate variability, Average and Minimum SpO_2, Body temperature, Systolic and Diastolic blood pressure. Respiratory rate, ECG-derived features, Temporal variations and Rate of change [14]. Feature engineering improves the ability of ML models to distinguish between normal and abnormal health patterns.

Machine Learning Model

The proposed framework can evaluate multiple ML algorithms and select the best-performing model based on experimental results. A general classification function can be expressed as:

$$\hat{y} = f(X; \theta)$$

where:

- X represents the physiological feature vector,
- θrepresents model parameters,
- $\hat{y}$ represents the predicted health state.

For example, a Random Forest classifier constructs multiple decision trees and combines their outputs:

$$\hat{y} = \text{mode}\{T_1, (X), T_2, (X) \ldots T_n(X)\}$$

The resulting prediction can be categorized into classes such as **Normal, Low Risk, Moderate Risk and High Risk.** The actual clinical categories should be defined using an appropriate validated dataset and medical expertise rather than arbitrary thresholds.

Cloud Connectivity

The cloud component provides centralized storage and remote access. The data flow can be expressed as:

Sensor Data
↓
IoT Gateway
↓
MQTT/REST API
↓
Cloud Server
↓
Database
↓
ML Prediction
↓
Dashboard

The cloud infrastructure enables authorized users to retrieve real-time and historical information from multiple devices. A cloud database may store records in the format as shown in table 1 below.

Table 1: Fields recorded in the cloud.

Field	**Description**
Patient ID	Unique patient identifier
Timestamp	Measurement time

Field	Description
Heart Rate	BPM
SpO_2	Oxygen saturation
Temperature	Body temperature
Systolic BP	Systolic pressure
Diastolic BP	Diastolic pressure
ECG Features	Extracted ECG parameters
ML Prediction	Predicted health state
Risk Score	Estimated model score
Alert Status	Normal/Warning/Critical

Alert and Notification Mechanism

The proposed system incorporates an alert mechanism to identify potentially abnormal measurements. The alert engine can operate using two complementary mechanisms: **Threshold-Based Detection -**A predefined clinically appropriate threshold can be used to identify immediate abnormal readings and **ML-Based Detection -** The ML model evaluates multiple physiological parameters simultaneously to identify patterns that may not be apparent from a single measurement. The combination of threshold-based and ML-based mechanisms can improve robustness, provided that thresholds and model outputs are clinically validated.

User Interface

The proposed dashboard provides a centralized view of patient information. Important dashboard components include patient profile, real-time vital signs, historical graphs, health-status indicator, ML prediction, risk score, alert notifications, device status, sensor connectivity and patient history [15]. Healthcare professionals can use historical graphs to identify changes in physiological measurements over time.

Experimental Setup

A prototype implementation can consist of the following:

Hardware

- ESP32/Raspberry Pi
- MAX30102 or equivalent pulse/SpO_2 sensor
- Temperature sensor
- Blood-pressure sensing interface
- ECG sensor
- Wi-Fi/BLE connectivity
- Optional OLED/LCD display

Software

- Python
- Machine learning libraries
- Flask/FastAPI or equivalent backend
- HTML/CSS/JavaScript or a frontend framework
- Cloud database
- MQTT broker
- Data visualization libraries

The ML model should be trained using a sufficiently large, representative, and appropriately labeled dataset. The dataset should be divided into training, validation, and testing subsets.

Statistical analysis:

The proposed system should be evaluated at both the **ML level** and the **IoT/cloud system level**.

Classification Metrics

Accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision:

$$Precision = \frac{TP}{TP + FP}$$

Recall:

$$Recall = \frac{TP}{TP + FN}$$

F1-score:

$$F1 = 2\frac{Precision \times Recall}{Precision + Recall}$$

where:

- TP = True Positive
- TN = True Negative
- FP = False Positive
- FN = False Negative

IoT Performance Metrics

The system can additionally be evaluated using sensor accuracy, end-to-end latency, cloud upload latency, packet-loss rate, network bandwidth, energy consumption, device uptime, alert-generation time, and cloud availability.

Security and Privacy

Healthcare information is highly sensitive, making security an important component of the proposed architecture. The system should incorporate user authentication, role-based access control, encrypted communication, secure API endpoints, encrypted cloud storage, device authentication, access logging, secure credential management and data anonymization for research datasets [16]. Patient identifiers should be separated from analytical datasets whenever possible.

III. Result

The gadget is linked to the system first and allows for message based communication. The connected mobile phone can then transmit the prescribe command to register the mobile number. For example, the user sends a message in the format. (cell number). After successful processing, the system will respond with a confirmation message indicating the mobile number has been successfully registered, as shown in figure 4. This registration step is used to identify the user's mobile number with the device for future communication and notifications. Successful registration means that the device can receive and process the specified registration command and send back a confirmation message.

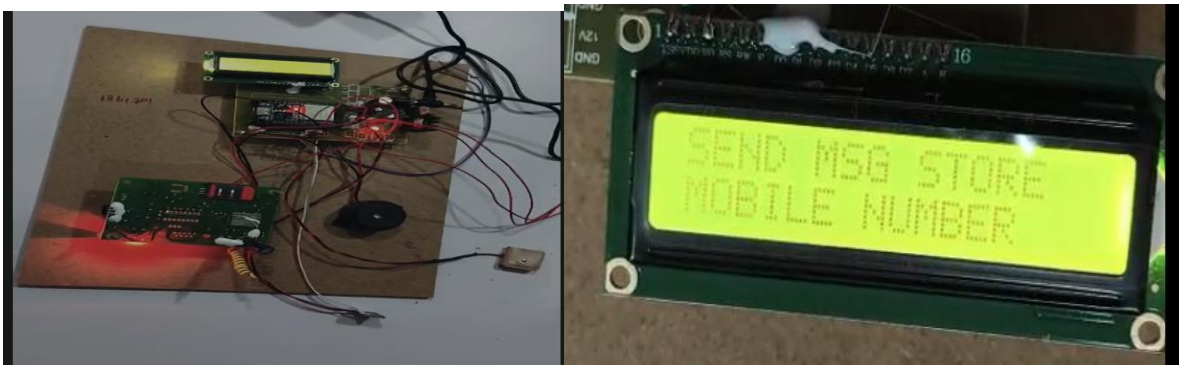

Fig 4: Device Connection and Mobile Number Registration Confirmation

When we give * mobile number, we can see the mobile number displayed on the LCD as shown in Figure 5. As soon as the sensors are mounted to the body in the intended measurement position, the system starts to acquire and display the available sensor readings. The prototype measured the following parameters:

1. **Heart Rate:** The MAX30100 sensor measures the pulse rate and reports the estimated number of heart beats per minute (BPM).
2. **Blood Oxygen Saturation (SpO_2):** The MAX30100 estimates the percentage of oxygen saturation in the blood.

3. **Temperature:** The connected temperature sensor displays its measured temperature value. If the DHT11 is used, this value represents ambient temperature rather than body temperature.

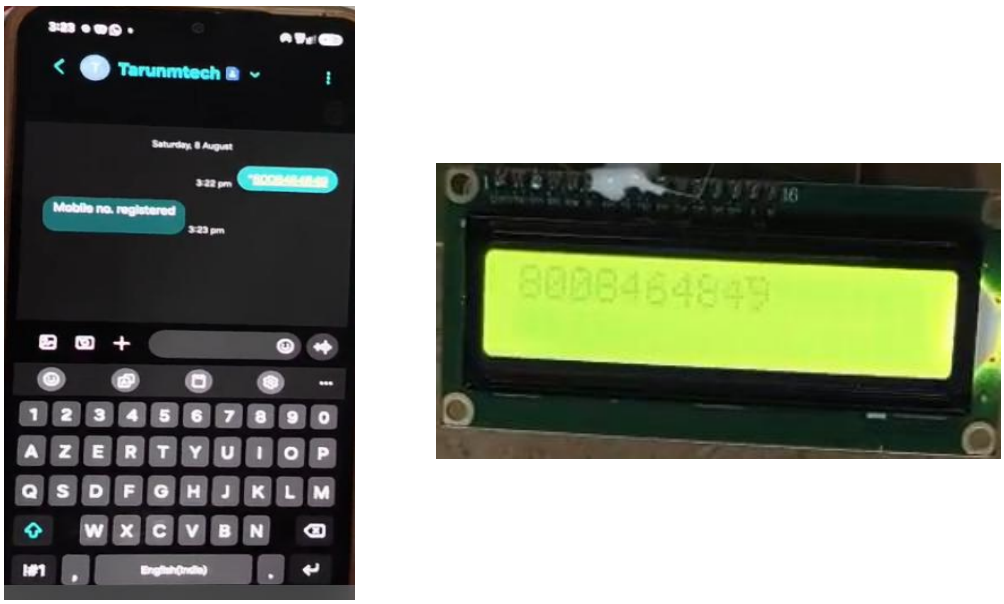

Figure 5: information displayed on the mobile screen and the LCD

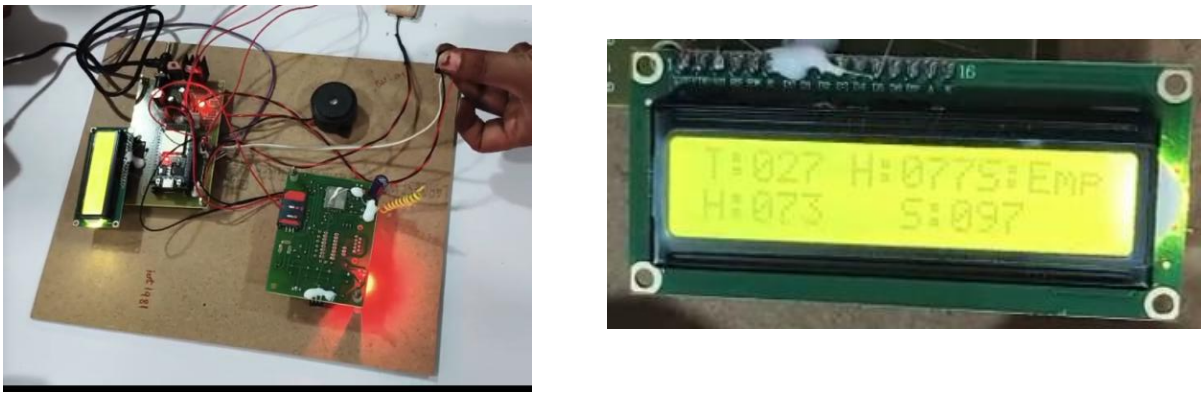

Fig 6: Display of Heart Rate, SpO_2, and Temperature Readings

The readings were processed in the microcontroller and displayed for observation as shown in Figure 6. The functionality shows how the prototype can collect and display the selected health-related measurements. The reading accuracy is affected by sensor position, motion, signal quality, calibration and operational conditions. Therefore, the measurements shown should be regarded as prototype readings rather than clinically confirmed diagnosis conclusions.

Saline Bottle Level Monitoring

The prototype consists of the wires or conductive probes placed at some levels of the saline bottle to detect the liquid level. The measured level can be shown by the relevant probe setup (Figure 7), when the saline solution reaches or dips below a specified detecting point. The system distinguishes the configured levels using separate wire connections. The blue and grey cables are Level 1 as confirmed after testing the layout that was installed. This setup allows the monitoring of the saline bottle level and can be expanded to warn when the liquid reaches a predetermined threshold. Its reliability depends on the configuration of the probes, electrical connections, conductivity of the liquid, and sensing circuit. The saline monitoring tool was designed to indicate the liquid level. It does not manage the infusion process and does not ensure that the fluid is safely administered.

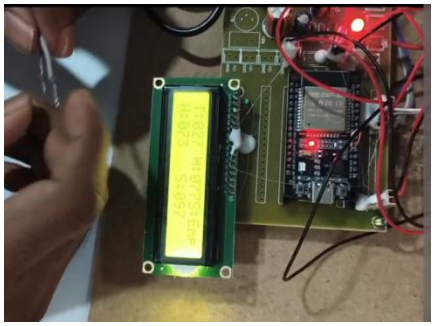

Fig 7: Saline Bottle Level Detection Using Connected Wires

The Current prototype setup has a combination of blue and grey wire as Level 1. Depending on the implemented software, the information can be shown locally or transferred to the server when the system detects a corresponding state.

Data Upload to the Web Server

The communication interface of the system transmits the sensor readings and monitoring information to the configured web server Project.factory.in. If the transmission is successful, the received data are available on the server remote monitoring purposes. The function allows access to the web-based monitoring platform from the hardware prototype. This allows authorized users to access uploaded data without relying just on a local display, as shown in Figure 8. The server integration supports the following functions:

1. Transmission of sensor readings from the device.
2. Uploading of monitoring data to the web server.
3. Remote access to uploaded information.
4. Centralized viewing of available measurements.
5. A foundation for historical data storage and analysis, depending on the server's configured features.

Successful data uploading demonstrates the feasibility of integrating embedded sensing with an IoT-based web monitoring system.



Fig 8 : Sensor Data Uploaded to Project.factory

GPS-Based Location Tracking

When a compatible GPS module is attached and obtains a correct position fix, the system can obtain geographical coordinates that show the location of the device, as shown in Figure 9. These coordinates can help determine the approximate position of the gadget in case of an emergency. When combined with a compatible communication module and notification mechanism, location information can be sent to an authorized mobile number or a monitoring platform. GPS tracking can assist careers or support staff locate a person wearing the device. However, for the system to work properly, the GPS signal must be available, the module must be configured, and the method for transmitting the location must be implemented.

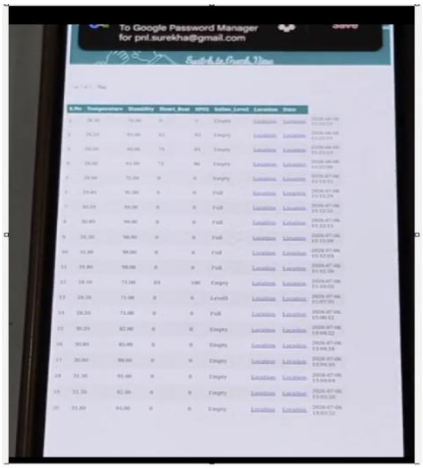

Fig 9: GPS-Based Location Tracking Interface

IV. Advantages and Limitations

The proposed platform provides several potential advantages such as **continuous monitoring, remote accessibility, intelligent analysis, early warning, historical analysis, scalability, reduced manual effort, and real-time visualization.**

Despite its potential, the proposed system has several limitations. First, sensor measurements can be affected by sensor placement, motion artifacts, environmental conditions, and hardware limitations. Second, ML predictions depend strongly on the quality, diversity, and representativeness of the training dataset. Third, network interruptions may affect real-time cloud synchronization. Fourth, healthcare predictions generated by the system should not automatically be interpreted as definitive medical diagnoses.

The proposed platform should therefore be considered a **health-monitoring and clinical decision-support system**, with final diagnosis and treatment decisions remaining under the responsibility of qualified healthcare professionals.

V. Conclusion

To summaries, the proposed platform provides a framework for the development of a cost effective and scalable IoT based health monitoring solution. It demonstrates the convergence of embedded technology, wireless connectivity, cloud computing and machine learning for remote healthcare applications. More testing, calibration, security evaluation and clinical validation is needed before the system may be used for reliable medical decision making. Models can be personalized using individual patient history, baseline physiological values, and longitudinal trends. Federated learning can allow multiple healthcare institutions to collaboratively train ML models while reducing the need to centralize sensitive patient data.

References

[1]. Hai Ziwei et al., "The Applications of Internet of Things in Smart Healthcare Sectors: A Bibliometric and Deep Study," Heliyon, vol. 10, no. 3, pp. 1-11, 2024.
[2]. Mahdi Safaei Yaraziz, Nader Sohrabi Safa, and Muhammad Ajmal Azad, "Edge Computing in IOT for Smart Healthcare," Journal of Ambient Intelligence and Smart Environments, pp. 1-30, 2024.
[3]. Prerna Kumari, Rupali Agarwal, and Shruti Kumari, Revolutionizing Healthcare: Empowering Faster Treatment with IoT-Powered Smart Healthcare, Optimized Computational Intelligence Driven Decision-Making: Theory, Application and Challenges, pp. 229-249, 2024.
[4]. Suliman Abdulmalek, Abdul Nasir, and Waheb A. Jabbar, "LoRaWAN-Based Hybrid Internet of Wearable Things System Implementation for Smart Healthcare," Internet of Things, vol. 25, pp. 1-21, 2024.
[5]. Maysa Khalil, Qasem Abu Al-Haija, and Samir Ahmad, Healthcare IoT Networks using LPWAN, Low-Power Wide Area Network for Large Scale Internet of Things, 1st ed., pp. 203-216, 2024.
[6]. Zahra Amiri et al., "The Deep Learning Applications in IoT-Based Bio-and Medical Informatics: A Systematic Literature Review," Neural Computing and Applications, vol. 36, no. 11, pp. 5757-5797, 2024.
[7]. Satish kumar Chintala, "IOT and AI Synergy: Remote Patient Monitoring for Improved Healthcare," 2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM), Noida, India, pp. 1-6, 2024.
[8]. Hisham Alasmary, "Scalable Digital Health (SDH): An IoT-Based Scalable Framework for Remote Patient Monitoring," Sensors, vol. 24, no. 4, pp. 1-14, 2024.
[9]. Hilal Ahmad Sofi, "IOT Based Remote Patient Care Bio-Monitoring System in Rural Areas," Biochemical and Cellular Archives, vol. 24, no. 1, 2024.
[10]. Rakesh Ranjan, and Bikash Chandra Sahana, "A Comprehensive Roadmap for Transforming Healthcare from Hospital-Centric to Patient Centric through Healthcare Internet of Things (IoT)," Engineered Science, vol. 30, pp. 1-16, 2024.

[11]. Anichur Rahman et al., "Machine Learning and Deep Learning-Based Approach in Smart Healthcare: Recent Advances, Applications, Challenges and Opportunities," AIMS Public Health, vol. 11, no. 1, pp. 58-109, 2024.
[12]. V. Rohan kumar et al., A Smart Health Care Ecosystem, Recent Trends in Computational Sciences, CRC Press, 1st ed., pp. 339-345, 2024.
[13]. Ali Ala et al., "Enhancing Patient Information Performance in Internet of Things-Based Smart Healthcare System: Hybrid Artificial Intelligence and Optimization Approaches," Engineering Applications of Artificial Intelligence, vol. 131, 2024.
[14]. K.K. Baseer et al., "Healthcare Diagnostics with An Adaptive Deep Learning Model Integrated with The Internet of Medical Things (IoMT) for Predicting Heart Disease," Biomedical Signal Processing and Control, vol. 92, 2024.
[15]. P.Brundavani, D Vishnu Vardhan, & B Abdul Raheem, "Ffsgc-Based Classification of Environmental Factors in IOT Sports Education Data during the Covid-19 Pandemic" Journal of Sensors, IoT & Health Sciences, Vol. 2(1), 2024. Pp. 28-54.
[16]. K Ravinder, A. Rajaiah, M. Govindu, & B Abdul Raheem, "A Wireless Voice-controlled Home Automation Framework using Arduino Uno", IOSR Journal of Electronics and Communication Engineering, Vol 21(1), 2026. Pp 01-05.