RESEARCH ARTICLE OPEN ACCESS

# Smart Wound Care: Real-time Monitoring and Predictive Analysis using IoT and ML

Jayaraj N
Assistant Professor
Department of Electronics and Communication Engineering
The Oxford College of Engineering, Bangalore, India
jayarajn0@gmail.com

Pavan Kumar KS
Electronics & communication Engineering
The Oxford College of Engineering
Bangalore, India
pavanreddy8431@gmail.com

Sharan SD
Electronics & communication Engineering
The Oxford College of Engineering
Bangalore, India
sharansdece2026@gmail.com

Vedant Ramesh Nidoni
Electronics & communication Engineering
The Oxford College of Engineering
Bangalore, India
vedantnidoniece2026@gmail.com

Vishwa
Electronics & communication Engineering
The Oxford College of Engineering
Bangalore, India
vishwa0118@gmail.com

----------------------------------------************************----------------------------------------

***Abstract-*** The IOT based Wound Monitor System will monitor and analyze the real time status of wound. The unit will include a combination of sensors such as infection/gas detection sensor MQ3, moisture sensor (used for wound that is already wet, damp but not completely soaked) and temperature/humidity sensor DHT11. The sensors are connected to a microcontroller-ESP-32 for processing. Measurements are streamed up to the ThingSpeak cloud for remote storage and visualization. This data is used to feed an ML model written in python and Flask that takes the parameters of the wound as input and predicts its severity level and provides treatment recommendations. Both healthcare professionals and patients are provided with a web application in which they can check the status of the wound, alerts as well to aid decision-making. The proposed intelligent system will improve wound care management through early onset complication detection and provide personalized treatment suggestions based on data analysis.

**Keywords – IoT, Machine Learning, Wound Monitoring, ESP-32, Sensors, HealthCare**

----------------------------------------************************----------------------------------------

## I. INTRODUCTION

The IoT-Based Wound Monitoring System is a way, in healthcare. The IoT-Based Wound Monitoring System aims to improve wound care and wound healing. The IoT-Based Wound Monitoring System watches wounds all the time. Use data to guide healing. Traditional wound assessment uses inspection. Manual inspection takes time. Manual inspection is not always accurate. The main purpose of the IoT- Based Wound Monitoring System is to use ideas and smart sensors. The IoT-Based Wound Monitoring System automatically estimates the condition of a wound. The system includes sensors. The MQ3 sensor detects. Gases. The moisture sensors check the wetness of the wound. The DHT11 sensor measures temperature and humidity. All the sensors are built into the device that records real-time health data, from the wound environment. The sensor readings are sent wirelessly to the ThingSpeak cloud platform. The ThingSpeak cloud platform lets healthcare professionals view the sensor readings remotely.

I have seen that assessment and wound management matter. Timely wound assessment and wound management help stop infection help speed up healing and help keep costs in the modern health care. Traditional wound monitoring depends on inspection. Traditional wound monitoring often gives results and takes a lot of time and needs health care professionals to be present. Traditional wound monitoring creates a problem for patients who live far from facilities. Patients in areas or, in scattered locations feel the hardest impact. I see the demand for efficient health monitoring is growing. The demand drives the creation of the IoT-based automated wound monitoring system. The IoT-based automated wound monitoring system can track wound conditions in time. The IoT-based automated wound monitoring system can analyze conditions. The IoT-based automated wound monitoring system can predict wound conditions. The IoT-based automated wound monitoring system works without supervision.

I think another driving factor behind this project is the growing importance of data driven health care and the integration of intelligence, in diagnosis. Integration of sensing technology with machine learning helps health professionals make accurate evidence-based decisions. Integration of sensing technology with machine learning can reduce errors. Integration of sensing technology with machine learning can also enable detection of complications and can allow personalized treatment suggestions based on patient specific data. I notice the motivation also comes from the urge for remote healthcare solutions. After the COVID-19 remote healthcare relies on the contactless and digital monitoring tools that keep patients safe and ensure ongoing care.

## II. LITERATURE SURVEY

Karthik et al. [1] developed an IoT-based smart wound monitoring prototype. This device uses temperature and humidity sensors to continuously evaluate the healing environment of a wound. The system sends sensor data to a cloud platform, allowing healthcare professionals to access and visualize the information in real time. Their work highlights the need for ongoing monitoring, less manual oversight, and early infection detection through data insights. The study shows that IoT-enabled monitoring improves patient care by allowing consistent and automatic wound assessments.

Rahman et al. [2] introduced a flexible "smart bandage" with embedded sensors to measure wound temperature, pH level, and moisture content. The bandage transmits data via Bluetooth Low Energy (BLE), making it suitable for wearable and mobile health applications. The authors note that real-time monitoring with these sensors helps clinicians identify abnormal wound conditions early, enabling timely medical interventions. Their research demonstrates the effectiveness of blending flexible electronics and wireless communication for practical wound-care solutions.

Sharma and Patel [3] investigated machine learning techniques to analyze wound-related parameters collected from IoT sensors. Their study includes temperature, humidity, and gas-emission data, and applies classification algorithms like Support Vector Machine (SVM) and Random Forest to predict wound severity and healing progress. The results suggest that ML models significantly improve the accuracy of assessing wound conditions compared to traditional methods. The authors conclude that combining IoT sensing with predictive analytics can aid automated decision-making for chronic wound management.

Mehta and Singh [4] proposed a general IoT-enabled health monitoring framework focused on remote patient care. Their system, based on ESP-32 and cloud connectivity, measures vital parameters like temperature, heart rate, and humidity. While not specifically targeting wound care, their design shows the scalability, real-time accessibility, and alerts possible with IoT in healthcare. The ideas presented lay the groundwork for developing specialized IoT- based health monitoring systems, including smart wound-care applications.

## III. METHODOLOGY

The proposed IoT-Based Wound Monitoring System follows a systematic approach. It integrates sensing, data transmission, machine learning analysis, and visualization for continuous wound assessment. Data acquisition begins with the MQ3 sensor for infection or gas detection, a moisture sensor for checking the wetness of the wound, and a DHT11 sensor for measuring ambient temperature and humidity, all linked to the ESP-32 microcontroller as the main unit. Continuous readings from the sensors, preprocessing of the values, and wireless data transmission to the ThingSpeak cloud platform are made possible using the built-in Wi-Fi module of the ESP-32.

Once uploaded to the cloud, the data is available for remote monitoring and storage. ThingSpeak provides real-time visualization of parameters with dynamic graphs that help healthcare professionals observe changes in wounds over time. The collected data is then imported into a machine learning environment using Python and Flask for further preprocessing and analysis. The ML model, trained on historical wound condition data, classifies the severity of the wound— such as normal, moderate, or critical—based on sensor readings. This intelligent analysis allows for early detection of infection risks and supports timely medical intervention.

Finally, the system's web application serves as the interface for patients and healthcare providers. Built with HTML, CSS, and JavaScript, and connected to the SQLite3 database, it displays real-time wound parameters and shows prediction results along with treatment recommendations. The web interface also sends alerts for abnormal readings, enabling doctors to monitor patient health remotely and make informed decisions. This entire workflow—from sensing to cloud communication, analysis, and user engagement—ensures a reliable, automated, and data- driven process for wound management, significantly improving efficiency and accuracy in traditional wound care.

### *1. Block Diagram*

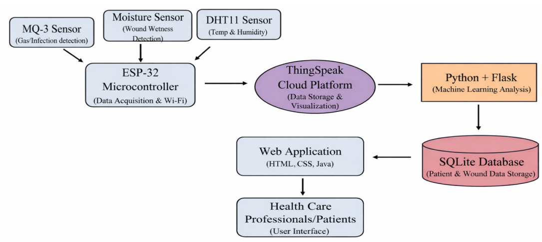



Fig. 1. Block diagram of the System

### *2. Algorithm*

Step 1: Sensor Data Acquisition:The system uses multiple sensors: MQ3 Sensor, detects infection or gas levels from wounds.
Moisture Sensor, monitors the wetness level of the wound.
DHT11 Sensor, measures temperature and humidity. These sensors continuously capture real- time wound conditions.

Step 2: Data Processing via ESP-32: The ESP-32 microcontroller serves as the central unit. It reads sensor data, changes analog signals to digital format, and processes the readings. The ESP-32 connects to Wi-Fi and sends data wirelessly to the ThingSpeak Cloud for further analysis.

Step 3: Cloud Storage & Visualization: ThingSpeak receives and stores sensor data. It provides real-time visualization through dynamic graphs and charts for remote monitoring by
healthcare professionals.

Step 4: Machine Learning Analysis: A Python and Flask-based ML model processes the cloud data. It classifies wound conditions into Normal, Moderate, or Critical levels. Algorithms like Decision Tree or Random Forest are used to predict severity. It offers intelligent insights and early infection alerts.

Step 5: Web Application Development:The application is developed using HTML, CSS, and JavaScript, with SQLite3 as the database. It displays live wound parameters, prediction results, and treatment suggestions. It allows doctors and patients to view wound status and receive automatic notifications.

Step 6: System Outcome:The system achieves continuous wound monitoring, remote accessibility, and early detection of complications. It improves accuracy, reduces manual inspection, and supports personalized treatment decisions.

### 3. *System Requirements*

Hardware Requirements

ESP-32 Microcontroller: Acts as the main processing and communication unit, gathering sensor data, performing signal processing, and sending data to the cloud via Wi-Fi.

MQ3 Sensor: Detects gases or infection- related volatile compounds from the wound area.

Moisture Sensor: Measures moisture levels in wound dressings to assess healing conditions and avoid the buildup of excess fluid.

DHT11 Sensor: Tracks temperature and humidity around the wound to maintain a suitable healing environment.

Power Supply (Battery or Adapter): Provides consistent power to all components and guarantees portability for ongoing monitoring.

Connecting Wires & Breadboard: Facilitates the connection of sensors to the ESP-32 microcontroller.

The dataset includes sensor readings correlating with normal, moderate, and infected conditions. Preprocessing includes normalization and feature extraction. Some of the algorithms used for training comprise Decision Tree and Random Forest. After training, this model is deployed within Flask, where it continuously receives new sensor data from the cloud, makes predictions, and returns the wound status in real time.

The final phase involves creating a web application that serves as the interface for doctors and patients. This application is developed using HTML, CSS, JavaScript, and linked to a SQLite3 database. It displays real-time wound data, prediction results, and recommended treatments. The application includes an alert system that notifies healthcare providers when wound parameter thresholds are exceeded, signalling potential infections or poor healing. The application provides an interactive platform for remote monitoring, enabling timely interventions without the need for frequent hospital visits. Thus, the system's implementation effectively integrates IoT sensing, cloud communication, and machine learning to create an intelligent, automated solution for wound management.

Software Requirements

Arduino IDE: Used for coding and uploading programs to the ESP-32 microcontroller.

ThingSpeak Cloud Platform: A cloud service for storing, visualizing, and analyzing sensor data in real time.

Python (with Flask Framework): Hosts the machine learning model, processes data, and offers backend APIs for predictions and analysis.

SQLite3 Database: A lightweight database for storing patient data, sensor history, and wound severity predictions in the web application.

HTML, CSS, and JavaScript: Technologies used to create the web interface where users can see wound data and get alerts.

Machine Learning Libraries: Includes scikit- learn, NumPy, Pandas, and Matplotlib for training, testing, and analyzing wound condition data.

Operating System: Should be compatible with Windows, Linux, or macOS for development and deployment.

### 4. *Implementation Details*

The implementation of the IoT-Based Wound Monitoring System begins with integrating hardware components to capture real-time wound data. The ESP-32 microcontroller serves as the core unit, linking with various sensors, including the MQ3 sensor for infection or gas detection, the moisture sensor to assess the wetness of the wound, and the DHT11 sensor for measuring temperature and humidity. Each sensor is calibrated to detect minor changes in wound conditions and produce corresponding analog or digital signals. These signals are processed by the on-board ADC of the ESP-32 and converted into readable data. The ESP-32's built-in Wi-Fi module is set up to communicate with the ThingSpeak cloud platform, where sensor readings are continuously sent for storage and visualization.

Upon reaching ThingSpeak, the system performs pattern recognition and anomaly detection. Python and the Flask web framework fetch cloud data and apply a machine learning model to

## IV. RESULTS & DISCUSSION

This section presents the functional outcomes of the proposed wound monitoring system, including its web interface, real-time prediction workflow, machine learning results, cloud-based data visualization, and hardware implementation. All observations demonstrate that the system operates reliably across the IoT layer, analytics layer, and user interface.

### A. *Web Application Results*

1) Home Page Interface

The landing page of the Wound Care AI platform provides a clean and intuitive layout, allowing users to navigate to essential modules such as login, registration, and analysis. The interface highlights the system’s core capabilities, including IoT-based sensor integration, AI-powered wound analysis, and continuous healing progress tracking.

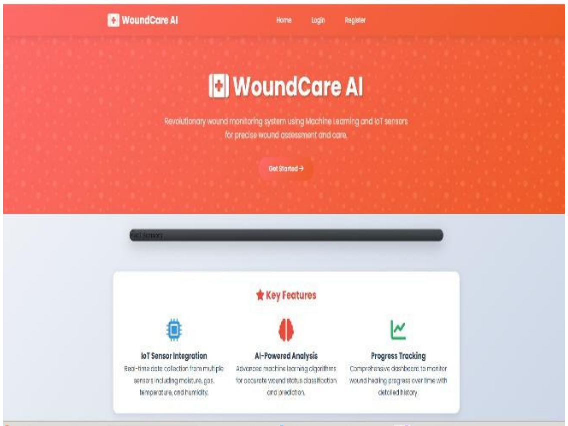



Fig. 2. Home page of the Wound Care AI monitoring system.

2) User Registration/Login Page

The registration module enables new users to securely create an account before accessing the analysis dashboard. Required details such as username, email, and password are captured through a simple form, ensuring accessibility for both patients and healthcare providers.

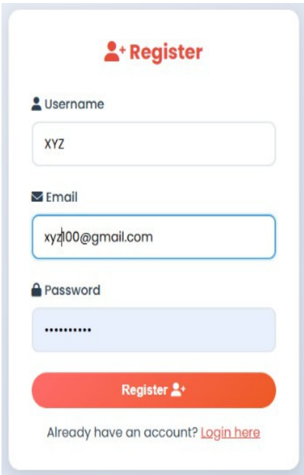



Fig. 3. Registration page for user onboarding.

3) Live Sensor Input and Analysis Page

The system supports both live IoT data and manual input. The interface displays the latest sensor readings—moisture, gas level, temperature, and humidity—along with their expected ranges. Users can request fresh live data from the IoT device or manually enter values for analysis. The design ensures transparency by showing measurement ranges and expected wound-area conditions.

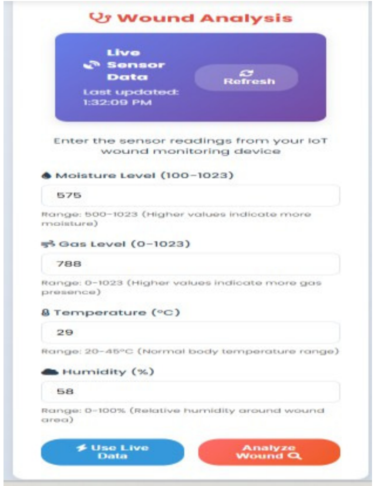
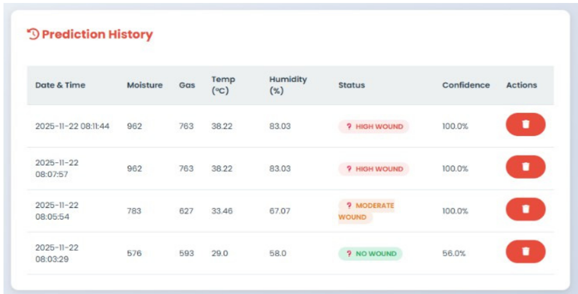



Fig. 4. Wound analysis interface displaying real-time sensor inputs.

4) Prediction History Dashboard

The system automatically stores each prediction along with corresponding sensor values, confidence scores, and wound severity class. This history table provides a traceable record that helps doctors review previous wound status trends and compare healing progress.



Prediction History

| Date & Time | Moisture | Gas | Temp (°C) | Humidity (%) | Status | Confidence | Actions |
|---|---|---|---|---|---|---|---|
| 2025-11-22 08:11:44 | 962 | 763 | 38.22 | 83.03 | HIGH WOUND | 100.0% | |
| 2025-11-22 08:07:57 | 962 | 763 | 38.22 | 83.03 | HIGH WOUND | 100.0% | |
| 2025-11-22 08:05:54 | 783 | 627 | 33.46 | 67.07 | MODERATE WOUND | 100.0% | |
| 2025-11-22 08:03:29 | 576 | 593 | 29.0 | 58.0 | NO WOUND | 56.0% | |

Fig. 5. Prediction history page showing previous wound assessments

## B. *Machine Learning Prediction Outcomes*

The ML model classifies wound severity into three categories—No Wound, Moderate Wound, and High Wound—based on real-time sensor readings. Instead of presenting the three results separately, they are grouped together below as a combined result panel for clarity.

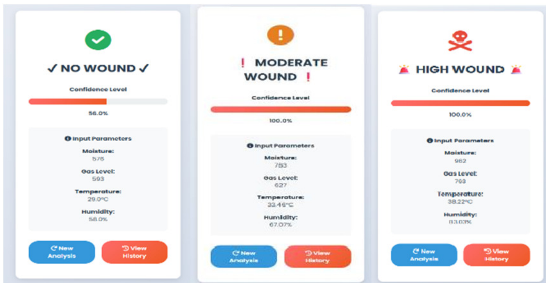



Fig. 6. Combined ML output results showing predictions for No Wound, Moderate Wound, and High Wound conditions

The three outcome screens shown in Fig. 6 represent the model's response to different wound environments:

- No Wound: Lower moisture, controlled gas values, stable temperature, and moderate humidity resulted in a *No Wound* prediction. This indicates that the wound site is dry, clean, and not exhibiting infection-related patterns.

- Moderate Wound: Higher moisture and increased temperature triggered a *Moderate Wound* alert. The model interprets these patterns as signs of early inflammation, prompting users to take precautionary steps before the condition worsens.
- High Wound: Extremely high moisture and gas readings, combined with elevated temperature and humidity, produced a *High Wound* alert. This severity level reflects conditions commonly associated with infection or delayed healing, requiring immediate medical attention.

The model consistently generated confidence values close to 100% for distinct wound profiles, indicating stable classification behavior.

### *C. Cloud-Based Visualization Using ThingSpeak*

All sensor readings from the IoT device are uploaded to ThingSpeak for continuous monitoring. The plotted graphs show how parameters evolve over time, enabling remote observation of environmental variations around the wound.

i. Temperature graph shows fluctuations that can indicate inflammation.
ii. Humidity graph reflects moisture retention around the wound.
iii. Gas sensor readings help identify infection- related volatile compounds.
iv. Moisture graph assists in detecting exudate levels.


Fig. 7. ThingSpeak cloud charts for temperature, humidity, gas values, and moisture levels

### *D. Hardware Prototype Evaluation*

The fully assembled hardware module integrates all required sensors along with the ESP-32 microcontroller and rechargeable battery pack. The compact, wearable-style layout allows the device to be conveniently attached to the wound dressing area. All components were tested under varying environmental conditions and performed consistently, validating the prototype's suitability for real-time monitoring.

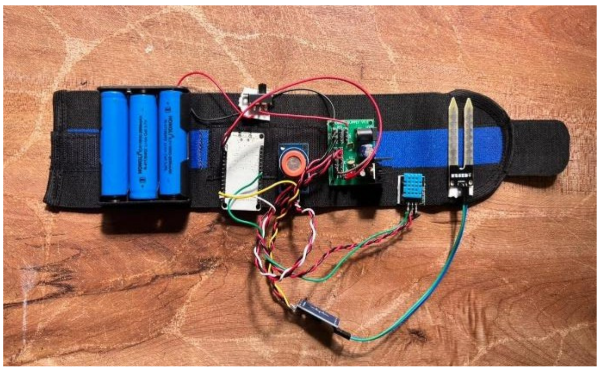


Fig. 8. Assembled IoT hardware prototype used for wound monitoring.

## V. CONCLUSION

The IoT-Based Wound Monitoring System presents an intelligent solution for effective wound care management in real-time and automatically. By integrating the ESP-32 microcontroller with various sensors, including MQ3, Moisture, and DHT11, the system continuously tracks critical parameters such as infection levels, moisture content, and temperature. The wireless transmission of this data to ThingSpeak delivers precise, real-time information about the wound to healthcare providers. The addition of machine learning via Python and Flask enhances the system's ability to analyze data and predict wound severity, supporting timely decisions for better patient outcomes.

This system shifts from traditional manual wound observation to digital healthcare by enabling remote monitoring, early infection detection, and providing data-driven treatment recommendations. The web interface, developed in HTML, CSS, and JavaScript, ensures easy access for patients and doctors, allowing continuous oversight without frequent hospital visits. Overall, the proposed system represents a practical and scalable method for smart healthcare and is expected to encourage future innovations in IoT-based medical monitoring systems that can improve care quality and patient safety.

## VI. REFERENCES

**[1]** R. Karthik, P. Suresh, and V. Kumar, "An IoT-Based Smart Wound Monitoring System for Healthcare Applications," *IEEE Access*, vol. 9, pp. 115233–115245, 2021.

**[2]** M. Rahman, S. Hossain, and L. Ahmed, "Smart Bandage: Real-Time Wound Monitoring Using Sensors and Wireless Communication," *International Journal of Biomedical Engineering*, vol. 15, no. 2, pp. 55–63, 2020.

**[3]** S. Sharma and A. Patel, "Machine Learning- Based Analysis for Chronic Wound Assessment Using IoT Data," *Journal of Medical Systems*, vol. 46, no. 7, pp. 1–12, 2022.

**[4]** D. Mehta and R. Singh, "IoT-Enabled Health Monitoring System for Remote Patient Care," *Journal of Healthcare Informatics Research*, vol. 3, no. 4, pp. 567–580, 2019.

**[5]** S. Kumar, N. Gupta, and A. Das, "Cloud-Based Smart Healthcare Monitoring Using IoT and Machine Learning," in *Advances in Computing and Communication*, Springer, 2021, pp. 301–312.

**[6]** M. G. R. Alam, M. S. Hossain, and A. Almogren, "A Lightweight IoT-Based Smart Healthcare System Using Body Sensor Networks," *IEEE Access*, vol. 7, pp. 185481–185495, 2019.

**[7]** S. Sundar, R. Ganapathy, and B. Prabhu, "Wearable Sensor-Based Early Detection of Wound Infection Using IoT and Deep Learning," *IEEE Sensors Journal*, vol. 21, no. 18, pp. 20714–20723, 2021.

**[8]** X. Fang, Z. Liu, and J. Zhang, "An Intelligent Wound Healing Assessment System Using Thermal and Moisture Sensors," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 4, pp. 1192–1203, 2021.

**[9]** A. M. El-Rawy and H. A. H. Mohamed, "Real-Time Medical Data Monitoring Using IoT and Cloud Computing," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11355–11365, 2021.

**[10]** J. Park, Y. Choi, and K. Lee, "A Hybrid Machine Learning Model for Health Status Prediction Using Multi-Sensor IoT Data," *IEEE Access*, vol. 8, pp. 145808–145820, 2020.