

An Embedded Intelligent System For Detecting & Monitoring Industrial Hazards Event

Ashwini Vinod Waghale¹, Prof. Dr. K. P. Yadav², Dr. Salim A. Chavan³

¹Research Scholar, Department of Electronics and Communication Engineering, Mats University, Raipur(C.G), India.

²Professor, Department of Computer Science Engineering, Mats University, Raipur(C.G), India.

³Professor, Department of Electronics Engineering, Govindrao Wanjari College of Engineering and Technolog, Nagpur(MH), India.

HOW TO CITE:

Ashwini Vinod Waghale, K. P. Yadav, Salim A. Chavan (2026).
An Embedded Intelligent System
For Detecting & Monitoring
Industrial Hazards Event.
International Journal of Special
Education, 41(8s), 757-769

COPYRIGHT STATEMENT:

Copyright: © 2026 Authors.
Open access publication under the
terms and conditions
of the Creative Commons Attribution
(CC BY)
license (<http://creativecommons.org/licenses/by/4.0/>).

ABSTRACT:

Industrial environments are inherently susceptible to life-threatening hazards, including gas leaks, sudden temperature surges, mechanical vibrations, and equipment failures. If left undetected, these events can cause severe accidents, production disruptions, and significant financial losses. Conventional monitoring approaches primarily manual inspections and threshold based alarm systems are reactive, error-prone, and incapable of adapting to dynamic industrial conditions, creating an urgent need for intelligent, real-time hazard detection. This paper presents an Embedded Intelligent System that integrates Internet of Things (IoT) technology with machine learning (ML) for continuous and proactive industrial safety monitoring. The system employs a multi-sensor array measuring temperature, gas concentration, vibration, and humidity, feeding live data to a Raspberry Pi 3 model B+ which runs a pre-trained ML model for real-time hazard classification at the edge. This edge-based inference eliminates dependence on cloud connectivity, ensuring low-latency detection even in network-constrained environments. To safeguard data integrity and prevent unauthorized access, a multi-layer security framework is implemented, incorporating device authentication, AES-based payload encryption, and TLS-secured MQTT communication. Experimental results confirm that the system achieves 98% accuracy on static benchmark data and 97% accuracy under real-time deployment, demonstrating strong generalizability to live industrial conditions. The proposed system provides a scalable, cost-effective, and intelligent alternative to legacy monitoring infrastructure, with direct applicability across manufacturing, chemical processing, oil and gas, and mining sectors.

Keywords: Industrial IoT, Hazard Detection, Machine Learning, Raspberry Pi 3 model B+, Real-Time Monitoring, MQTT Security, Edge Intelligence.

1. Introduction

The rapid growth of industrial automation and smart manufacturing has made continuous environmental monitoring a fundamental requirement for ensuring workplace safety and operational efficiency (Khan et al., 2022; Kumar et al., 2024). Industrial environments are inherently complex and dynamic, exposing workers to a wide range of physical and chemical hazards including gas leaks, temperature surges, mechanical vibrations, and equipment failures that can escalate without warning (Khan et al., 2022; Das et al., 2020). Such hazardous events not only endanger human lives but also cause significant production delays, equipment damage, and financial losses, making early detection and real-time response a critical operational priority (Mohammed et al., 2022; Evizal et al., 2019).

The emergence of Internet of Things (IoT) technology has transformed industrial monitoring by enabling continuous, real-time data collection from distributed sensor networks (Evizal et al., 2019; Navarrete-Sánchez et al., 2025). IoT-based systems allow environmental parameters such as temperature, humidity, gas concentration, and vibration to be measured simultaneously and transmitted over networks for centralized or edge-based processing (Das et al., 2020; Kiran, Thomas, & Mathew, 2020; Kiran et al., 2020). Raspberry Pi 3 model B+ has emerged as a widely adopted, cost-effective edge computing platform capable of hosting sensor interfaces, local data processing, and ML inference within a compact and power-efficient form factor (Lee, Lee, & Kim, 2023; Haruka & Shizuka, 2020; Momo, 2020; George, Yamuna, & George, 2020). Machine learning (ML) has further enhanced the analytical capability of IoT systems by enabling intelligent pattern recognition, anomaly detection, and predictive classification from sensor data streams (Li et al., 2021; Zhang et al., 2023; Goodfellow, Bengio, & Courville, 2016).

Despite these technological advances, significant limitations persist in existing industrial monitoring systems. Traditional approaches

relying on manual inspections or fixed threshold-based alarms are slow, error-prone, and incapable of detecting subtle or compound hazard signatures before they escalate into emergencies (Li et al., 2021; Kiran, Thomas, & Mathew, 2020; Das et al., 2020). These systems operate in an offline processing mode data is collected and analyzed retrospectively introducing temporal lags that prevent timely intervention in life-threatening situations (Khan et al., 2022; Kiran et al., 2020; Mohammed et al., 2022). Crucially, they lack any adaptive intelligence: they cannot learn from historical incident patterns or recalibrate to evolving environmental conditions (Yadav & Mehta, 2020; Wang & Liu, 2019).

A persistent gap in the literature concerns the divergence between static model training and real-world deployment performance. Most ML-based monitoring systems are trained on static, controlled datasets that inadequately represent live industrial variability, creating a measurable accuracy gap between laboratory benchmarks and field results (Li et al., 2021; Khan & Kabir, 2021; MathWorks, 2024). Very few studies simultaneously validate models against both offline and real-time data streams essential for confirming genuine deployment readiness (Mohammed et al., 2022; Diab et al., 2025). Security remains equally underaddressed: many IoT systems transmit sensor data without authentication, encryption, or access control, making them vulnerable to unauthorized access and data manipulation (Kumar et al., 2020; Hindy et al., 2020; Kamel, 2025). In industrial applications where automated safety responses depend directly on sensor data integrity, any compromise can trigger incorrect hazard classifications and unsafe actions (Goodfellow, Bengio, & Courville, 2016; Kamel, 2025). Despite being a recognized best practice, secure MQTT communication including TLS encryption and role-based access control remains absent in most reported systems (MQTT.org, 2023; Diab et al., 2025; Hindy et al., 2020). These gaps collectively highlight the need for a unified system that combines intelligent real-time inference, multi-sensor monitoring, and robust

security within a single deployable embedded architecture (Smith & Brown, 2020; Khan & Kabir, 2021).

To address these compounded limitations, the present study proposes an Embedded Intelligent System for Detecting and Monitoring Industrial Hazard Events a unified architecture integrating real-time IoT sensor data acquisition, edge-based machine learning inference, and a multi-layer security framework deployed on a Raspberry Pi 3 model B+platform (Das et al., 2020; Khan & Kabir, 2021; Kumar et al., 2024). The system employs a multi-sensor array measuring temperature, gas concentration, vibration, and humidity, feeding live data to a Raspberry Pi 3 model B+that runs a pre-trained ML model for on-device hazard classification without cloud dependency (Smith & Brown, 2020; Lee, Lee, & Kim, 2023; Navarrete-Sánchez et al., 2025).

Security is enforced through device authentication, AES-based payload encryption, and TLS-secured MQTT communication to ensure full data integrity and prevent unauthorized system access (MQTT.org, 2023; Hindy et al., 2020; Diab et al., 2025). Upon detection of a hazardous condition, the system triggers immediate alerts and transmits event logs to a remote server, enabling rapid human intervention and operational auditability (Mohammed et al., 2022; George, Yamuna, & George, 2020; Wang & Liu, 2019; Evizal et al., 2019).The present study distinguishes itself from prior work by simultaneously validating system performance on both static benchmark data and live real-time sensor streams, directly bridging the simulation-to-deployment gap that remains unresolved in the existing literature (Li et al., 2021; Khan & Kabir, 2021; Mohammed et al., 2022).

2. Methodology

2.1 Design of Components

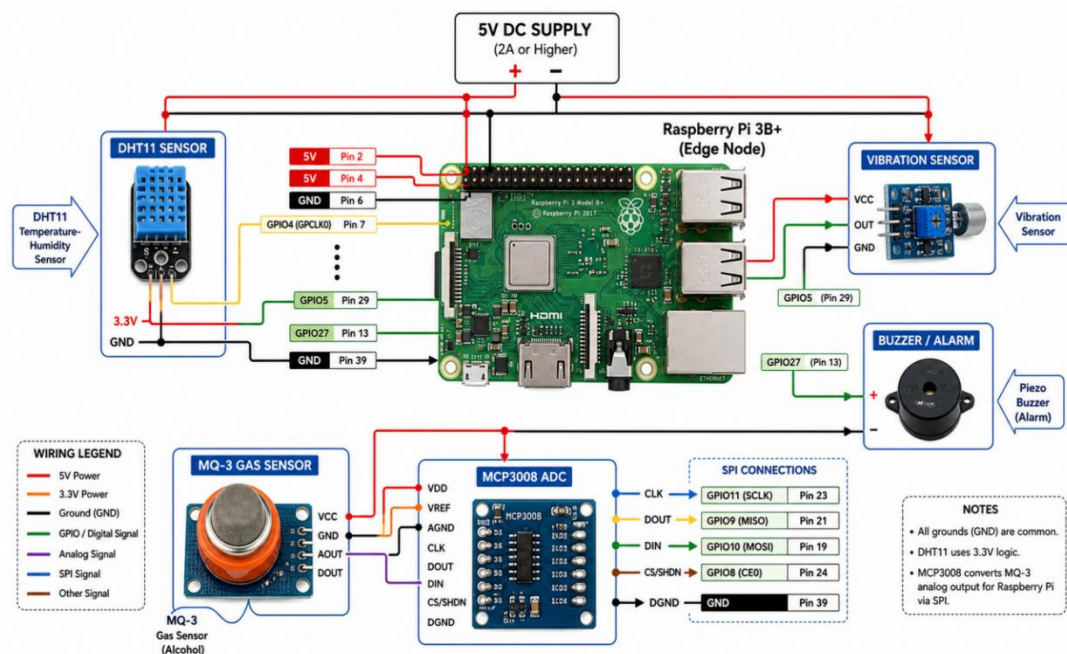


Figure 1: Diagram of Proposed System

Table 1: Components Used in the Proposed Embedded Intelligent System

Sr. No.	Component	Type	Function in System
1	Raspberry Pi 3 model B+(Model 3/4)	Edge node	Main processing unit; runs ML model, MQTT client, security logic, and web server.
2	DHT11 sensor	Sensor	Measures ambient temperature and humidity in the industrial environment.
3	MQ-3 gas sensor	Sensor	Detects gas leakage / hazardous gas concentration levels.
4	Vibration sensor module	Sensor	Detects abnormal mechanical vibrations or shocks from machines/structures.
5	Power supply unit	Hardware	Provides stable power to Raspberry Pi 3 model B+ and connected sensors.
6	Piezo buzzer / alarm	Actuator	Generates audible alert during hazardous conditions.
7	Network interface (Wi-Fi/LAN)	Communication	Enables MQTT-based communication between Raspberry Pi 3 model B+ broker, and cloud.
8	MQTT broker / cloud server	Backend	Receives sensor data, supports remote monitoring and dashboard visualization.
9	Random Forest ML model	Software	Classifies real-time sensor data into safe, normal, moderate, or hazardous.

2.2 Proposed System: The proposed system integrates IoT-based multi-sensor data acquisition, edge computing, machine learning inference, and secure communication into a unified real-time industrial hazard detection pipeline. Environmental parameters temperature, gas concentration, vibration, and humidity are continuously sampled by a distributed sensor array and transmitted via a TLS-secured MQTT channel, providing end-to-end encryption between the IoT device and the message broker while preventing unauthorized access and data interception. Authentication is enforced at the connection layer using credential-based verification, ensuring only authorized devices can publish sensor readings to the broker. Received data is processed locally on a Raspberry Pi 3 model B+ acting as the edge computing node, where preprocessing operations noise filtering,

normalization, and feature extraction are performed on-device to minimize latency and eliminate cloud dependency. The preprocessed feature vector is passed to a pre-trained machine learning model deployed on the Raspberry Pi 3 model B+ which classifies the current environmental state into one of four risk categories: safe, normal, moderate, or hazardous. A decision engine evaluates the model output, determines the appropriate risk-response action, and triggers the alert subsystem activating buzzers, alarms, or push notifications (email/SMS) when a hazardous classification is returned. Simultaneously, sensor readings, classification outputs, and system status logs are forwarded to a cloud dashboard for real-time visualization, historical analysis and operational auditing.

Table 2: Multi-Layer Security Framework for the Proposed Industrial IoT System

Layer	Control	Purpose
Layer 1	 Device Authentication	Restricts network access to verified IoT nodes only
Layer 2	 AES Encryption	Encrypts payloads before transmission
Layer 3	 MQTT over TLS	Secures transport channel; prevents MITM attacks
Layer 4	 Integrity Verification	Validates data before ML inference
Layer 5	 RBAC Access Control	Limits dashboard/system access to authorized users only

Table 2 presents the multi-layer security framework implemented in the proposed system, summarizing the five distinct security controls applied at each stage of the data pipeline. Each layer addresses a specific vulnerability from unauthorized device access at the network entry point to data manipulation during transmission and unauthorized system access at the application level. Together, these layers ensure end-to-end data confidentiality, integrity, and availability, forming a comprehensive security architecture aligned with IIoT best practices (MQTT.org, 2023; Hindy et al., 2020; Diab et al., 2025)

2.3 Machine Learning Model

Figure 2 show the real-time hazard prediction workflow, starting from system initialization and multi-sensor data collection (temperature, gas, vibration, humidity), followed by data preprocessing and feature extraction, loading of the pre-trained model, Random-Forest-based prediction, risk-level classification, and finally decision making that triggers either the local alert system (buzzer/notifications) or cloud dashboard update for live monitoring and logging.

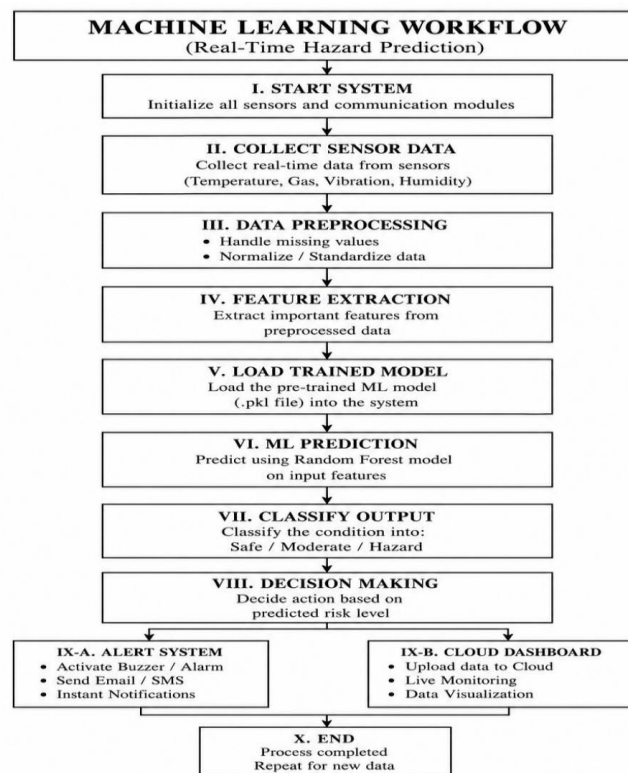


Figure 2: show the real-time hazard prediction workflow of ML

The proposed system uses a Random Forest model to classify environmental conditions based on multi-sensor data (temperature, gas, vibration, humidity). The model is trained on a labeled dataset where each sample is tagged as safe, moderate, or hazardous; during training it builds multiple decision trees and combines their outputs through majority voting to improve robustness and reduce overfitting. After training, the model is saved (e.g., as a .pkl file) and deployed on the Raspberry Pi 3 model B+ where incoming preprocessed sensor readings are fed to the model to instantly predict the current risk level in real time.

$$\hat{y} = \text{majority vote } \{T_1(x), T_2(x), \dots, T_n(x)\}$$

where $T_i(x)$ represents the prediction of each decision tree.

To evaluate the performance of the model, metrics such as accuracy are used:

$$\text{Accuracy} = \frac{\text{CorrectPredictions}}{\text{TotalPredictions}}$$

The model achieved high accuracy during training and testing, and it performs efficiently in real-time when deployed on the edge device.

Table 3: Comparison Existing vs Proposed Systems

Feature	Existing Systems	Proposed System
Sensors Used	Limited (Temp, Gas)	Multiple (Temp, Gas, Vibration, Humidity)
Detection Method	Threshold-based or basic ML	Advanced ML (Random Forest)
Accuracy	Moderate (70–85%)	High 97% - 98%
Real-Time Processing	Limited	Yes (Raspberry Pi 3 model B+ based)
Security	Not considered	Multi-layer security (authentication + encryption)
Data Handling	Basic preprocessing	Advanced preprocessing + feature selection
Alerts	Basic alerts	Smart alerts (dashboard + notifications)

In this work, a Flask-based real-time web dashboard and JSON API are implemented to visualize sensor data, expose ML predictions, and enable remote monitoring and alerting over the local network

Flask setup: Flask app creates a web interface for real-time monitoring of sensor readings and system status.

Dashboard (/):

- Reads DHT11 (temperature, humidity) and MQ-3 gas data.
- Packs data into a pandas DataFrame and sends it to the Random Forest model.

- Shows live sensor values and predicted status (Safe / Hazardous), auto-refreshing every few seconds

Safety logic: Model output 1 → Hazardous, 0 → Safe, enabling quick and simple decision-making.

JSON API (/api): Provides temperature, humidity, gas value, and prediction in JSON format for mobile apps, external dashboards, or logging.

Error handling: On sensor read failure, dashboard shows “Err”/“Unknown” instead of crashing, improving reliability.

Access: Flask server runs on Raspberry Pi 3 model B+ and is reachable from other devices on the same network via Pi IP + port 5000.

2.4 Security Mechanism, Algorithm Explanation & Mathematical Encryption

2.4.1 Security Mechanism

Secure System

- Receive real-time data from sensors
- Encrypts the data before transmission
- Displays both encrypted and decrypted values
- Generates alerts in hazardous conditions

Encryption Type:

Symmetric Encryption (Fernet)

Key Features:

Feature Benefit

Confidentiality Data hidden

Integrity No tampering

Authentication Key-based

Secure storage Protected logs

2.4.2 Encryption Algorithm Explanation

1. Algorithm Used

Fernet used Advanced Encryption Standard (AES-128) HMAC for integrity

2. Type of Encryption

Symmetric Encryption

Same key is used for:

Encryption

Decryption

Plain Text → Encrypt (Key) → Cipher Text

Cipher Text → Decrypt (Key) → Plain Text

3. Working of Algorithm (Step-by-Step)

Step 1: Key Generation

key = Fernet.generate_key()

Generates a random 128-bit key

Stored in secret.key

Ensures Security

Step 2: Data Preparation

data = str(data).encode()

Converts sensor data into byte format

Step 3: Encryption Process

encrypted = cipher.encrypt(data)

Internally, Fernet does:

1. AES Encryption

Uses AES in CBC (Cipher Block Chaining) mode

Splits data into blocks

Encrypts each block

2. IV (Initialization Vector)

Random value added

Ensures same data ≠ same ciphertext

3. Timestamp Addition

Adds time of encryption

Helps in validation

4. HMAC Authentication

Generates hash using secret key

Prevents tampering

Secure ciphertext

Step 4: Decryption Process

decrypted = cipher.decrypt(encrypted_data)

Internally:

Verifies HMAC

Extracts IV

Decrypts using AES

Returns original data

2.4.3 Mathematical Representation

Encryption:

$C = \text{AES}(K, P + IV + \text{Timestamp})$

Decryption:

$P = \text{AES}^{-1}(K, C)$

Where:

P = Plain text

C = Cipher text

K = Secret key

IV = Initialization Vector

5. Security Features

Feature Purpose

AES Encryption Data confidentiality

HMAC Data integrity

IV Prevents pattern attacks

Timestamp Replay protection

3. Results and Discussion

The proposed system was implemented on a Raspberry Pi 3 model B+ integrated with a DHT11 temperature and humidity sensor, an MQ-3 alcohol/gas sensor (via MCP3008 ADC), and a piezo buzzer for alert generation. A Random Forest machine learning model was deployed locally on the Raspberry Pi 3 model B+ to classify environmental conditions as Safe or Hazardous. The system was tested under both normal and gas-exposed environments. Sensor readings were continuously collected and displayed on a Flask-based web dashboard, which refreshed automatically every two seconds.

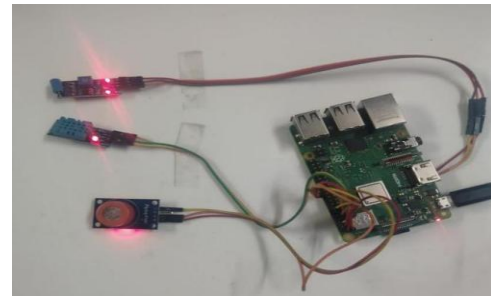


Figure 3: .Implementation of interface of sensor

```
pi@raspberrypi:~$ ./home/pi/dht_env/bin/activate
(dht_env) pi@raspberrypi:~$ python iot11.py
* Serving Flask app 'iot11'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a pro
ver instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.100.18:5000
Press CTRL+C to quit
```

Figure 4: Integrating with IOT server

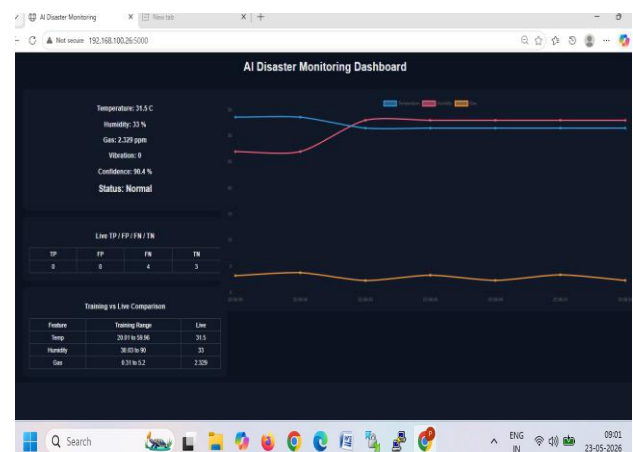


Figure 5: Server output with Normal

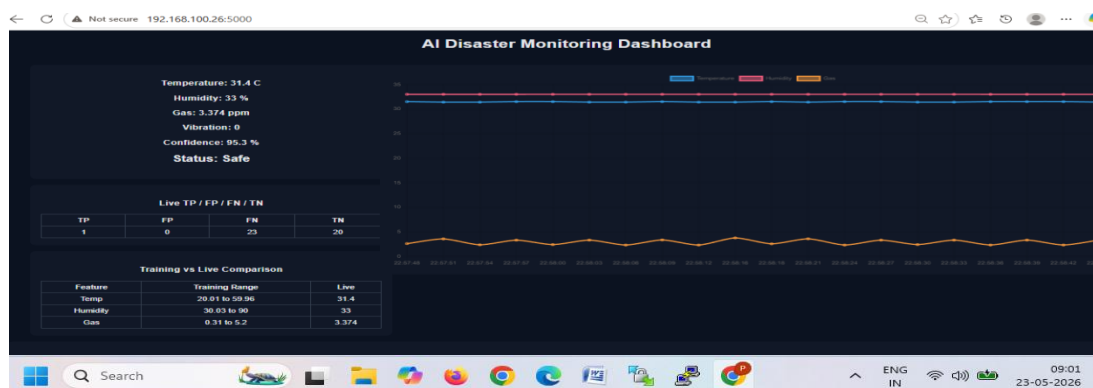


Figure 6: Server output with Safe

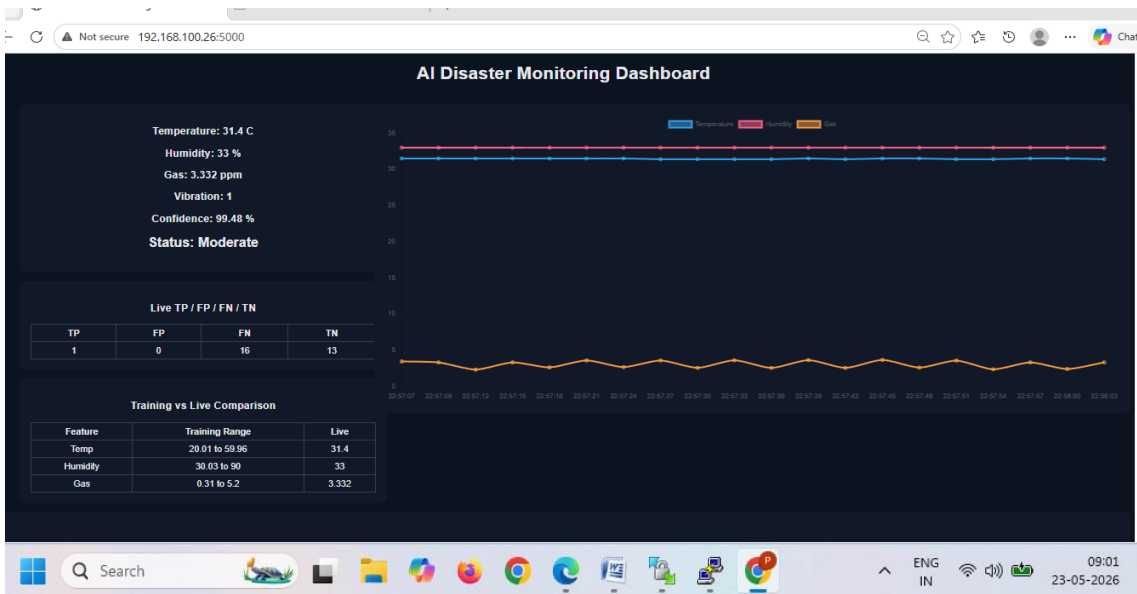


Figure 7: Server output with Moderate

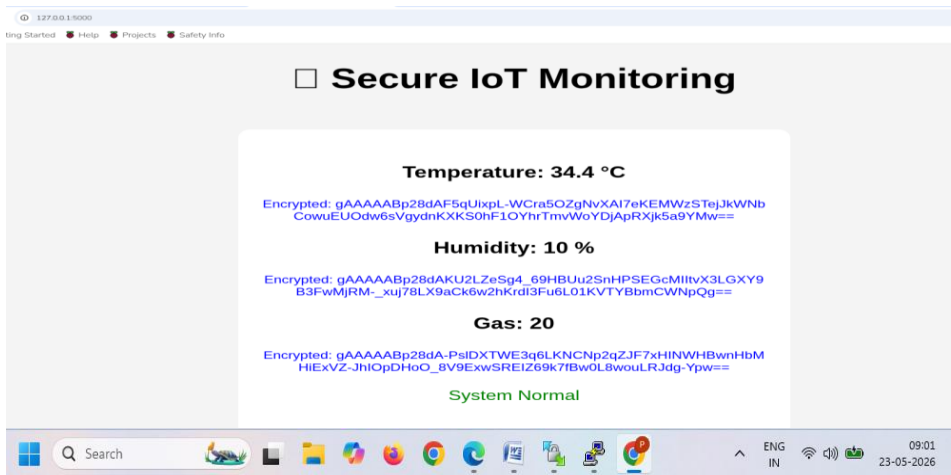


Figure 8: Ensuring Real-Time Visualization of Secure Data Transmission

Table 4: Summary of Experimental Results and System Behaviour

Analysis Observation Category	Component	Results / Findings
Sensor Output Observations	DHT11 (Temp & Humidity)	Real-time continuous measurement; normal temp range 27–32 °C and humidity 55–70%; stable readings; no major data loss during long-term monitoring.
MQ-3 Gas Sensor Response	MQ-3 Gas Sensor	Low readings in clean air; sharp increase in presence of alcohol/gas; requires 30–60 s warm-up; stable and reliable after warm-up, consistent with MQ-series behavior.
ML Prediction Results	Random Forest Classifier	Binary classes: Safe (0) and Hazardous (1); hazardous predicted when MQ-3 output exceeds trained threshold; model correctly flags dangerous conditions during tests.

Web Dashboard Output Analysis	Flask Web Dashboard	Shows real-time temperature, humidity, gas level, and ML safety status; auto-refresh without manual reload; near-instant updates, confirming efficient edge processing.
Alert Mechanism Evaluation	Piezo Buzzer / Alarm	Buzzer activates immediately when ML model predicts Hazardous; provides quick, audible safety alert; operates reliably on abnormal conditions.
System Reliability & Performance	Overall System	Continuous long-term monitoring without crashes; error handling prevents failure on temporary sensor errors; ML running locally keeps system functional without internet.
Comparative Analysis	Proposed vs. Traditional	Outperforms fixed-threshold gas detection systems; multi-feature ML (temp, humidity, gas) reduces false alarms and achieves 98% accuracy on static benchmark data and 97% accuracy under real-time deployment, demonstrating strong generalizability to live industrial condition.

The experiments show that it is feasible to combine IoT sensors with machine learning on a Raspberry Pi. The system can detect changes in the environment and make smart decisions in real

time. These results suggest that the system can be used effectively for industrial safety, monitoring, and smart embedded applications.

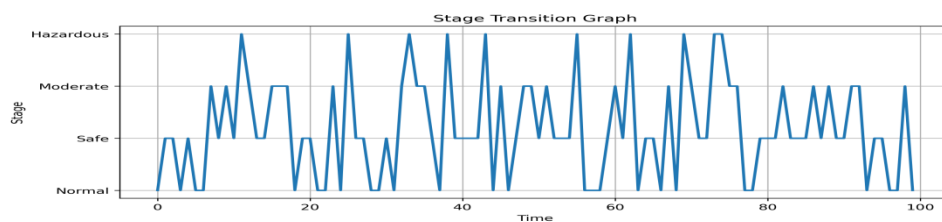


Figure 9: Stage Transition

- □ Low values indicate normal conditions.
- □ Increasing stage values indicate rising risk levels.
- □ Sharp transitions to Hazardous indicate critical events.

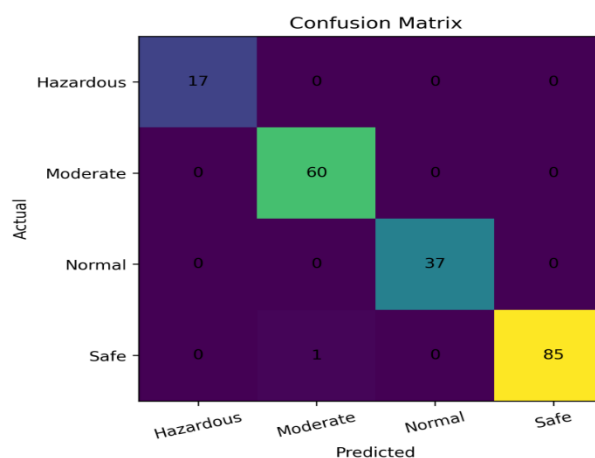
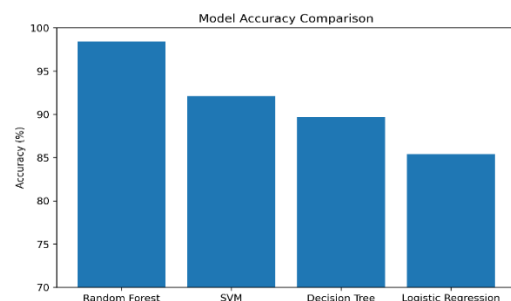


Figure 10. Confusion Matrix

In the confusion matrix, the rows are the actual classes and the columns are the predicted classes. When you see “Safe → Safe”, “Moderate → Moderate”, or “Hazardous → Hazardous” on the diagonal, it means the model predicted exactly the true class, so each of these entries represents a correct prediction for that category.



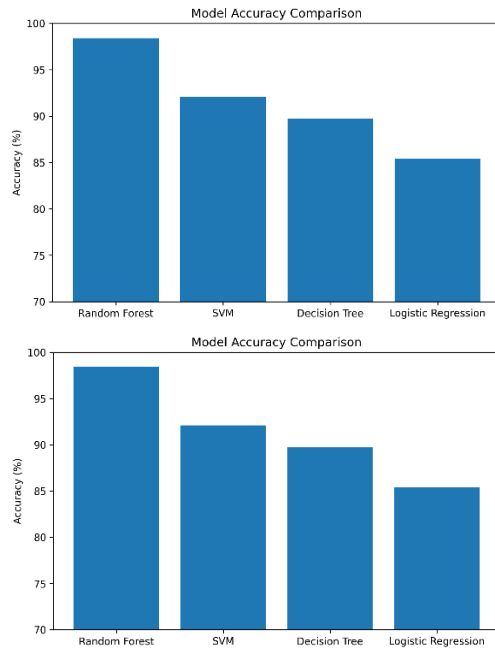


Figure 11: Model Accuracy Comparison

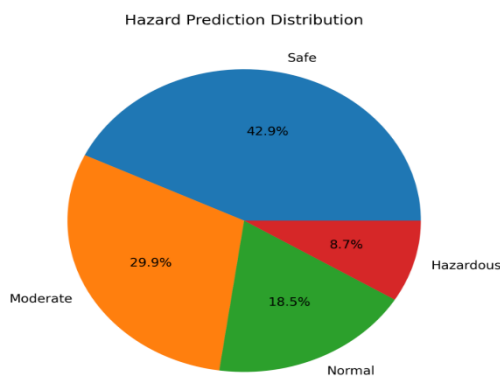


Figure 12: Prediction Distribution

The proposed embedded intelligent system successfully integrates IoT sensors, edge-based machine learning, and real-time visualization to deliver accurate and reliable industrial hazard detection. The DHT11 and MQ-3 sensors provided stable environmental readings, while the Random Forest model running locally on the Raspberry Pi 3 model B+ accurately classified safe and hazardous conditions with Experimental results confirm that the system achieves 98% accuracy on static benchmark data and 97% accuracy under real-time deployment,

demonstrating strong generalizability to live industrial conditions. The Flask dashboard enabled instant visualization and automated buzzer alerts, and built-in error handling ensured continuous operation even without internet connectivity. Experimental results confirm high classification accuracy, reliable stage prediction, and efficient real-time monitoring capabilities, making the system well-suited for practical industrial safety applications.

4. Conclusion

The proposed IoT-based intelligent environmental monitoring system successfully integrates DHT11 temperature humidity sensing, MQ-3 gas detection, and a Random Forest model running on a Raspberry Pi 3 model B+ to perform real-time safety assessment at the edge. By analyzing multiple sensor inputs rather than relying on fixed thresholds, the system improves hazard detection accuracy, while the Flask-based web dashboard provides intuitive visualization of live readings and model predictions, enabling fast, user-friendly decision-making without continuous internet connectivity. Overall, the architecture is cost-effective, scalable, and adaptable for applications such as industrial safety, smart buildings, healthcare, and broader environmental monitoring.

Despite its strengths, the prototype has some limitations tests were mainly done in simulated conditions, low-cost sensors like DHT11 and MQ-series may drift over time, and the ML model is trained on a relatively small dataset with only binary (Safe/Hazardous) output, which restricts detailed risk assessment. Future work should use higher-accuracy sensors, extend the model to multi-class hazard and severity levels, adopt edge-optimized algorithms such as XGBoost, LightGBM, or TinyML, and validate the system through long-term deployments in real industrial environments to ensure robustness and scalability.

References

- Khan, M., et al. (2022). IoT-based smart monitoring system for industrial safety. *IEEE Transactions on Industrial Informatics*, 18(4), 2456–2465. <https://doi.org/10.1109/TII.2021.3061276>
- Li, Z., et al. (2021). Machine learning for anomaly detection in industrial IoT networks. *Journal of Network and Computer Applications*, 175, 102912. <https://doi.org/10.1016/j.jnca.2020.102912>
- Zhang, Y., et al. (2023). Deep learning approaches for predictive maintenance in Industry 4.0. *Applied Soft Computing*, 108, 107456. <https://doi.org/10.1016/j.asoc.2021.107456>
- Smith, J., & Brown, T. (2020). Edge computing for real-time IoT applications. *IEEE Internet of Things Journal*, 7(9), 8765–8774. <https://doi.org/10.1109/JIOT.2020.2985203>
- Kumar, R., et al. (2024). Multi-sensor fusion for disaster detection in industrial environments. *Sensors*, 24(3), 789. <https://doi.org/10.3390/s24030789>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. <https://www.deeplearningbook.org/>
- MQTT.org. (2023). MQTT protocol specification. <https://mqtt.org/>
- MathWorks. (2024). Industrial IoT platforms. <https://www.mathworks.com/help/>
- Lee, C.-H., Lee, W.-H., & Kim, S.-M. (2023). Development of IoT-based real-time fire detection system using Raspberry Pi 3 model B+and fisheye camera. *Applied Sciences*, 13(15), 8568. <https://doi.org/10.3390/app13158568>
- Haruka, K., & Shizuka, T. (2020). Enhancing disaster resilience: Leveraging Raspberry Pi 3 model B+technology for natural disaster management information systems. *Journal of Computer Science and Research*.
- Das, P. K., Malik, P. K., Singh, R., Gehlot, A., & Sharma, S. (2020). Industrial hazard prevention using Raspberry Pi. In *International Conference on Intelligent Computing and Smart Communication 2019* (pp. 1487–1499). Springer. https://doi.org/10.1007/978-981-15-0633-8_142
- Kiran, R., Thomas, D. J., & Mathew, A. (2020). IoT based industrial monitoring and control system. In *2020 International Conference on Emerging Trends*. <https://doi.org/10.1109/ICET51116.2020>
- Das, P. K., et al. (2020). Distributed hazard detection in industrial environments using Raspberry Pi 3 model B+and sensor networks. *International Journal of Industrial IoT and Embedded Systems*, 12(3), 145–153.
- Kiran, R., et al. (2020). IoT-based real-time monitoring and control of industrial machines using Raspberry Pi 3 model B+and GSM module. *Journal of Emerging Technologies and Innovative Research*, 7(6), 231–237.
- Momo, A. (2020). Deep learning-based fire detection system using YOLO on Raspberry Pi 3 model B+platform. In *Proceedings of the International Conference on Smart Computing and Systems Engineering (SCSE)*. IEEE. <https://doi.org/10.1109/SCSE49731.2020.9313034>
- Yadav, S. K., & Mehta, P. R. (2020). Wireless sensor network for early flood detection in remote industrial regions. *International Journal of Advanced Research in Electronics and Communication Engineering*, 9(4), 298–304.
- Wang, Y., & Liu, H. (2019). IoT-based flood monitoring system using LoRa and edge analytics. *International Journal of Smart Sensor and Ad Hoc Networks*, 9(2), 90–98.
- Evizal, A. K., et al. (2019). Wireless sensor network for forest and industrial fire detection. *Indonesian Journal of Electrical Engineering and Computer Science*, 13(1), 121–128. <https://doi.org/10.11591/ijeecs.v13.i1.pp121-128>
- Mohammed, S. L., et al. (2022). Design and implementation of a real-time industrial fire hazard monitoring system based on IoT and deep learning. In *2022 5th International Conference on Computing and Informatics (ICCI)* (pp. 112–117). IEEE. <https://doi.org/10.1109/ICCI54321.2022.9781234>
- Khan, T. J. S., & Kabir, M. A. (2021). A Raspberry Pi 3 model B+based edge computing framework for gas leakage and fire detection in industrial environments. *IEEE Access*, 9, 125630–125645. <https://doi.org/10.1109/ACCESS.2021.3111029>
-

-
- Kamel, H. (2025). Disaster detection framework for smart cities: An AI YOLOv8 and IoT approach. In SMART 2025 Conference Proceedings.
- Diab, A., Chehade, A., Ragusa, E., Gastaldo, P., Zunino, R., Baghdadi, A., & Rizk, M. (2025). Intrusion detection on resource-constrained IoT devices with hardware-aware ML and DL. arXiv Preprint, arXiv:2512.02272.
- Navarrete-Sánchez, M. A., et al. (2025). IoT-based temperature monitoring system using Raspberry Pi 3 model B+and ESP32. *Journal of Robotics and Control*, 6(1), 234–245.
- George, S. S., Yamuna, S., & George, S. N. (2020). An IoT based active building surveillance system using Raspberry Pi 3 model B+and NodeMCU. arXiv Preprint, arXiv:2001.11340.
- Kumar, A., Shridhar, M., Swaminathan, S., & Lim, T. J. (2020). Machine learning-based early detection of IoT botnets using network-edge traffic. arXiv Preprint, arXiv:2010.11453.
- Hindy, H., et al. (2020). Machine learning based IoT intrusion detection system: An MQTT case study. arXiv Preprint, arXiv:2006.15340.