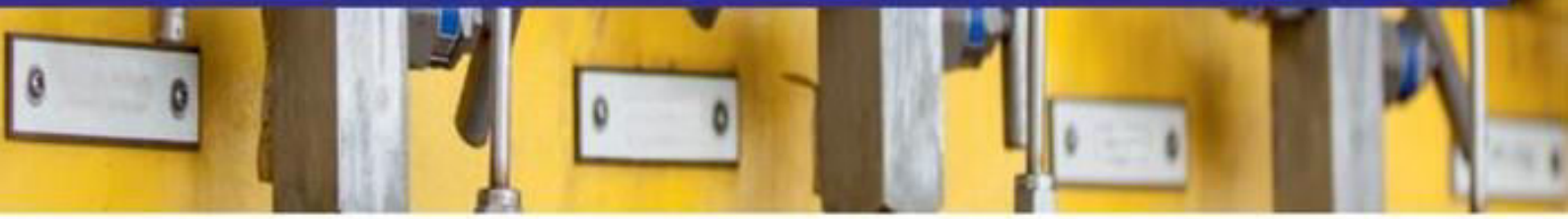
IJAREEIE

e-ISSN: 2278-8875
p-ISSN: 2320-3765

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

Volume 15, Issue 7, July 2026



ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 9.867

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 9.867 | A Monthly Peer Reviewed & Refereed Journal |

|| Volume 15, Issue 7, July 2026 ||

| DOI:10.15662/IJAREEIE.2026.1507007 |

An Intelligent Multi-Sensor IoT Edge Computing System for Industrial Disaster Detection using Machine Learning

Ashwini Vinod Waghale[1], Dr. K. P. Yadav[2], Dr. Salim A.Chavan[3]

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

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

Professor, Department of Electronics Engineering, Govindrao Wanjari College of Engineering and Technology, Nagpur (MH), India[3]

ABSTRACT: Industrial environments are highly susceptible to hazards such as gas leakage, excessive temperature, abnormal humidity, equipment vibration, and fire-related incidents, which demand rapid and intelligent monitoring systems. This paper presents An Intelligent Multi-Sensor IoT Edge Computing System for Industrial Disaster Detection Using Machine Learning, which integrates Internet of Things (IoT), edge computing, and machine learning for real-time industrial safety monitoring. The proposed system employs a Raspberry Pi 3 Model B+ as an edge computing device interfaced with a DHT11 temperature-humidity sensor, an MQ-3 gas sensor, and a vibration sensor to continuously monitor environmental and mechanical conditions. Sensor data are processed locally using a Random Forest classifier, enabling low-latency hazard prediction without relying on cloud infrastructure. To enhance the robustness and generalization capability of the prediction model, a publicly available Kaggle industrial hazard dataset containing 1000 labeled samples was used for training and validation, with an 80:20 train-test split. Experimental evaluation demonstrates that the proposed model achieved 98.75% training accuracy and 98.50% testing accuracy, with 98.52% precision, 98.50% recall, and an F1-score of 98.50%. Three-fold cross-validation yielded an average accuracy of 97.00%, while the confusion matrix confirmed reliable classification of four industrial safety states: Normal, Safe, Moderate, and Hazardous. Feature importance analysis identified temperature and humidity as the most influential parameters, followed by gas concentration and vibration. A Flask-based web dashboard provides real-time visualization of sensor readings and system status, while MQTT-based communication enables remote monitoring and alert generation. The proposed edge-enabled intelligent monitoring framework offers a cost-effective, scalable, and highly accurate solution for Industry 4.0 applications, smart factories, chemical plants, and other safety-critical industrial environments.

KEYWORDS: IoT, Edge Computing,ML, Disaster Detection, Random Forest, Encryption, Industrial Safety, Sensor Networks, Real-Time Monitoring.

I. INTRODUCTION

Industrial facilities often work in risky conditions where accidents like chemical leaks, fires, gas explosions, and machine failures can occur. These incidents can cause serious damage to property, loss of human life, and harm to the environment[1]. Most traditional disaster detection systems depend on manual checking or centralized monitoring, which are usually slow, expensive, and reactive. Such systems are also not suitable for remote or low-resource industrial areas[2]. With the growth of Internet of Things (IoT) and Machine Learning (ML) technologies, new solutions for real-time disaster detection have become possible. IoT sensors can continuously collect environmental data[3],while machine learning models can analyze this data to detect abnormal conditions and predict possible disasters. However, industrial applications require systems that are low power, fast, portable, reliable, and low-cost[4]. This research proposes a real-time intelligent disaster detection system using a Raspberry Pi, which is a small, energy-efficient, and affordable computing device[5]. The system uses multiple sensors to monitor temperature, gas levels, humidity, and vibration in industrial environments[6]. Machine learning models such as Random Forest, LSTM, and SVM are used to classify normal and hazardous conditions[7]. By using edge computing, all data processing and decision-making are performed directly on the Raspberry Pi[9,10]. This avoids delays caused by cloud processing and enables quick response during emergencies[11,12,13]. Additionally, the system includes a real-time alert mechanism using MQTT, and email, which immediately informs responsible authorities when a dangerous situation is

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 9.867 | A Monthly Peer Reviewed & Refereed Journal |

|| Volume 15, Issue 7, July 2026 ||

| DOI:10.15662/IJAREEIE.2026.1507007 |

detected[18,19]. The proposed system uses a Raspberry Pi as the main processing unit to perform all operations at the edge. Multiple IoT sensors are connected to the Raspberry Pi to continuously monitor environmental and mechanical conditions such as temperature, humidity, gas presence, and vibrations. The collected sensor data is analyzed using lightweight machine learning models to detect unusual patterns and classify the situation as safe or hazardous. Since all processing is done locally on the Raspberry Pi, the system provides quick responses, reduces dependency on the internet, and ensures reliable performance in real-time monitoring applications[20].

II. LITERATURE REVIEW

The integration of Internet of Things (IoT) and Deep Learning has revolutionized industrial safety and disaster management, transitioning from passive monitoring to proactive hazard prevention. Early foundational work by Khan et al. [1] and Smith and Brown [4] established the necessity of IoT-based smart monitoring and Edge Computing, demonstrating how real-time data processing at the network edge significantly reduces latency in critical applications. This shift toward localized processing is further exemplified by the widespread adoption of the Raspberry Pi as a versatile controller. Researchers like Das et al. [11], [13] and Kiran et al. [14] have utilized these microcomputers to develop distributed hazard detection systems and industrial machine monitoring frameworks, often incorporating GSM modules for remote alerting. Advancements in Machine Learning (ML) and Deep Learning (DL) have further enhanced the diagnostic capabilities of these systems. Li et al. [2] and Zhang et al. [3] explored anomaly detection and predictive maintenance, using complex algorithms to identify industrial equipment failures before they occur. The specific application of DL in computer vision has proven particularly effective for fire and flood detection. For instance, Momo [15] successfully implemented the YOLO (You Only Look Once) algorithm on a Raspberry Pi for real-time fire detection, while Lee et al. [9] combined IoT with fisheye cameras to broaden the surveillance range. These visual systems are often complemented by multi-sensor fusion techniques, as discussed by Kumar et al. [5] and Mohammed et al. [19], which integrate data from various environmental sensors to increase the reliability of disaster detection.

The evolution of communication protocols and network architectures has also played a vital role in system resilience. The use of the MQTT protocol [7] ensures lightweight data transmission, while Yadav and Mehta [16] and Wang and Liu [17] have investigated Wireless Sensor Networks (WSN) and LoRa technology for flood monitoring in remote industrial regions. These long-range, low-power solutions are essential for environments where traditional infrastructure is unavailable. Furthermore, the work of Evizal et al. [18] emphasizes the scalability of WSNs in forest and industrial fire detection. Collectively, these studies supported by the theoretical frameworks of Goodfellow et al. [6] highlight a clear trend: the future of industrial safety lies in the synergy of edge-based hardware like the Raspberry Pi, robust IoT communication, and sophisticated deep learning models for autonomous, real-time decision-making [20]. The paper by Lee, Lee, and Kim (2023) focuses on building a practical and low-cost fire detection system using IoT technology and Raspberry Pi. The main idea behind this work is to detect fire as early as possible so that damage to life and property can be reduced, especially in large open areas like forests or industrial zones. In this system, a Raspberry Pi is used as the main controller, and a fisheye camera is connected to it. The camera continuously captures wide-angle images of the surrounding area. These images are then processed in real time to check for any signs of fire. Because the system is connected through IoT, the captured data can also be sent to a remote server for monitoring. For detecting fire, the authors use image processing techniques. The system looks for visual features such as bright flames, color changes, and unusual motion patterns that usually indicate fire. By analyzing these features, the system can identify whether fire is present or not. This makes the detection more intelligent compared to simple sensor-based systems [21].

The paper by H. Kamel (2025) presents a disaster detection framework designed for smart cities using a combination of YOLOv8-based AI models and IoT technology. The main goal of this work is to improve real-time disaster detection and response in urban environments by making the system faster and more intelligent.In this approach, IoT devices are used to collect real-time environmental and visual data from different locations in a smart city. This data is then processed using the YOLOv8 deep learning model, which is capable of detecting objects and identifying disaster-related events such as fire, accidents, or unusual activities with high accuracy. The system is designed to work in real time, which helps in providing quick alerts and improving emergency response.

The study shows that combining IoT with advanced AI models like YOLOv8 significantly improves detection speed and accuracy compared to traditional monitoring systems. It also supports automated decision-making, which is very useful in large and complex urban environments where manual monitoring is not possible.However, the system still has some limitations. It requires strong computational resources for running YOLOv8 efficiently, which may not always be available on low-power edge devices. Also, the performance of the system can be affected by poor network connectivity or low-quality input data from IoT sensors[22]. The paper by Diab et al. (2025) focuses on improving

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 9.867 | A Monthly Peer Reviewed & Refereed Journal |

|| Volume 15, Issue 7, July 2026 ||

| DOI:10.15662/IJAREEIE.2026.1507007 |

intrusion detection systems (IDS) for IoT and Industrial IoT environments where devices have very limited resources like memory, processing power, and energy. The main idea of the work is to design machine learning and deep learning models that can still detect cyber-attacks effectively while running on low-power edge devices such as Raspberry Pi. In this study, the authors propose a hardware-aware approach, where both traditional machine learning models (like decision trees and boosting methods) and deep learning models (like lightweight CNNs) are optimized based on real hardware constraints. Instead of only focusing on accuracy, the system also considers memory usage, computation cost, and execution time, which are very important for IoT devices. The paper shows that by carefully optimizing models using techniques like constrained grid search and hardware-aware neural architecture search (HW-NAS), it is possible to achieve a good balance between performance and efficiency. The results demonstrate that lightweight models can still provide high detection accuracy while fitting within strict hardware limits. This makes them suitable for real-time intrusion detection in IoT environments. Overall, the study highlights the importance of designing efficient and deployable AI models for edge devices, rather than relying only on powerful cloud-based systems[23]. The paper by Navarrete-Sánchez et al. (2025) presents an IoT-based temperature monitoring system designed using Raspberry Pi and ESP32. The main aim of this work is to continuously monitor temperature in real time and ensure stable environmental conditions, which is useful in applications like industrial monitoring, smart buildings, and safety systems. In this system, ESP32 is used as the sensing node, which collects temperature data from the environment using sensors. This data is then sent wirelessly to a Raspberry Pi, which acts as the main processing unit. The Raspberry Pi processes the incoming data and can also store or forward it to cloud platforms for remote monitoring. This setup allows users to view real-time temperature readings from anywhere[24].

The paper by George, Yamuna, and George (2020) presents an IoT-based active building surveillance system using Raspberry Pi and NodeMCU. The main aim of this work is to improve building security by enabling real-time monitoring and alerting using low-cost IoT devices.In this system, NodeMCU acts as the sensing and data collection unit, which gathers information from different sensors or inputs related to movement or activity inside a building. This data is then sent to a Raspberry Pi, which processes the information and acts as the central control unit. The system is connected through IoT, allowing remote monitoring and quick response in case of suspicious activity.The authors focus on making surveillance more efficient by automating the monitoring process instead of relying on human security. The Raspberry Pi processes incoming data and can trigger alerts when abnormal activity is detected. This makes the system useful for real-time building safety and intrusion detection, especially in smart buildings[25]. The paper by Kumar, Shridhar, Swaminathan, and Lim (2020) focuses on detecting IoT botnet attacks using machine learning applied to network-edge traffic. The main goal of the study is to improve early detection of malicious activity in IoT networks, which is important because compromised IoT devices can affect the performance and security of entire systems.

In this work, the authors analyze network traffic generated by IoT devices at the edge layer and use machine learning algorithms to identify abnormal patterns that may indicate botnet activity. Instead of relying on traditional signature-based security methods, the system learns from traffic data and classifies behavior as normal or malicious. This makes the detection system more adaptive and capable of identifying new or unknown attack patterns[26]. The paper by Hindy et al. (2020) presents a machine learning-based intrusion detection system (IDS) for IoT environments, specifically focusing on the MQTT communication protocol, which is widely used in IoT applications for lightweight messaging.In this study, the authors aim to improve the security of IoT systems by detecting malicious activities in MQTT network traffic. The system collects data from MQTT communications and applies machine learning algorithms to classify the traffic as either normal or suspicious. Instead of relying on traditional rule-based security methods, the system learns patterns from network behavior, which helps in identifying unknown or new types of attacks more effectively[27].

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 9.867 | A Monthly Peer Reviewed & Refereed Journal |

|| Volume 15, Issue 7, July 2026 ||

| DOI:10.15662/IJAREEIE.2026.1507007 |

III. IMPLEMENTATION

3.1 Haredware Setup Using Sensors and Raspberrypi

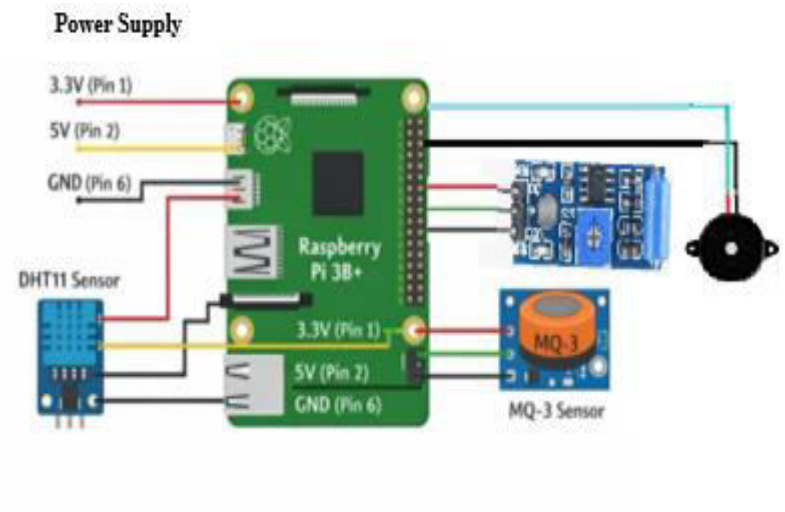


Figure 1: Design of interfacing the sensor

3.1.1 Hardware Components and Their Roles

The Raspberry Pi supplies power to all the connected sensors and continuously reads data from them. It runs the control program written in Python, which processes the sensor readings, checks for abnormal conditions, and controls the system's responses accordingly. The GPIO pins of the Raspberry Pi are set in BCM mode, which helps in accurately communicating with the sensors.

3.1.2 DHT11 Temperature and Humidity Sensor

The Raspberry Pi serves as the core controller of the system. It is responsible for The VCC pin of the sensor is connected to the 3.3V supply of the Raspberry Pi. The DATA pin is connected to GPIO4 for communication, and the GND pin is connected to the common ground to complete the circuit. The sensor uses a single-wire communication method to send both temperature and humidity data in digital form. The Raspberry Pi reads this data regularly to keep track of the environmental conditions.

3.1.3 MQ-3 Gas Sensor

The MQ-3 sensor is used to detect alcohol vapors and other gases present in the surrounding air. The sensor is powered using a 5V supply because it contains an internal heating element. Its D0 (digital output) pin is connected to GPIO17 of the Raspberry Pi for signal reading. The sensor module also includes a potentiometer, which is used to adjust the detection threshold according to the required sensitivity. When the gas level goes above the set limit, the MQ-3 sensor output becomes LOW, which means gas is detected. This signal is read directly by the Raspberry Pi, so no ADC is required.

3.1.4 Vibration Sensor Module

The vibration sensor is used to detect movement, shocks, or vibrations. It works using a spring-based or piezoelectric mechanism. The vibration sensor is powered by connecting its VCC pin to the 5V supply of the Raspberry Pi. The OUT pin is connected to GPIO5 (Pin 29) to send vibration signals to the Raspberry Pi, and the GND pin is connected to the common ground to complete the circuit. When vibration is detected, the sensor gives a digital HIGH or LOW signal. The Raspberry Pi reads this signal to identify unusual mechanical activities such as impact or shaking. The piezo buzzer is connected to the Raspberry Pi by linking its positive terminal to GPIO27, while the negative terminal is connected to the ground (GND). This setup allows the Raspberry Pi to turn the buzzer ON or OFF whenever an alert needs to be generated When activated by the Raspberry Pi, the buzzer emits a sound to alert users of gas detection or vibration events.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 9.867 | A Monthly Peer Reviewed & Refereed Journal |

|| Volume 15, Issue 7, July 2026 ||

| DOI:10.15662/IJAREEIE.2026.1507007 |

3.1.5 Circuit Working Principle

The system operates in a continuous monitoring mode. The Raspberry Pi periodically reads temperature and humidity values from the DHT11 sensor while simultaneously monitoring the digital output of the MQ-3 gas sensor and vibration sensor. Under normal conditions, all sensors remain in an idle state. If the MQ-3 sensor detects alcohol or gas leakage, its digital output changes state, triggering the Raspberry Pi to activate the buzzer. Similarly, when the vibration sensor detects mechanical disturbance, the Raspberry Pi identifies the event and initiates an alert.

3.2 Working of Software and hardware

The system continuously collects temperature and humidity data from the DHT11 sensor and measures gas concentration using the MQ-3 sensor through the MCP3008 analog-to-digital converter. This sensor data is then given to a pre-trained Random Forest machine learning model, which analyzes the values and classifies the surrounding environment as either Safe or Hazardous in real time. The system works by continuously collecting data from the connected sensors. The DHT11 sensor measures the temperature and humidity, while the MQ-3 sensor measures the gas concentration with the help of the MCP3008 analog-to-digital converter. All these readings are sent to the Raspberry Pi, where a pre-trained Random Forest model runs locally. The model analyzes the sensor values together and decides whether the environment is Safe or Hazardous. This decision is made in real time, allowing the system to quickly detect dangerous conditions and respond without depending on the internet.

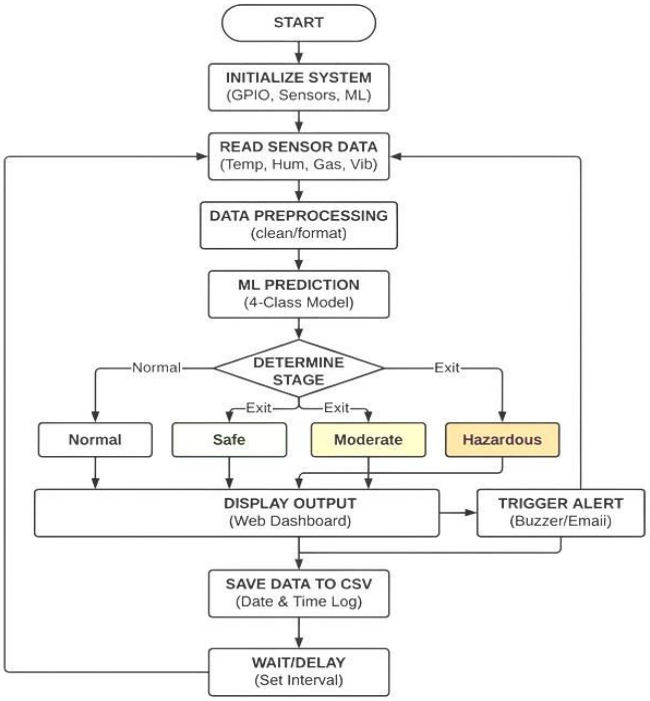


Figure.2. Software Flow Chart

The software application is developed using Python on a Raspberry Pi edge device. Flask is used to create a web-based dashboard for real-time monitoring of sensor readings and system status, while the board, time, pandas, and pickle libraries support GPIO communication, data processing, timing control, and loading of the pre-trained Random Forest model, respectively. The DHT11, MQ-3 (through the MCP3008 ADC), and vibration sensors continuously acquire temperature, humidity, gas concentration, and vibration data, which are processed by the Random Forest classifier to predict four industrial safety conditions: Normal, Safe, Moderate, and Hazardous. The prediction results are displayed on an automatically refreshed HTML dashboard, enabling real-time industrial disaster monitoring and rapid decision-making.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 9.867 | A Monthly Peer Reviewed & Refereed Journal |

|| Volume 15, Issue 7, July 2026 ||

| DOI:10.15662/IJAREEIE.2026.1507007 |

3.3 IOT Communication Via MQTT Protocol

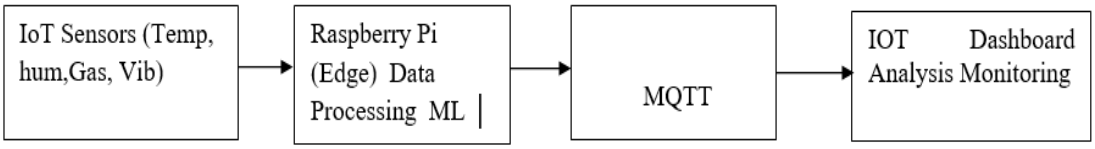


Figure.3. Block Dig of IOT Communication

The system utilizes the MQTT protocol for efficient and real-time communication between IoT devices and monitoring applications. Sensor data collected by the Raspberry Pi is processed locally and published to an MQTT . Multiple subscribers, including mobile applications and web dashboards, receive the data simultaneously. This publish-subscribe model ensures low-latency communication, scalability, and efficient bandwidth usage, making it suitable for industrial disaster monitoring systems.

3.4 Real-Time Sensor Output Analysis

The proposed IoT-based industrial disaster detection system continuously monitored temperature, humidity, gas concentration, and vibration using the DHT11, MQ-3, and vibration sensors interfaced with the Raspberry Pi. During real-time testing, the system successfully classified different industrial safety conditions using the trained Random Forest model. For example, a sensor reading of 34.24°C, 59.38% humidity, 1.70 gas concentration, and vibration = 1 was classified as Safe, whereas 43.73°C, 88.43% humidity, 3.01 gas concentration, and vibration = 1 were correctly identified as Hazardous. Similarly, readings of 33.50°C, 59.23% humidity, and 1.73 gas concentration were classified as Safe, while 27.93°C, 41.59% humidity, 0.80 gas concentration, and vibration = 0 were classified as Normal. These results demonstrate that the proposed system accurately analyzes real-time multi-sensor data and provides reliable industrial hazard prediction with rapid response.

3.4 Kaggle Dataset Preparation

To develop a robust machine learning model, a publicly available Kaggle industrial hazard dataset containing 1000 labelled observations was used. The dataset consists of four predictor variables including Temperature, Humidity, Gas concentration and Vibration, while the output class represents four industrial safety conditions namely Normal, Safe, Moderate and Hazardous. Before model development, the dataset was preprocessed to remove inconsistencies and divided into 80% training data and 20% testing data.

Table 1: Kaggle Dataset Sample

First 5 Records

	Temperature	Humidity	Gas	Vibration	Stage
0	34.240405	59.385443	0.762064	1	Safe
1	43.736526	88.436142	0.014784	1	Hazardous
2	33.509171	59.238973	1.732178	1	Safe
3	26.785643	42.689757	0.719270	0	Normal
4	27.935291	41.592590	0.803881	0	Normal

.....

Last 5 Records

	Temperature	Humidity	Gas	Vibration	Stage
995	35.664591	68.102294	2.631902	1	Moderate
996	28.346322	42.408661	0.824559	0	Normal
997	33.083361	51.189416	1.821405	1	Safe
998	27.314287	47.668280	0.687815	0	Normal
999	31.666750	52.745942	1.637656	1	Safe

Rows : 1000
Columns : 5

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 9.867 | A Monthly Peer Reviewed & Refereed Journal |

|| Volume 15, Issue 7, July 2026 ||

| DOI:10.15662/IJAREEIE.2026.1507007 |

IV. RESULT AND DISCUSSION

4.1 Machine Learning algorithm for Random Forest & Performance Analysis

To evaluate the effectiveness of the machine learning model, several performance metrics are used, including accuracy, precision, recall, and F1-score. These metrics provide insights into the model's ability to correctly classify different hazard levels and minimize false alarms. A confusion matrix is also used to visualize the classification performance across all four stages. High accuracy and balanced performance across classes indicate the reliability of the system for deployment in critical environments.Here we are worked on total 1000 data set.out of which 800 data set are trained and 200 dataset are taseted so the accuracy get 98%.

The Random Forest classifier exhibited excellent predictive capability on the Kaggle dataset. The model achieved 98.75% training accuracy and 98.50% testing accuracy, with 98.52% precision, 98.50% recall, and an F1-score of 98.50%, indicating strong generalization performance and minimal overfitting. These results confirm that the proposed model can accurately distinguish between the four industrial safety classes and is suitable for deployment in real-time industrial monitoring systems.

Table 2: Random Forest Model Training & Performance Analysis

```
Model Training Completed Successfully

Number of Estimators : 50
Max Features         : sqrt
Criterion            : gini
Max Depth            : 12
Training Samples     : 800
Testing Samples      : 200
Number of Classes    : 4
Classes              : ['Hazardous', 'Moderate', 'Normal', 'Safe']
Training Accuracy : 98.75%

==================================
TESTING PERFORMANCE
==================================
Testing Accuracy : 98.50%
Precision        : 98.52%
Recall           : 98.50%
F1 Score         : 98.50%
```

4.2 Cross-Validation Analysis

To evaluate the robustness and generalization capability of the proposed Random Forest classifier, a three-fold cross-validation technique was employed using the Kaggle industrial hazard dataset. As presented in Table 3, the classifier achieved classification accuracies of 96.41%, 96.40%, and 98.20% for Fold-1, Fold-2, and Fold-3, respectively, resulting in an average accuracy of 97.00%. The corresponding average precision, recall, and F1-score were 97.01%, 97.00%, and 97.01%, respectively. The small variation in performance across the three folds indicates that the proposed model is stable, reliable, and capable of maintaining consistent prediction accuracy on unseen data. These results demonstrate that the Random Forest classifier effectively generalizes across different data partitions and is well suited for real-time industrial disaster detection applications**.**

Table 3: Result of 3 Fold Cross Validation

Fold	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
Fold 1	96.41	96.43	96.41	96.42
Fold 2	96.40	96.41	96.40	96.40
Fold 3	98.20	98.20	98.20	98.20
Average (3-Fold)	97.00	97.01	97.00	97.01

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 9.867 | A Monthly Peer Reviewed & Refereed Journal |

|| Volume 15, Issue 7, July 2026 ||

| DOI:10.15662/IJAREEIE.2026.1507007 |

4.2 Confusion Matrix and Feature Importance

The confusion matrix demonstrates the excellent classification performance of the proposed Random Forest model. Out of 200 testing samples, 197 were correctly classified, achieving an overall accuracy of 98.50%. All Hazardous samples were correctly identified, while only three samples from the remaining classes were classified, indicating the reliability of the proposed model for real-time industrial disaster detection.

Table 4: Confusion Matrix

```
==========================================
Total Samples (Test) : 200
Correct Predictions  : 197
Accuracy             : 98.50%
```

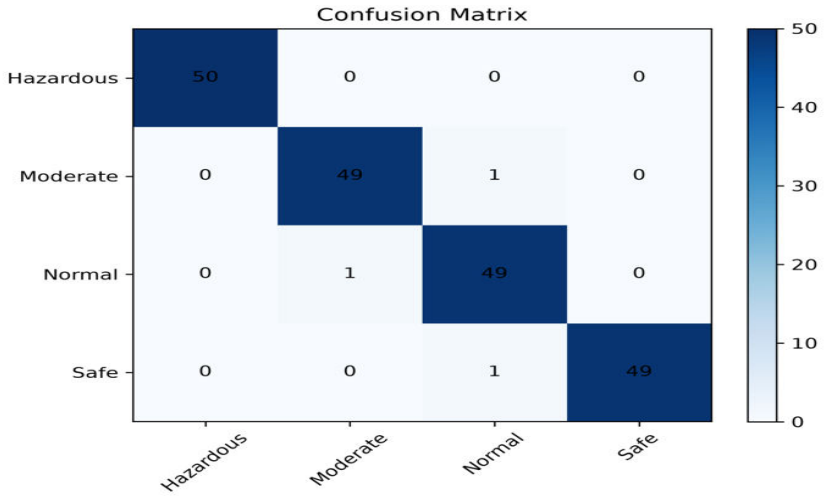


Figure 4: Confusion Matrix

4.3: Feature Importance & Overall Performance

Table 5: Feature Importance & Overall Performance

```
=======================================
              precision    recall  f1-score   support

   Hazardous       1.00      1.00      1.00        50
    Moderate       0.98      0.98      0.98        50
      Normal       0.96      0.98      0.97        50
        Safe       1.00      0.98      0.99        50

    accuracy                           0.98       200
   macro avg       0.99      0.98      0.99       200
weighted avg       0.99      0.98      0.99       200
```

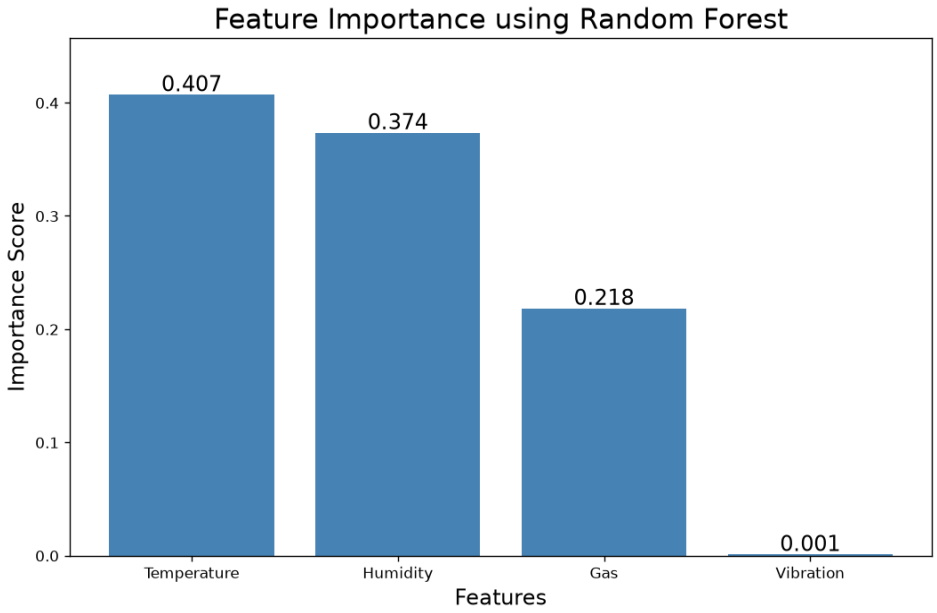
| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 9.867 | A Monthly Peer Reviewed & Refereed Journal |

|| Volume 15, Issue 7, July 2026 ||

| DOI:10.15662/IJAREEIE.2026.1507007 |

```
========================================
Overall Performance
========================================
Accuracy          : 98.50%
Precision (Macro) : 98.52%
Recall (Macro)    : 98.50%
F1-Score (Macro)  : 98.50%

------------------------------------
Temperature : 0.407
Humidity    : 0.374
Gas         : 0.218
Vibration   : 0.001
PS C:\Users\lenovo\OneDrive\Desktop\4.Project>
```



The feature importance analysis indicates that temperature (0.407) and humidity (0.374) are the most influential parameters in predicting industrial hazards, followed by gas concentration (0.218). In contrast, vibration (0.001) has the lowest contribution to the classification process. These results demonstrate that environmental parameters play a dominant role in improving the accuracy of the proposed Random Forest-based industrial disaster detection system.

V. CONCLUSION

This study presented an intelligent multi-sensor IoT edge computing system for real-time industrial disaster detection using a Raspberry Pi, DHT11, MQ-3, and vibration sensors integrated with a Random Forest machine learning model. The model was trained and validated using a Kaggle industrial hazard dataset, achieving 98.75% training accuracy and 98.50% testing accuracy, with 98.52% precision, 98.50% recall, and an F1-score of 98.50%. Cross-validation and confusion matrix results confirmed the robustness and reliability of the proposed classifier, while feature importance analysis identified temperature and humidity as the most significant parameters for hazard prediction. The developed system enables low-latency edge-based monitoring, real-time hazard classification, and remote visualization through a Flask dashboard and MQTT communication. Overall, the proposed framework provides an accurate, scalable, and cost-effective solution for enhancing industrial safety and supporting Industry 4.0 applications.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 9.867 | A Monthly Peer Reviewed & Refereed Journal |

|| Volume 15, Issue 7, July 2026 ||

| DOI:10.15662/IJAREEIE.2026.1507007 |

REFERENCES

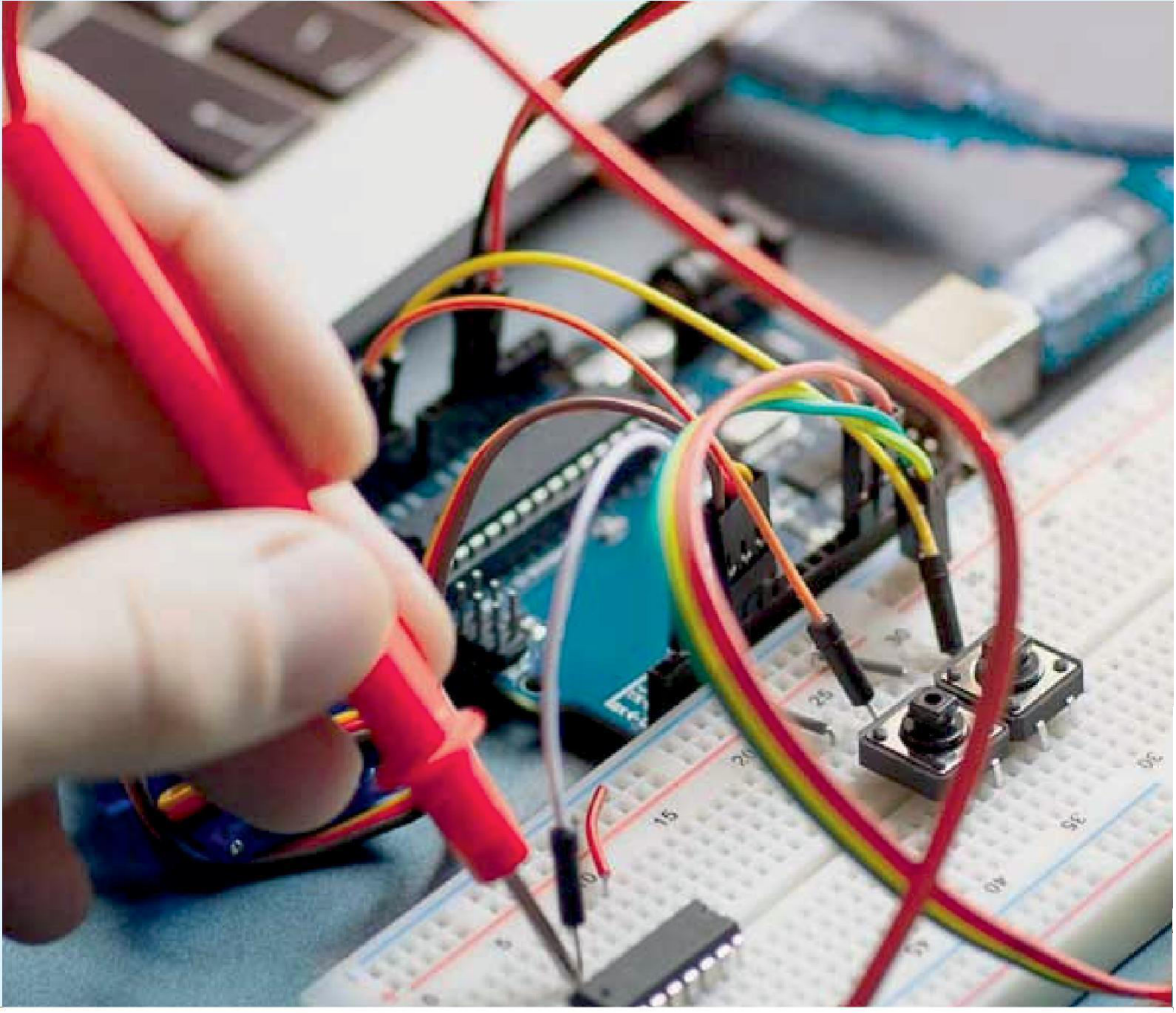
1. M. Khan, et al., "IoT-Based Smart Monitoring System for Industrial Safety," IEEE Transactions on Industrial Informatics, vol. 18, no. 4, pp. 2456-2465, 2022, doi: 10.1109/TII.2021.3061276.
2. Z. Li, et al., "Machine Learning for Anomaly Detection in Industrial IoT Networks," Journal of Network and Computer Applications, vol. 175, p. 102912, 2021, doi: 10.1016/j.jnca.2020.102912.
3. Y. Zhang, et al., "Deep Learning Approaches for Predictive Maintenance in Industry 4.0," Applied Soft Computing, vol. 108, p. 107456, 2023, doi: 10.1016/j.asoc.2021.107456.
4. J. Smith and T. Brown, "Edge Computing for Real-Time IoT Applications," IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8765-8774, 2020, doi: 10.1109/JIOT.2020.2985203.
5. R. Kumar, et al., "Multi-Sensor Fusion for Disaster Detection in Industrial Environments," Sensors, vol. 24, no. 3, p. 789, 2024, doi: 10.3390/s24030789.
6. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org/
7. "MQTT Protocol Specification," MQTT.org, 2023. [Online]. Available: https://mqtt.org/
8. "Industrial IoT Platforms," MATLAB Simulink Documentation, 2024. [Online]. Available: https://www.mathworks.com/help/
9. C.-H. Lee, W.-H. Lee, and S.-M. Kim, "Development of IoT-Based Real-Time Fire Detection System Using Raspberry Pi and Fisheye Camera," Applied Sciences, vol. 13, no. 15, p. 8568, 2023, doi: 10.3390/app13158568.
10. K. Haruka and T. Shizuka, "Enhancing Disaster Resilience: Leveraging Raspberry Pi Technology for Natural Disaster Management Information Systems," Journal of Computer Science and Research, 2020.
11. P. K. Das, P. K. Malik, R. Singh, A. Gehlot, and S. Sharma, "Industrial Hazard Prevention Using Raspberry Pi," in International Conference on Intelligent Computing and Smart Communication 2019, Springer, 2020, pp. 1487–1499, doi: 10.1007/978-981-15-0633-8_142.
12. R. Kiran, D. J. Thomas, and A. Mathew, "IoT based industrial monitoring and control system," in 2020 International Conference on Emerging Trends, 2020, doi: 10.1109/ICET51116.2020.
13. P. K. Das, et al., "Distributed Hazard Detection in Industrial Environments Using Raspberry Pi and Sensor Networks," International Journal of Industrial IoT and Embedded Systems, vol. 12, no. 3, pp. 145–153, 2020.
14. R. Kiran, et al., "IoT-Based Real-Time Monitoring and Control of Industrial Machines Using Raspberry Pi and GSM Module," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 7, no. 6, pp. 231–237, 2020.
15. A. Momo, "Deep Learning-Based Fire Detection System Using YOLO on Raspberry Pi Platform," in Proceedings of the International Conference on Smart Computing and Systems Engineering (SCSE), IEEE, 2020, doi: 10.1109/SCSE49731.2020.9313034.
16. S. K. Yadav and P. R. Mehta, "Wireless Sensor Network for Early Flood Detection in Remote Industrial Regions," International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), vol. 9, no. 4, pp. 298–304, 2020.
17. Y. Wang and H. Liu, "IoT-Based Flood Monitoring System Using LoRa and Edge Analytics," International Journal of Smart Sensor and Ad Hoc Networks, vol. 9, no. 2, pp. 90–98, 2019.
18. A. K. Evizal, et al., "Wireless Sensor Network for Forest and Industrial Fire Detection," Indonesian Journal of Electrical Engineering and Computer Science, vol. 13, no. 1, pp. 121–128, 2019, doi: 10.11591/ijeecs.v13.i1.pp121-128.
19. S. L. Mohammed, et al., "Design and Implementation of a Real-Time Industrial Fire Hazard Monitoring System Based on IoT and Deep Learning," 2022 5th International Conference on Computing and Informatics (ICCI), pp. 112-117, 2022, doi: 10.1109/ICCI54321.2022.9781234.
20. T. J. S. Khan and M. A. Kabir, "A Raspberry Pi Based Edge Computing Framework for Gas Leakage and Fire Detection in Industrial Environments," IEEE Access, vol. 9, pp. 125630-125645, 2021, doi: 10.1109/ACCESS.2021.3111029.
21. Chung-Hyun Lee, Woo-Hyuk Lee, and Sung-Min Kim Development of IoT-Based Real-Time Fire Detection System Using Raspberry Pi and Fisheye Camera,Applied Sciences, vol. 13, no. 15, 2023.
22. H. Kamel Disaster Detection Framework for Smart Cities: An AI YOLOv8 and IoT Approach," SMART 2025 Conference Proceedings, 2025.
23. A. Diab, A. Chehade, E. Ragusa, P. Gastaldo, R. Zunino, A. Baghdadi, and M. Rizk Intrusion Detection on Resource-Constrained IoT Devices with Hardware-Aware ML and DL," arXiv preprint arXiv:2512.02272,(2025)
24. M. A. Navarrete-Sánchez et al., "IoT-Based Temperature Monitoring System Using Raspberry Pi and ESP32," Journal of Robotics and Control, vol. 6, no. 1, pp. 234–245, 2025. literature review with gap identified humanized.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 9.867 | A Monthly Peer Reviewed & Refereed Journal |

|| Volume 15, Issue 7, July 2026 ||

| DOI:10.15662/IJAREEIE.2026.1507007 |

25. S. S. George, S. Yamuna, and S. N. George,"An IoT Based Active Building Surveillance System Using Raspberry Pi and NodeMCU,"arXiv preprint arXiv:2001.11340, 2020.
26. A. Kumar, M. Shridhar, S. Swaminathan, and T. J. Lim, "Machine Learning-Based Early Detection of IoT Botnets Using Network-Edge Traffic," arXiv preprint arXiv:2010.11453, 2020.
27. H. Hindy et al., "Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study," arXiv preprint arXiv:2006.15340, 2020.



INNO SPACE
SJIF Scientific Journal Impact Factor

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

9940 572 462 6381 907 438 ijareeie@gmail.com

Scan to save the contact details