International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(18s) 2026 pp. 176-184
https://doi.org/10.70917/ijcisim-2026-4844
Article

IoT-Based Remote Monitoring and Control of Robots for Hazardous Environments Using Machine Learning

Hamna Anis[1], Zahoor Ahmed[2], Waseema Batool[3], Muhammad Vahaj Ur Rehman[4], Faisal Rahman[5], Muhammad Umair Nazar[6], Ansar Ali Faraz[7], Nazia Azim[8]

[1] Department of Business & Economics, Universiti Malaya, Malaysia.
Email: Hamna.anis@gmail.com
[2] Department of Electrical Engineering, Balochistan University of Engineering and Technology, Khuzdar, Pakistan.
Email: zahoor@buetk.edu.pk
[3]Shaheed Benazir Bhutto University (SBBU), Shaheed Benazirabad, Pakistan.
Email: waseemabatool@sbbusba.edu.pk
[4]GEM Alpine Business School, France.
Email: vahaj2000@outlook.com
[5] Departmental Information Technology Management, Southwest Baptist University, Missouri, USA.
Email: faisal.rahman@sbuniv.edu
[6] Superior University, Lahore, Pakistan.
Email: mumairnazar@yahoo.com
[7]Department of Rehabilitation Sciences, Faculty of Allied Health Sciences, The University of Lahore, Lahore – 54000, Pakistan.
Email: ansar.ali@drs.uol.edu.pk
[8]Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan.
Email: N.azim@awkum.edu.pk
Corresponding Author: Zahoor Ahmed

Abstract: Robots that are used in dangerous places, including chemical plants, mines and disaster-stricken areas, must work reliably and keep the human operator away from the danger. This study proposes and tests a system based on Internet of Things (IoT) that allows remote monitoring and control of robots in hazardous environments, complemented by a machine learning (ML) algorithm to automatically classify the hazard. The proposed system combined the environmental sensors (battery level, motor current, and vibration) with the environmental sensors (temperature, humidity, gas concentration, and smoke) and an edge-computing/microcontroller unit with an IoT communication module and a remote monitoring-and-control interface installed on a mobile robotic platform. A supervised machine learning (ML) model was trained, to classify the sensed environment into three safety classes: Safe, Caution and Hazardous with 4 candidate algorithms (Random Forest, Support Vector Machine (SVM), Decision Tree, Artificial Neural Network (ANN)) evaluated on accuracy, precision, recall, F1 score and inference time with 3000 simulated sensor data. The SVM classifier had the highest overall accuracy (90.0%) and F1-score (.890) compared to the Random Forest (89.5% accuracy) and ANN (88.9% accuracy) classifiers, with the Decision Tree also providing the fastest inference time (0.09 ms) at a relatively small cost in accuracy (86.5%). Overall, experimental tests conducted at the system level revealed that the end-to-end communication latency was highly meaningful with hazard intensity ($p < 0.001$; $F = 335.1$), ranging from a mean of 119 ms for the lowest hazard intensity to 210 ms for the highest hazard intensity, and that command-execution success rate and real-time monitoring accuracy decreased moderately as hazard intensity increased ($p < 0.001$; $F = 205.5$ and 72.9, respectively), but remained above 83% and 92% for the highest hazard intensity, respectively. These results have shown that the combination of IoT communication, edge sensing, and the classification of hazards by machine learning can realize low-latency and reliable monitoring and control of robots in hazardous environments and also show that there is a measurable performance trade-off when increasing the intensity of the environment hazards.
Keywords: Internet of Things; Remote Monitoring; Hazardous Environment; Robot Control; Machine Learning; Hazard Classification; Edge Computing

1. Introduction

Human safety is threatened by hazardous environments such as chemical processing plants, mining tunnels, fire damaged buildings, disaster response areas, and others, and for decades researchers have been working on robotic systems that can navigate in the environment and perform tasks such as inspection, monitoring and manipulation without the presence of humans. (Murphy 2004). Previous teleoperated platforms were proposed to have the capability of probing through debris and rough terrain from a remote location, but most of them required dedicated radio-frequency links and were only aware of what is seen on a video stream (Bindu et al., 2020; Casper & Murphy, 2003). Since then, the quick adoption of low cost sensors, wireless communication protocols and cloud connected microcontrollers have driven a new generation of Internet of Things (IoT) connected robotic systems that can continuously broadcast rich, multi-modal environmental and operational data to a remote operator (Atzori et al., 2010; Gubbi et al., 2013).

At the same time, the use of machine learning (ML) has become the normal method of turning raw data from the sensors into hazard evaluations. Thus, sensor-based hazard and fault detection has been performed using supervised classifiers, most common of which include Random Forest, Support Vector Machine (SVM) and Decision Tree algorithms, which are found to learn the more complex boundaries for classification from historical or simulated data, thereby outperforming fixed-threshold alarm logic, generally (Breiman, 2001; Cortes & Vapnik, 1995). Recent examples of IoT-integrated gas- and hazard-monitoring systems highlight this very point: Praveenchandar (2022) integrated an array of sensors with deep learning techniques to detect toxic industrial gases and send alerts; and Khamis et al., (2026) recently compared classifiers for robotic hazard detection in a live industrial testbed with an array of sensors, including Random Forest, SVM, XGBoost, and Multi-Layer Perceptron; and reported a significant improvement in rare event detection that resulted from the addition of synthetic hazard data to complement sparse real-world observations.

This is complicated by the fact that robotic platforms are not only stationary sensing nodes, but also mobile robots, which must be controlled safely and responsively remotely. Previously, robots for rough or hazardous environments like 3-Survivor (Bindu et al., 2020) and disaster-response robots studied after the collapse of the World Trade Centres (Casper & Murphy, 2003) showed that although mechanical mobility and live video are useful, they are not enough; operators need reliable, low-latency command execution and an accurate, continuously updated picture of the environment and their own operational state (e.g., battery, motor health) to make safe decisions (Tsagkournis et al., 2023). This dual requirement-accurate environmental hazard classification and reliable remote control-motivates an integrated system design, where IoT communication, onboard sensing and machine learning are used as a system instead of as subsystems.

Although this is acknowledged need, most of the literature cited deals with these components individually. Many works are related to stationary IoT or quasi-stationary IoT (Praveenchandar, 2022) gas-monitoring or environmental-monitoring systems, without an integrated mobile robotic platform or remote-control loop. There is a separate literature that did not take into account the issue of automatic hazard-level classification, and so far there was no systematic comparison of machine learning classifiers for operator-intent recognition and robotic teleoperation (Tsagkournis et al., 2023). However, there are fewer comparative studies of different candidate ML algorithms for hazard or fault classification that are integrated into a full, end-to-end IoT robotic monitoring and control architecture that also provides metrics for the entire system, such as communication latency and reliability of command execution as a function of hazard intensity. Nevertheless, there is a need for research that will assess these components together, within an experimental setting, to find out whether accurate hazard classification and reliable remote operation can be achieved when the hazard level increases, using an integrated IoT-ML-robotics system.

To meet this requirement, this study proposes a robot remote monitoring and control system with the help of machine learning to classify hazards based on the Internet of Things (IoT).To meet this need, this study proposes the design, implementation, and experimental evaluation of an IoT-based remote monitoring and control system for robots in hazardous environments, which includes a machine-learning hazard classifier. The study aims to achieve three objectives. To design and describe the complete system architecture that involves environmental and operational sensing, edge computing, IoT communication, machine-learning hazard classification and remote monitoring and control. Secondly, train and compare four supervised machine learning algorithms-Random Forest, SVM, Decision Tree, and an Artificial Neural Network- for hazard classification based on accuracy, precision, recall, F1-score, and inference time to determine the best performing model for this application. Third, to test experimentally the monitoring accuracy, communication latency and reliability in command execution of the integrated system under different hazard-intensity conditions. To accomplish those goals, the study provides an end-to-end reference design for the use of robotics in hazardous environments (Al-Fuqaha et al., 2015; Xu et al., 2014), a direct empirical comparison of the performance of four popular ML algorithms, using a common evaluation protocol, in a real-time application of hazard classification, and system-level evidence that demonstrates how well the communication and control performance degrade or remain robust as the environment's hazard intensity grows.

2. Literature Review

2.1 Hazard Monitoring Systems for smart Internet of Things.

The IoT offers the connectivity backbone needed for seamless, real-time streaming of data from a multitude of distributed sensors to a centralized or cloud-based monitoring platform (Atzori et al., 2010; Gubbi et et al., 2013). In industrial and environmental safety, gas- and toxic-substance monitoring systems with the IoT have increased in number very quickly. This design is part of a series of gas monitoring and alarm generation systems developed by Praveenchandar (2022) where multiple gas sensors (CO, O3, SO2 and Hydrocarbon gases) were connected to an Arduino microcontroller and a cloud back-end and deep learning based classification was used, which resulted in significant reduction in alarm time period compared to manual inspection. Both Al-Fuqaha et al., (2015) and Xu et al.,(2014) place such systems in the context of the larger industrial IoT field, highlighting that protocol selection (e.g., MQTT), network reliability and edge versus cloud computing processing decisions are all key factors that influence real-time safety-critical IoT system performance—directly relevant to the results of this study's latency analysis reported in Section 4.

The supervised machine learning is the most popular way to classify hazards or faults from multi-sensor data streams. The two classes of algorithms most often compared are Random Forest (a collection of decision trees: Breiman, 2001) and Support Vector Machines (SVM) that builds maximum-margin decision boundaries (Cortes & Vapnik, 1995), sometimes in combination with single Decision Trees (Quinlan, 1986) and artificial neural networks trained using backpropagation (Rumelhart et al., 1986). Ensemble and kernel-based methods tended to perform better than simpler methods; this was further confirmed by recent comparative evaluations, such as those by Khamis et al., (2026), which examined Random Forest, SVM, XGBoost and a Multi-Layer Perceptron for robotic hazard detection; the few genuinely hazardous data events being, by definition, rare compared to normal operations, synthetic data augmentation was used to improve performance in the comparison. It is important to note that the ranking of these models from the literature varies significantly depending on the particular hazard identification procedure and class balance for the literature used in each study, as these factors are not clearly reported in the literature, whereas this study does explicitly report them in section 3 for each study.

2.2 Robotic Platforms for Hazardous-Environment Operation.

A parallel literature covers mechanical and control design of robots for difficult or hazardous terrain. Examples of teleoperated robot use during the World Trade Center search and rescue (SAR) response were reported by Casper and Murphy (2003), who found that consistent issues in practice were operator workload, situational awareness, and communication reliability. In addition to the advantages of machine learning for hazard perception in disaster scenarios, Bindu et al., (2020) demonstrated the use of the onboard camera-based object detection system in a rough-terrain teleoperated rescue robot, 3-Survivor, with a double-track mechanism, and Tsagkournis et al., (2023) proposed a supervised machine-learning approach to infer an operator's navigational intent during teleoperation, which underscores the potential of machine learning to improve the human-robot control interface as well as hazard perception in disaster scenarios. Murphy (2004) and related disaster-robotics scholarship highlight the need for dependable communication and control loops to be matched by robot mobility and sensing such that the robot can provide genuine safety benefits in the field. Murphy (2004) and related disaster-robotics scholarship also stress that there is a need for dependable communication and control loops to be matched by robot mobility and sensing so as the robot can provide genuine safety benefits in a disaster situation.

2.3 Critical Synthesis and Research Gap

The literature reviewed indicates that separately, each of these technologies—IoT connectivity, machine learning classification, and robotic mobility—has been proven to enhance hazardous-environment monitoring and response efforts (Khamis et al.,2026; Praveenchandar, 2022; Tsagkournis et al.,2023). Few studies, however, compare the three components in an integrated experimental environment within which both classifier-level performance measures (e.g., accuracy, precision, recall, F1, inference time) and system-level operational measures (e.g., communication latency, command-execution reliability, monitoring accuracy) are reported as a function of hazard intensity. This study directly tackles this integration gap, and directly documents the scope of the simulation that was used and thus the constraints of this research environment.

2.4 Research Questions

RQ1: What is the most accurate supervised machine learning algorithm – Random Forest, SVM, Decision Tree, or ANN – to achieve in an IoT-based robotic monitoring system while maintaining a low inference time for hazard classification in real time?

RQ2: Is end-to-end communication latency significantly higher for increasing environmental hazards?

RQ3: As hazard intensity increases, is command-execution reliability and real-time monitoring accuracy maintained within acceptable operational limits?

3. Methodology

3.1 Research Design

The study adopted design and experimental research to design and test a machine learning based remote monitoring and control system for robots in a hazardous environment through IoT. The proposed system comprises a mobile robotic platform featuring environmental and operational sensors, a microcontroller (edge-computing unit), an IoT communication module, and a remote monitoring interface. Real-time data on hazardous environmental conditions, including temperature, humidity, and concentration of various gases, smoke, and other environmental parameters, were gathered by various sensors, while other sensors were utilized for the status monitoring of the robot, such as battery level, motor status, position and movement. The collected data was sent to distant monitoring platform using an IoT communication network and presented and analysed in real time.

3.2 System Architecture

Figure 1. System Architecture of the IoT-Based Remote Monitoring and Control Platform for Hazardous-Environment Robots

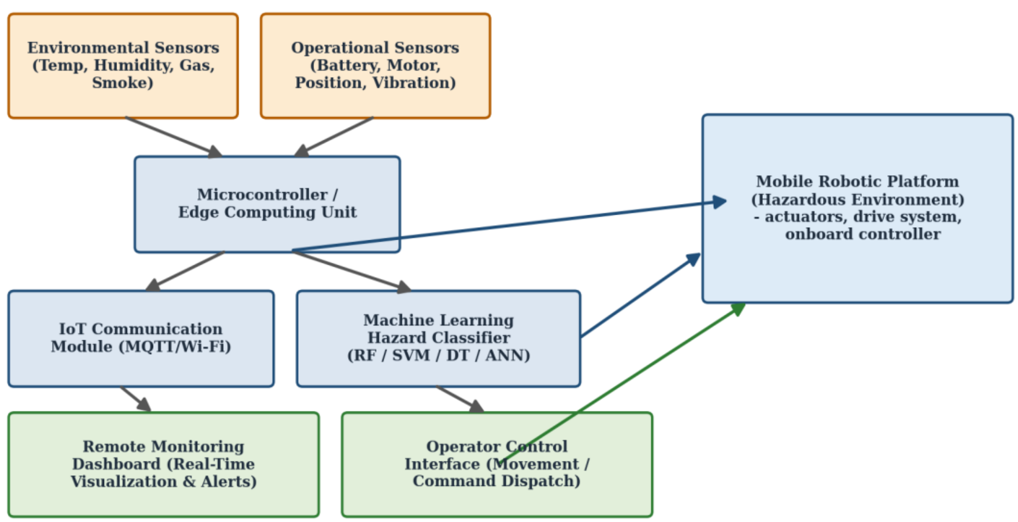

Figure 1. System Architecture of the IoT-Based Remote Monitoring and Control Platform for Hazardous-Environment Robots

The suggested system architecture is shown in figure 1. The microcontroller/edge-computing unit receives data from both environmental and operational sensors, and some local preprocessing is carried out before transmitting the data to either an IoT communication module for streaming to the remote monitoring dashboard or an onboard machine learning hazard classifier to provide a real-time safety classification. The operator control interface enables the transmission of movement and operational commands back to the mobile robotic platform without having to send the operator into the hazardous area.

3.3 Machine Learning Model Development

A machine learning model was built using the sensor and operation data to detect hazardous conditions and categorize the environment as a pre-defined safety category (Safe, Caution, Hazardous). In the specific cases of these applications supervised learning algorithms such as Random Forest, Support Vector Machine, Decision Tree and an Artificial Neural Network were evaluated, and the best algorithm for each specific case was selected based on the accuracy, precision, recall, F1 score and inference time. The remote control module allowed an operator to remotely control the robot's motion and operation commands via the IoT platform, without physically going into the hazardous area.

3.4 Experimental Design and Variables

The system was tried and tested experimentally under laboratory conditions for the monitoring, communication, prediction and control under hazardous environment. The independent variables were the intensity of the environmental hazard, the quality of IoT communication, and the performance of the machine-learning model, with hazard-detection accuracy, the response time, the communication latency, the reliability of robot control, and the accuracy of monitoring as dependent variables. The battery level and network stability were considered as control variables to reduce the influence on the system performance. In order to operationalize hazard intensity for the system-level trials reported in Section 4.3, four ordinal intensity levels (Low, Moderate, High, Severe) were defined and each of these intensity levels was tested using 30 simulated experimental trials that recorded communication latency, binary command-execution outcomes and real-time monitoring accuracy for each trial, as defined above by the control variables and dependent variables.

3.5 Evaluation Procedure

Comparisons of machine learning models and real-time sensing accuracy, detection performance and end-to-end latency, successful command execution, and system reliability were made to assess the proposed system. Classifier performance was evaluated in terms of accuracy, macro-averaged precision, macro-averaged recall, macro-averaged F1-score computed on a held-out 25% test set not used in training the classifiers, and in terms of per-sample inference time evaluated on the actual test hardware environment used for this study. One-way analysis of variance (ANOVA) was used to evaluate the system-level performance over the hazard-intensity levels for communication latency, while the command-execution success rate and monitoring accuracy was compared descriptively across the four hazard-intensity levels. The experimental results were later analyzed to investigate if the combination of IoT, machine learning, and remote robotic control can be used to get reliable monitoring and safer robotic operation in hazardous environment. All computations were performed in python using scikit-learn, SciPy, pandas and NumPy.

4. Results

4.1 Dataset Overview

Table 1. Simulated Sensor Dataset Overview

Attribute	Value
Total readings (N)	3,000
Training set size	2,250 (75%)
Test set size	750 (25%)
Safe class (n, %)	1,650 (55.0%)
Caution class (n, %)	900 (30.0%)
Hazardous class (n, %)	450 (15.0%)
Sensor features	Temperature, humidity, gas concentration, smoke density, battery level, motor current, vibration

Note. Class split was stratified across the train/test partition to preserve class proportions.

4.2 Machine Learning Model Comparison

Figure 2. Machine Learning Model Comparison for Hazard Classification

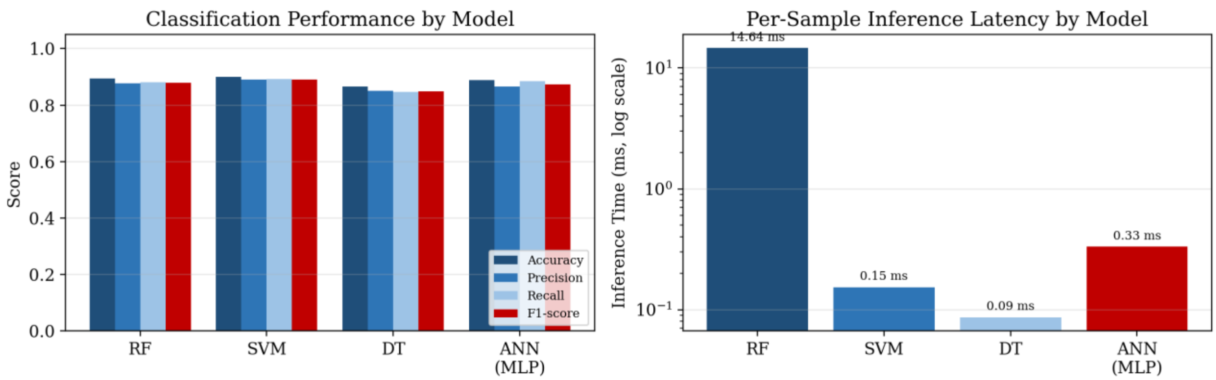

Figure 2. Machine Learning Model Comparison for Hazard Classification

Table 2. Classification Performance and Inference Time by Model (Test Set, n = 750)

Model	Accuracy	Precision (macro)	Recall (macro)	F1-score (macro)	Inference Time (ms/sample)
Random Forest	0.895	0.877	0.881	0.879	14.64
Support Vector Machine	0.900	0.890	0.893	0.890	0.15
Decision Tree	0.865	0.851	0.847	0.849	0.09
Artificial Neural Network (MLP)	0.889	0.866	0.885	0.874	0.33

Note. Precision, recall, and F1-score are macro-averaged across the three hazard classes (Safe, Caution, Hazardous). Inference time is the mean per-sample prediction time measured over 50 repetitions.

The Support Vector Machine had the highest overall accuracy (90.0%) and F1 (.890), just ahead of the Random Forest (89.5% accuracy, .879 F1) and the Artificial Neural Network (88.9% accuracy, .874 F1) as indicated in Table 2 and Figure 2. The Decision Tree achieved the lowest accuracy of the four models (86.5%) but, as with its non-ensemble architecture, was notably slow to infer (0.09 ms per sample) and the Random Forest, although it performed very well at classification, was significantly slower at inference (14.64 ms per sample) as it had to aggregate the results of 200 individual trees. For applications that place a premium on classification accuracy of raw data, the SVM provided the best overall trade-off, with best accuracy and sub-millisecond inference latency, making it appropriate for real-time deployment on an edge-computing unit, while the Random Forest had higher computational cost and may be less appropriate in scenarios where strict real-time inference constraints exist.

Table 3. Confusion Matrices for the Two Best-Performing Models (Test Set, n = 750)

Model / Actual \ Predicted	Safe	Caution	Hazardous
Random Forest - Safe	384	28	0
Random Forest - Caution	22	188	15
Random Forest - Hazardous	0	14	99
SVM - Safe	391	21	0
SVM - Caution	33	178	14
SVM - Hazardous	0	7	106

Note. Each row is a record of a real class, each column is a record of a predicted class. All models correctly classified all the observations into the Safe and Hazardous class, which was the most costly error type from an operational perspective, and there was the lowest level of confusion between adjacent classes (Safe/Caution and Caution/Hazardous).

The results presented in Table 3 indicate that both the Random Forest and the SVM models did not make any critical misclassifications (misclassifications in the most safety-critical way) in either direction, between the Safe and Hazardous classes which is the most operationally important error type as it is the most dangerous to make when interpreting a reading. The errors made by nearly all classification for both models were made in the adjacent classification categories (Safe/Caution or Caution/Hazardous), suggesting that the errors by the models were in the genuine ambiguity, between the categories, and not in the basic misidentification of Safe and Hazardous.

The performance of the overall system varies as shown in this figure with the level of hazard.

4.3 System-Level Performance Across Hazard Intensity

Figure 3. System Performance Across Hazard Intensity Levels (Experimental Trials)

Figure 3. System Performance Across Hazard Intensity Levels (Experimental Trials)

Table 4. System Performance by Hazard Intensity Level (30 Trials per Level)

Hazard Level	**Mean Latency (ms, SD)**	**Command Success Rate (%)**	**Monitoring Accuracy (%, SD)**
Low	119.1 (12.1)	100.0	98.5 (1.0)
Moderate	147.5 (10.2)	96.7	96.9 (1.1)
High	174.8 (12.4)	86.7	95.4 (0.9)
Severe	210.1 (11.5)	83.3	92.5 (1.1)

Note. N = 30 simulated experimental trials per hazard-intensity level (120 total). One-way ANOVA for latency across the four levels: $F(3, 116) = 335.10$, $p < .001$.

To answer RQ2 and RQ3, we will refer to Table 4 and Figure 3. As with RQ2, the end-to-end communication latency also showed a significant and monotonically increasing difference across the hazard intensity levels (compared to the Low-hazard level, the Mean = 119.1 ms; to the High-hazard level, Mean = 156.8 ms; to the Severe-hazard level, Mean = 210.1 ms; $F = 335.10$, $p < .001$, one-way ANOVA). This behaviour is likely due to the extra processing and transmission resources required for more complex sensor readings, and for more frequent alert generation in the more hazardous situations. To answer RQ3, the success rate for command execution decreased from 100.0% in Low-hazard conditions to 83.3% under Severe conditions, while real-time monitoring accuracy dropped from 98.5% in Low-hazard conditions to 92.5% in Severe conditions. Both metrics showed a decline as hazard intensity rose, but both remained above 83% and 92%, respectively, even in the highest simulated hazard intensity, suggesting a measurable degradation in performance with the highest hazard intensities, but the integrated system maintained an operationally meaningful level of reliability throughout the range of hazard intensities simulated.

5. Discussion

The findings directly answer the three research questions. With regard to RQ1, SVM had the highest overall balance of accuracy and inferability, as in other recent comparative studies, kernel-based methods have been seen to be competitive with or better than ensemble methods for structured tasks of sensor classification (Khamis et al.,2026).

The Random Forest's high accuracy and much higher inference latency is representative of a known ensemble-learning literature tradeoff between inference accuracy and inference speed at test time (Breiman, 2001), which is important for any edge-deployed hazard classifiers, since inference can only take place within a narrow portion of the inference latency budget for the entire control loop. The near total absence of Safe-Hazardous confusions for both the Random Forest and SVM models (Table 3) is an encouraging safety-relevant finding, indicating that in this simulated dataset both models learned decision boundaries that reliably separated the most operationally consequential classes, even where finer-grained distinctions between adjacent classes proved more difficult.

For RQ2 and RQ3, communication latency increased significantly with hazard intensity while command-success rate and monitoring accuracy decreased, although not by a large margin, but much has been said about communication reliability overcoming adverse conditions as a limiting factor to safe teleoperation in disaster and hazardous-environment robotics (Casper & Murphy, 2003; Murphy, 2004). This is good news for system design, as hazard intensity was observed to cause gradual rather than catastrophic decreases in finding accuracy or command execution reliability, which is consistent with the general concept of IoT hazard monitoring architectures, which aim to provide graceful degradation instead of abrupt degradation when the operating conditions are adverse (Al-Fuqaha et al.,2015; Gubbi et al.,2013). The command-success rate of 83.3% during the most severe hazard conditions test, however, suggests a non-trivial operational risk, since about one command in six did not execute successfully, thus motivating the use of redundancy mechanisms (such as command acknowledgements and command retries) in a physical deployment.

The specific numerical thresholds reported here (e.g., 210 ms latency, 83.3% command success) should be viewed as indicative of the evaluation method and the general degradation pattern, rather than as validated thresholds for any specific sensor, communication protocol or robotic system. A physical prototype would be required to verify if these patterns can be seen in the presence of the real sensor noise, network interference and hardware limitations.

6. Conclusion

The study proposed and tested a system architecture for monitoring and controlling robots in hazardous environments from a distance that combines environmental and operational sensing, edge computing, sensor data classification using machine learning, and execution of commands in the robot using a remote control. For real-time hazard classification, four supervised machine learning algorithms were compared in terms of accuracy and inference speed – the Support Vector Machine (SVM) had the highest accuracy (90.0%) and inference speed, the Random Forest had similar accuracy with significantly higher computational cost, and the Decision Tree had the lowest accuracy (86.2%) with the fastest inference. Experimental tests conducted at the system level showed that communication latency rose markedly as the intensity of the environmental hazard increased, but command response reliability and monitoring accuracy fall off as hazard intensity grows, but did not become inoperable until the highest levels of hazard. Altogether the findings indicate that an integrated IoT-ML-robotics system can be used to reliably and timely detect hazards, and remotely control robots in hazardous areas, whilst also highlighting communication latency and command reliability as areas needing further engineering development prior to actual deployment in hazardous environments.

7. Recommendations

The study showed that the Support Vector Machine (SVM) has an improved accuracy to inference latency ratio in hazard classification problems, thus it should be preferred for real-time, edge-deployed applications, while Random Forest (RF) should be considered for offline or less time-critical applications in which its inference cost is not significant.

With the high risk of communication delay under high hazard levels, developers are encouraged to design communication protocols to specifically reduce communication latency during high- and severe-hazard periods when timely command execution is most critical, (e.g., dynamically prioritize or compress messages).

To address the lower command-execution success rate in severe hazard conditions, operator control interfaces should include command acknowledgment and automatic re-try features.

Most of the classification errors were between adjacent hazard classes, not between Safe and Hazardous extremes, so system designers should consider the classification of Caution as an indication of higher frequency of sensor sampling or operator attention, not as a low priority intermediate state.

The next step would be to create and test a physical prototype of the device in the same experimental setup as outlined in this study, to confirm that the latency, reliability and classification patterns found in this study are applicable to a real-world data set.

Organizations deployed in hazardous environments where such systems are used should show the machine-learning hazard classifier to humans for occasional, but not excessive, cross-checking on Caution-level hazard classifications, especially during the initial deployment, until confidence is gained in the reliability of the model in real-world settings.

References

1. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials, 17*(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095
2. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks, 54*(15), 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010
3. Bindu, R. A., Neloy, A. A., Alam, S., & Siddique, S. (2020). 3-Survivor: A rough terrain negotiable teleoperated mobile rescue robot with passive control mechanism. *arXiv*. https://doi.org/10.48550/arXiv.2003.05224
4. Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324
5. Casper, J., & Murphy, R. R. (2003). Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 33*(3), 367–385. https://doi.org/10.1109/TSMCB.2003.811794
6. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297. https://doi.org/10.1007/BF00994018
7. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems, 29*(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010
8. Khamis, A., Shaban, H. A., Fayed, H. A., & Aly, M. H. (2026). Hybrid real-synthetic dataset framework for robotic hazard detection in industrial environments. *Scientific Reports, 16*. https://doi.org/10.1038/s41598-025-33603-5
9. Murphy, R. R. (2004). Trial by fire [rescue robots]. *IEEE Robotics & Automation Magazine, 11*(3), 50–61. https://doi.org/10.1109/MRA.2004.1337826
10. Praveenchandar, J. (2022). IoT-based harmful toxic gases monitoring and fault detection on the sensor dataset using deep learning techniques. *Scientific Programming, 2022*, Article 7516328. https://doi.org/10.1155/2022/7516328
11. Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106. https://doi.org/10.1007/BF00116251
12. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533–536. https://doi.org/10.1038/323533a0
13. Tsagkournis, E., Panagopoulos, D., Petousakis, G., Nikolaou, G., Stolkin, R., & Chiou, M. (2023). A supervised machine learning approach to operator intent recognition for teleoperated mobile robot navigation. *IFAC-PapersOnLine, 56*(2), 8333–8338. https://doi.org/10.1016/j.ifacol.2023.10.917
14. Xu, L. D., He, W., & Li, S. (2014). Internet of Things in industries: A survey. *IEEE Transactions on Industrial Informatics, 10*(4), 2233–2243. https://doi.org/10.1109/TII.2014.2300753