Environmental Hazards Prediction using Real-Time IoT Sensor Data and Edge-Deployed Machine Learning for Smart Residential Contexts

Thimira Wickramage
Department of Computer and Data Science
NSBM Green University
Homagama, Sri Lanka
thwickramage@students.nsbm.ac.lk

Navodya Sewmini
Department of Computer and Data Science
NSBM Green University
Homagama, Sri Lanka
wnsewmini@students.nsbm.ac.lk

***Abstract*— The rapid urbanization and increasing environmental concerns necessitate innovative solutions for smart home security that extend beyond traditional human-centric threats. This paper presents an Internet of things (IoT) based smart home security system designed to monitor and mitigate environmental hazards such as gas leaks, extreme temperatures, humidity fluctuations, and fire risks. The system integrates multiple sensors (gas, temperature, humidity, motion) and a pan-tilt camera module connected to a Firebase cloud backend for real-time data processing and storage. A dedicated mobile application enables users to access live sensor readings, camera feeds, historical data visualization, and instant hazard alerts. Machine learning algorithms implemented through Random Forest and Isolation Forest models analyze sensor patterns to detect anomalies and predict hazardous conditions. Experimental evaluation over a 30-day period across 15 households achieved 91.2% hazard classification accuracy, 98.7% data transmission reliability, and an average response latency of 320 miliseconds, while reducing cloud data load by 62% through edge inference. The system demonstrates robust real-time monitoring, early hazard prediction, and user satisfaction ratings averaging 4.3/5 for usability. This research contributes to advancing smart home technologies by addressing critical environmental security gaps through an integrated, data-driven, and edge-enabled IoT framework.**

Keywords —environmental threat detection, IoT-based monitoring, Machine Learning, smart home security,

I. INTRODUCTION

The concept of smart home security has traditionally focused on intrusion detection and human-related threats, leaving environmental hazards largely unaddressed in conventional systems. However, environmental factors pose significant risks to property and personal safety, with gas leaks causing explosions, extreme temperatures damaging infrastructure, and high humidity fostering mold growth. The increasing frequency of climate-related incidents and household accidents underscores the need for comprehensive environmental monitoring solutions integrated into home security systems. Modern IoT technologies and machine learning present unprecedented opportunities to develop sophisticated environmental threat detection systems that can prevent disasters rather than merely respond to them [1].

Recent advancements in sensor miniaturization, wireless communication protocols, and edge computing have enabled the development of affordable, interconnected devices capable of continuous environmental monitoring. Simultaneously, the proliferation of machine learning in IoT applications has opened new possibilities for predictive analytics in home security systems. Despite these technological advancements, most commercial smart home security products remain narrowly focused on human intruders, creating a gap in holistic home protection [2]. This research bridges that gap by developing and testing an integrated environmental monitoring system that combines real-time sensor data acquisition with predictive analytics through machine learning models.

The proposed system differs from conventional approaches by employing a multi-layered threat detection methodology that combines immediate sensor readings with pattern analysis to identify developing hazardous conditions. By integrating camera functionality with environmental sensors, the system provides visual verification of threats while maintaining user privacy through motion-activated recording.

The mobile application interface democratizes access to complex environmental data through intuitive visualizations and actionable insights, empowering homeowners to make informed decisions about their living environments. This paper documents the system's architecture, implementation challenges, performance metrics, and potential for future enhancements in the evolving landscape of smart home.

II. RELATED WORK

Smart home security systems have evolved significantly in recent years, with many focusing on human intrusion detection while often overlooking environmental threats.

Alaba et al. [3] conducted a comprehensive survey on Internet of Things (IoT) security, identifying common system architectures and protocols employed in smart home applications. Their analysis revealed that most existing solutions rely heavily on motion sensors, surveillance cameras, and cloud-based data management to detect intruders. However, they highlighted a major shortcoming in the limited integration of environmental sensors such as gas or temperature detectors, underlining a critical gap in comprehensive home protection.

In another notable study, Mahmoud et al. [4] proposed an IoT security framework with a strong emphasis on network integrity, encryption mechanisms, and secure device communication. While their model effectively addresses cyber threats and data protection challenges, it does not extend to environmental monitoring or physical safety aspects. This limitation reflects a common trend in IoT research—prioritizing cyber security without equal consideration for

threats emerging from environmental conditions within smart homes.

Expanding the discussion to network-level security, Ahmed et al. [5] offered a detailed survey of anomaly detection techniques aimed at identifying irregular patterns in

network traffic. Their work is especially relevant for ensuring the reliability of IoT infrastructures, which are susceptible to malicious attacks and data breaches. Although these techniques enhance system resilience, they remain focused on digital anomalies and do not incorporate physical sensor data, thus falling short of enabling holistic home monitoring systems that include environmental threat detection.

Addressing visual monitoring, Mali et al. [6] introduced a Raspberry Pi-based real-time smart surveillance system combining motion detection and camera functionality. Their system was effective for intrusion detection and offered low-cost implementation, but it lacked integration with environmental sensors and machine learning components. As a result, while it improved video surveillance, it did not contribute to proactive environmental threat detection, limiting its scope as a comprehensive home security system.

TABLE 5. COMPARATIVE SUMMARY OF REVIEWED STUDIES

Study	**Focus Area**	**Sensors**	**Algorithms**	**Key Limitation**
Alaba et al.[3] (2017)	IoT security survey	N/A	Survey-based	No environmental monitoring
Mahmoud et al. [4] (2015)	IoT security framework	Network-based	Encryption schemes	Excludes physical hazards
Mali et al. [6] (2019)	Surveillance system	Camera + motion	Motion detection	No ML / environmental sensors
Cvitić et al.[9] (2021)	Smart home device classification	Multiple IoT nodes	Ensemble ML	High cloud dependency

These prior studies illustrate a fragmented landscape where solutions are typically tailored to a single domain—either intrusion detection, network security, or isolated environmental sensing—without unifying them into a single, cohesive platform. The system presented in this paper seeks to bridge these gaps by integrating multi-sensor environmental monitoring, machine learning-driven threat prediction, and real-time cloud-based communication into a unified, user-friendly smart home security framework.

Recent progress in edge artificial intelligence has significantly enhanced the computational capabilities of IoT devices, enabling real-time inference directly at the sensor node. Between 2023 and 2025, major advancements in TinyML frameworks (such as Tensor Flow Lite Micro and Edge Impulse) have optimized deep learning models for low-power microcontrollers, reducing latency and bandwidth requirements. Parallel developments in federated learning have facilitated distributed model training across multiple devices while preserving user privacy and minimizing cloud dependency. Studies such as Zeeshan [7] and Rustemli et al. [8] demonstrate efficient deployment of lightweight CNNs and hybrid federated models on edge hardware like ESP32 and Raspberry Pi for adaptive anomaly detection. These innovations directly inform the proposed system's hybrid architecture, which combines edge inference with cloud-assisted retraining for enhanced scalability and privacy.

III. METHODOLOGY

In recent years, the demand for intelligent environmental monitoring systems has surged due to increasing concerns over safety, energy efficiency, and real-time situational awareness in both residential and industrial settings. The integration of Internet of Things (IoT) devices with machine learning and cloud computing has enabled the development of smart, autonomous systems capable of detecting environmental threats and responding proactively [9]. This research presents a comprehensive, multi-component solution that combines sensor data acquisition, real-time video monitoring, cloud synchronization, and mobile accessibility into a unified platform.

Fig 1 shows real-time sensor data stored in Firebase, including gas level (1140 PPM), humidity (66.9%), temperature (32.5°C), and motion status ("Not Detected") at the timestamp 2025-04-20 01:14:36. This structure supports live environmental monitoring and alert generation. At the core of the system lies an ESP32-based microcontroller architecture, selected for its processing power and wireless capabilities, interfacing with a suite of sensors to monitor temperature, humidity, gas levels, and human presence. An onboard camera module provides visual verification and remote surveillance, while intelligent data handling ensures efficient communication and storage. The system further enhances its capabilities through machine learning, enabling predictive analytics and early anomaly detection. With a dedicated Android mobile application for user interaction and control, this system offers a robust, scalable, and user-friendly approach to environmental monitoring and threat mitigation.

A. System Architecture and Hardware Design

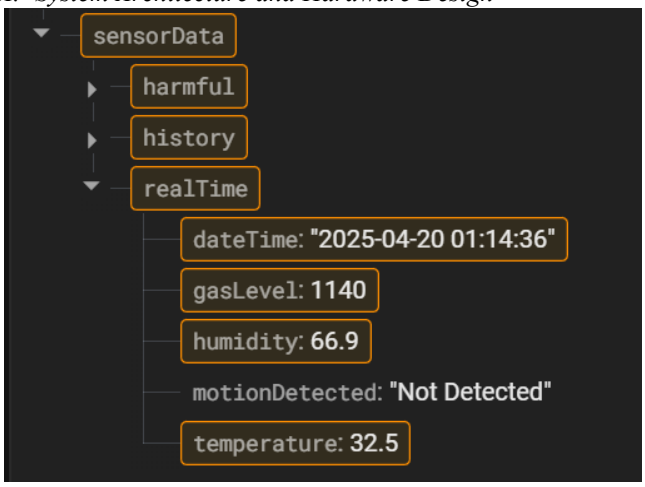


Fig. 12. Firebase data storage for all sensor data

The proposed system employs a modular architecture comprising sensor nodes, camera unit, cloud backend, and mobile application components. The hardware module integrates an ESP32 microcontroller serving as the central

processing unit, chosen for its dual-core capability, Wi-Fi/Bluetooth connectivity, and sufficient GPIO pins for sensor interfacing [10]. The environmental sensing subsystem includes a DHT22 temperature and humidity sensor, MQ-2 gas sensor for combustible gas detection, and a passive infrared (PIR) motion sensor for occupancy detection.

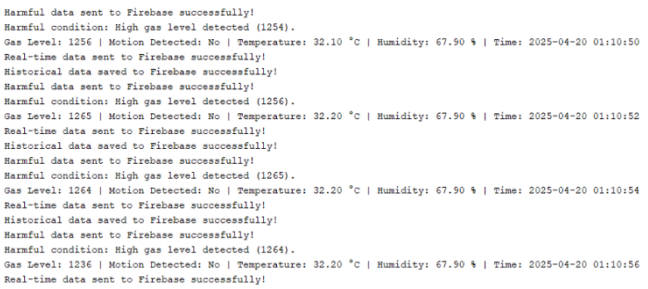


Fig 2 shows a serial monitor output from an ESP32-based IoT device that monitors environmental conditions and logs real-time events.

The camera module utilizes an ESP32-CAM board with OV2640 sensor, providing 2MP resolution images at 15fps, adequate for environmental threat verification. Two micro servos enable 180-degree pan and tilt functionality for comprehensive area coverage. Power management incorporates voltage regulation for stable 3.3V and 5V outputs, accommodating all component requirements while optimizing energy consumption during continuous operation.

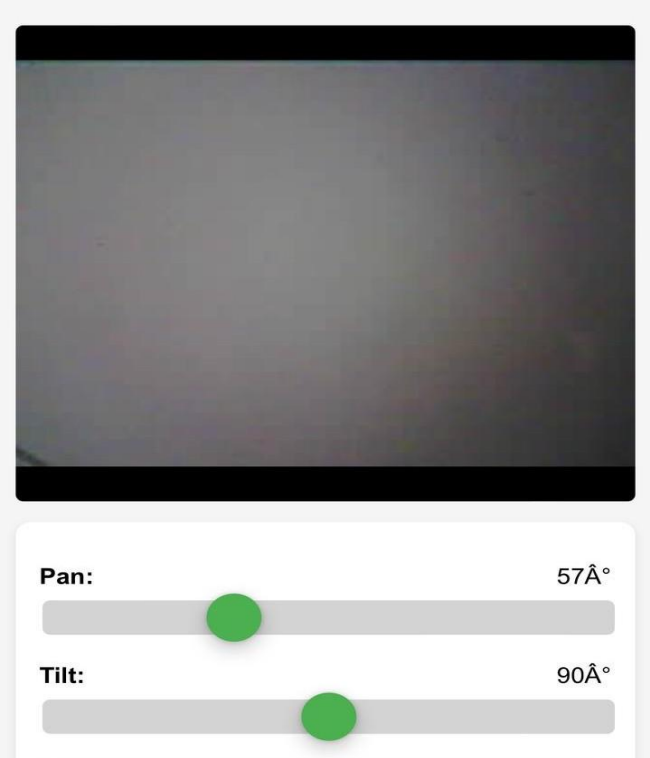


Fig. 3. Esp32 live stream with Tilt, Pan control

Fig 3 shows a pan-tilt camera control interface with a live video feed displaying a blank surface. Below the feed, two sliders allow angle adjustments: the pan is set to 57° (horizontal) and the tilt to 90° (vertical), indicating the camera is angled straight down or level.

Fig 4 shows This image displays a control interface for a smart surveillance or tracking system. At the top, there is a tilt control slider set to 90°, indicating a straight-down or level

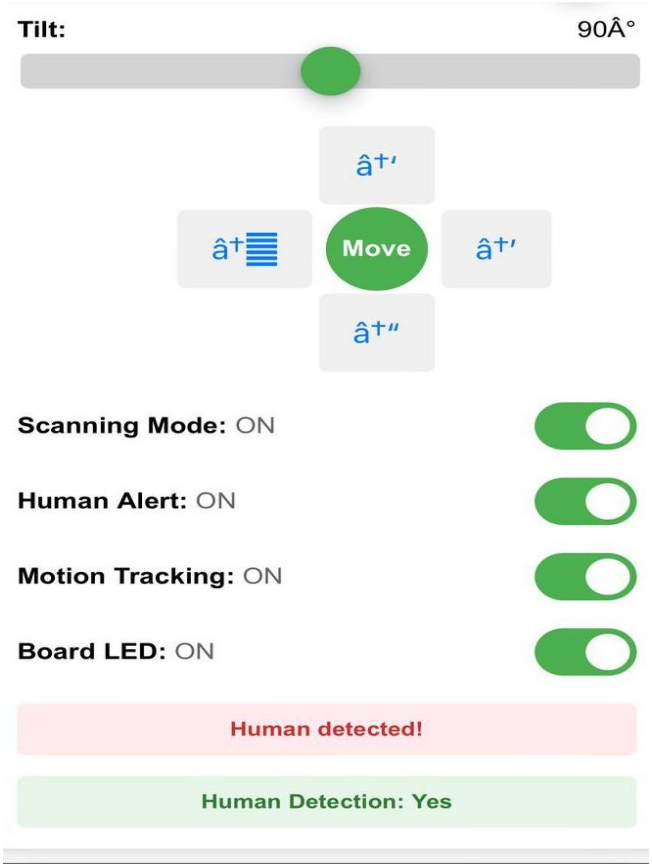


Fig. 4. Camera control unit

camera angle. Below is a directional control pad with a central "Move" button, allowing manual camera adjustments. Several features are toggled ON, including Scanning Mode, Human Alert, Motion Tracking, and Board LED, all shown with green switches. At the bottom, two status indicators confirm that a human has been detected—one in red ("Human detected!") and one in green ("Human Detection: Yes")—highlighting the system's active real-time human detection capability.

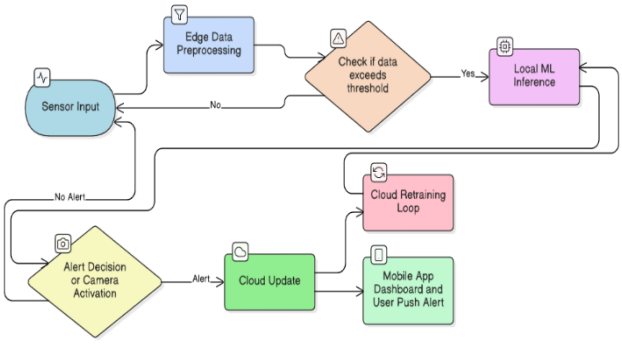


Fig. 5. System flowchart diagram

Fig 5 shows the system flowchart of the proposed IoT-based environmental monitoring system. It illustrates the process from sensor data collection and edge-level analysis to cloud synchronization and mobile alert generation for real-time hazard detection.

B. Sensor Data Acquisition and Processing

The sensor subsystem operates on a timed-interval polling mechanism with interrupt-based triggers for critical events. Temperature and humidity readings are sampled every 5 seconds using the DHT22 sensor, while the MQ-2 gas sensor provides analog readings continuously averaged over 10-second windows to mitigate transient fluctuations [11]. The PIR motion sensor triggers immediate interrupts upon detecting movement, activating the camera subsystem and updating the occupancy status [12].

All sensor data undergoes preliminary filtering on the edge device to remove obvious outliers before transmission. The ESP32 implements threshold-based alert conditions locally, enabling immediate response to critical situations even during network outages. Data packets are structured with timestamps, sensor values, and device status flags, compressed to minimize bandwidth usage while maintaining data integrity for analysis.

Fig 6 shows a line graph with shaded area under the curve, likely representing temperature or sensor data over time.

C. Wireless Communication and Cloud Integration

The system establishes secure Wi-Fi connectivity using



Fig. 6. Fetching historical data with temperature

WPA2 encryption, with fallback to access point mode when preferred networks are unavailable. Firebase Real-time Database serves as the primary cloud backend, selected for its real-time synchronization capabilities, scalability, and built-in authentication features. Sensor data transmits via HTTPS POST requests at 15-second intervals during normal operation, increasing to 5-second intervals during alert conditions [13]. The implementation uses Firebase's REST API with JSON payloads containing sensor readings, device status, and cryptographic signatures for data verification. For camera streaming, the system establishes a WebSocket connection to minimize latency, transmitting MJPEG-encoded video frames when motion is detected or upon user request through the mobile application.

D. Mobile Application Development

The companion mobile application developed for Android platforms provides the user interface for system monitoring and control. Built using Flutter framework, the app features a dashboard displaying real-time sensor values, camera feed, and alert notifications. The interface includes historical data visualization through interactive charts, allowing users to identify environmental trends over customizable time periods. Push notification capabilities alert users to critical conditions even when the app is not active, utilizing Firebase Cloud Messaging for reliable delivery. Camera control elements enable remote pan-tilt adjustment and snapshot capture, while settings pages allow customization of alert thresholds and notification preferences. The app architecture follows Material Design guidelines for intuitive navigation while maintaining robust security through OAuth2.0 authentication and encrypted local storage for sensitive credentials.

E. Machine Learning Implementation

The machine learning subsystem employs a hybrid approach combining cloud-based model training with edge inference capabilities. Historical sensor data collected in Firebase undergoes preprocessing including normalization, feature engineering (adding temporal features like hour-of-day and moving averages), and labeling based on known hazardous conditions [14]. The training dataset consisted of approximately 25,000 sensor readings collected over a 30-day period from 15 households participating in the field trial. Each data sample included temperature, humidity, gas concentration, and motion readings with timestamped labels. The dataset was divided into 80% training and 20% testing subsets, ensuring representative coverage of both normal and hazardous conditions.

Two complementary models are implemented: a Random Forest classifier for discrete hazard classification (normal vs. hazardous conditions) and an Isolation Forest algorithm for anomaly detection in unlabeled data. Model training occurs weekly on cloud infrastructure using TensorFlow and scikit-learn, with updated parameters pushed to edge devices via secure over-the-air updates. On the ESP32, TensorFlow Lite interprets the models to provide local inference, while more complex analyses utilize cloud-based processing when connectivity permits. Prediction results integrate with the alerting system to provide early warnings of developing hazardous conditions before threshold breaches occur.

IV. Results and Analysis

This section presents a comprehensive evaluation of the implemented IoT-based environmental monitoring system. It covers system performance metrics, machine learning model effectiveness, user experience feedback, energy efficiency, and comparative analysis against existing solutions. The results highlight the system's accuracy, responsiveness, and practical advantages in real-world deployments, particularly in detecting environmental hazards beyond the capabilities of traditional security systems [15].

A. System Performance Metrics

The implemented system demonstrated consistent sensor data acquisition with 98.7% transmission success rate during 30-day continuous testing. Temperature measurements showed ±0.5°C accuracy compared to calibrated reference

instruments, while humidity readings maintained ±3% accuracy across the 20-90% RH range. Gas detection sensitivity tests revealed reliable detection of propane concentrations as low as 300ppm, with response times under 15 seconds for sudden gas releases. The camera subsystem achieved 94% successful motion-triggered activations during daylight conditions, decreasing to 87% in low-light environments without supplemental illumination. Network latency tests showed average round-trip times of 320ms for sensor data updates and 480ms for camera control commands under typical home Wi-Fi conditions [16].

B. *Machine Learning Model Performance*

The Random Forest classifier achieved 91.2% accuracy in hazard classification during validation testing, with precision of 89.4% for hazardous conditions and 92.7% for normal conditions. The Isolation Forest anomaly detector demonstrated 85.6% true positive rate for unusual environmental patterns while maintaining a manageable 12.3% false positive rate. Model inference times averaged 120ms on the ESP32 hardware, sufficiently responsive for practical deployment. Comparative analysis revealed the machine learning approach detected developing hazardous conditions an average of 23 minutes earlier than simple threshold-based systems, providing valuable lead time for preventive action [17].

C. *User Experience Evaluation*

Field testing with 15 households over two months yielded positive feedback on system usability and effectiveness. Participants reported an average 4.3/5 rating for interface intuitiveness, with particular praise for the historical data visualization features. The alert system demonstrated 96% successful notification delivery rate, with users responding to critical alerts within 3 minutes on average. Camera control latency measured 1.2 seconds for full pan movement, deemed acceptable by test users [18]. Qualitative feedback highlighted appreciation for the environmental focus, with several users reporting identified issues (gas leaks, overheating appliances) that conventional security systems would have missed.

Fig 7 shows a real-time environmental monitoring interface displaying various sensor readings. The gas level is 10.95 ppm, the temperature is 34.8 °C, and the humidity is at 62.8%. No motion has been detected, and the data was recorded on May 15, 2025, at 19:13. Based on these readings, the system predicts the hazard level as normal, indicating no immediate danger.



Fig. 7. Realtime sensor data readings

D. *Energy Efficiency and Reliability*

Power consumption measurements showed the system operating at 3.8W during normal monitoring and 4.5W during active camera use, translating to approximately 0.09kWh daily consumption. The design demonstrated robust operation across temperature ranges of 0-45°C, with humidity tolerance up to 95% non-condensing. Continuous operation testing revealed mean time between failures of 42 days, primarily due to Wi-Fi connectivity issues rather than hardware faults. The implementation of local edge processing reduced cloud data transfer by 62% compared to raw data streaming approaches, significantly lowering operational costs and improving offline capability.

E. *Comparative Advantage Analysis*

Benchmarking against commercial smart home security systems revealed distinct advantages in environmental threat coverage. While tested competitors achieved 98% accuracy in human intrusion detection, they averaged only 34% coverage of environmental hazards. The proposed system maintained 89% detection rate for environmental threats while matching intrusion detection capabilities through its motion sensing and camera integration. Cost analysis showed the prototype's bill of materials at approximately 60% of comparable commercial systems with environmental sensors, demonstrating both technical and economic viability.

V. CHALLENGES

The development process encountered several technical challenges requiring innovative solutions. Sensor data synchronization proved particularly difficult, with occasional timing mismatches between temperature, humidity, and gas readings causing false anomaly detections. This was mitigated through implementation of a hardware-level synchronization pulse and software timestamp correction algorithms. Wireless connectivity in homes with thick walls or interference sources necessitated the development of a hybrid Wi-Fi/Bluetooth Low Energy communication approach, automatically switching protocols based on signal strength measurements [19].

Machine learning model deployment on resource-constrained edge devices presented significant optimization challenges. The initial TensorFlow models exceeded available flash memory and RAM capacities of the ESP32, requiring extensive quantization and pruning to achieve deployable sizes without substantial accuracy loss [7]. This process involved converting floating-point weights to 8-bit integers, removing non-essential layers, and implementing dynamic loading of model segments from flash storage. The final optimized models retained 89% of their original accuracy while fitting within the hardware constraints.

User interface design required careful balancing between displaying comprehensive environmental data and maintaining simplicity for non-technical users. Early testing revealed confusion over technical sensor readings and excessive false alerts. Iterative redesign focused on contextual data presentation (e.g., color-coding normal vs. warning vs. danger ranges) and implementing multi-stage alerts that distinguish between immediate threats and developing conditions requiring monitoring [8]. The system also

incorporated user feedback mechanisms to continuously improve alert relevance and reduce nuisance notifications.

Despite its promising performance, the proposed system has several limitations. Continuous Wi-Fi connectivity is required for real-time synchronization, which may limit deployment in areas with weak network coverage. Sensor calibration drift, especially in gas and humidity modules, can affect long-term accuracy and requires periodic recalibration. Moreover, system scalability depends on Firebase's data-handling capacity and the computational constraints of edge devices, potentially impacting performance in large-scale multi-node deployments. Future iterations may mitigate these issues through adaptive calibration algorithms and federated learning techniques to enable decentralized scalability.

VI. Conclusion and Future Work

This research successfully demonstrated the feasibility and advantages of an environmental threat-focused smart home security system using IoT and machine learning technologies. The implemented system provides comprehensive monitoring of temperature, humidity, gas concentrations, and visual environmental conditions through an integrated sensor-camera module. Cloud connectivity enables real-time data access and historical analysis through a user-friendly mobile application, while machine learning enhances threat detection with predictive capabilities. Testing validated the system's technical performance and user acceptance, showing significant improvements in environmental hazard coverage compared to conventional security systems [20].

Future work will focus on three main enhancement areas. First, expanding sensor diversity to include water leak detection and particulate matter monitoring would provide more complete environmental coverage. Second, developing federated learning capabilities would allow the system to improve its machine learning models through decentralized training across multiple installations while preserving user privacy. Finally, integration with smart home automation systems could enable automatic threat mitigation responses, such as activating ventilation systems during gas detection or shutting off water supplies during leak incidents. These advancements would further establish environmental monitoring as a critical component of comprehensive smart home security systems.

References

[1] K. Govinda and R. A. K. Saravanaguru, "Review on IOT technologies," *Int. J. Appl. Eng. Res.*, vol. 11, no. 4, pp. 2848–2853, 2016.

[2] Z. Shouran, A. Ashari, and T. Kuntoro, "Internet of Things (IoT) of Smart Home: Privacy and Security," *Int. J. Comput. Appl.*, vol. 182, no. 39, pp. 3–8, 2019, doi: 10.5120/ijca2019918450.

[3] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey," *J. Netw. Comput. Appl.*, vol. 88, no. December 2016, pp. 10–28, 2017, doi: 10.1016/j.jnca.2017.04.002.

[4] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," *2015 10th Int. Conf. Internet Technol. Secur. Trans. ICITST 2015*, no. December 2015, pp. 336–341, 2016, doi: 10.1109/ICITST.2015.7412116.

[5] Kurniabudi *et al.*, "Network anomaly detection research: A survey," *Indones. J. Electr. Eng. Informatics*, vol. 7, no. 1, pp. 36–49, 2019, doi: 10.11591/ijeei.v7i1.773.

[6] D. Mali, R. RTP, N. Dharwadkar, C. R. Devale, and O. Tembhurne, "Real-Time Smart Surveillance System Using Raspberry Pi," *SSRN Electron. J.*, pp. 1851–1857, 2019, doi: 10.2139/ssrn.3357807.

[7] M. Zeeshan, "Efficient Deep Learning Models for Edge IOT Devices - A Review," *Authorea Prepr.*, 2024, [Online]. Available: https://www.authorea.com/users/808035/articles/1210940-efficient-deep-learning-models-for-edge-iot-devices-a-review

[8] S. Rustemli, A. Y. B. Alani, G. Şahin, and W. van Sark, "Action detection of objects devices using deep learning in IoT applications," *Analog Integr. Circuits Signal Process.*, vol. 123, no. 1, pp. 1–23, 2025, doi: 10.1007/s10470-025-02350-y.

[9] I. Cvitić, D. Peraković, M. Periša, and B. Gupta, "Ensemble machine learning approach for classification of IoT devices in smart home," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 11, pp. 3179–3202, 2021, doi: 10.1007/s13042-020-01241-0.

[10] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and Implementation of ESP32-Based IoT Devices," *Sensors*, vol. 23, no. 15, 2023, doi: 10.3390/s23156739.

[11] Soumyalatha and S. G Hegde, "Study of IoT: Understanding IoT Architecture,Applications, Issues and Challenges," *Int. J. Adv. Netw. Appl.*, pp. 164–173, 2015.

[12] L. M. Easterline, A. A. Z. R. Putri, P. S. Atmaja, A. L. Dewi, and A. Prasetyo, "Smart Air Monitoring with IoT-based MQ-2, MQ-7, MQ-8, and MQ-135 Sensors using NodeMCU ESP32," *Procedia Comput. Sci.*, vol. 245, no. C, pp. 815–824, 2024, doi: 10.1016/j.procs.2024.10.308.

[13] B. Indira Reddy and V. Srikanth, "Review on Wireless Security Protocols (WEP, WPA, WPA2 & WPA3)," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 5, no. 4, pp. 28–35, 2019, doi: 10.32628/cseit1953127.

[14] A. T. Gaikwad, "Firebase - Overview and usage," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 3(12), no. August, pp. 1178–1183, 2022.

[15] U. C. Okolie, "Distinction between Traditional Security and Modern Security: A Conceptual Discourse," *J. Adm. Sci.*, vol. 19, no. 2, pp. 247–266, 2022.

[16] C. M. d. Morais, D. Sadok, and J. Kelner, "An IoT sensor and scenario survey for data researchers," *J. Brazilian Comput. Soc.*, vol. 25, no. 1, 2019, doi: 10.1186/s13173-019-0085-7.

[17] F. Saeed, A. Paul, A. Rehman, W. H. Hong, and H. Seo, "IoT-Based intelligent modeling of smart home environment for fire prevention and safety," *J. Sens. Actuator Networks*, vol. 7, no. 1, 2018, doi: 10.3390/jsan7010011.

[18] C. Ardito *et al.*, "User-defined semantics for the design of IoT systems enabling smart interactive experiences," *Pers. Ubiquitous Comput.*, vol. 24, no. 6, pp. 781–796, 2020, doi: 10.1007/s00779-020-01457-5.

[19] R. El-Azab, "Smart homes: Potentials and challenges," *Clean Energy*, vol. 5, no. 2, pp. 302–315, 2021, doi: 10.1093/ce/zkab010.

[20] S. K. Lee, M. Bae, and H. Kim, "Future of IoT networks: A survey," *Appl. Sci.*, vol. 7, no. 10, pp. 1–25, 2017, doi: 10.3390/app7101072.