SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15478
e-ISSN : 2541-2019
p-ISSN : 2541-044X

IoT Sensor Data Analysis for Early Fire Detection Using Dynamic Threshold

Widia Br Tarigan[1)*], **Robet**[2)], **Feriani Astuti Tarigan**[3)]
[1)2)3)] Department of Informatics, STMIK Time, Medan, Indonesia
[1)*]widiatarigan2312@gmail.com, [2)] robertdetime@gmail.com, [3)] ferianiastutitime@gmail.com

Submitted :Oct 23, 2025 | **Accepted** : Nov 10, 2025 | **Published** : Jan 02, 2026

Abstract: Early fire detection using Internet of Things (IoT) technology plays a vital role in minimizing potential material losses and casualties. Conventional systems generally still rely on static thresholds that are less adaptive to environmental dynamics, leading to high false alarm rates and delayed detection. This study proposes a dynamic threshold approach based on a hybrid method of Fuzzy Logic–Random Forest–Adaptive Z-Score and compares it with the static threshold method. Testing was conducted using publicly available secondary datasets, and the algorithms were implemented and tested in Jupyter Notebook. Evaluation was performed using accuracy, false alarm rate (FAR), detection time, F1-score, precision, and recall metrics. The test results show that the dynamic threshold method provides better performance with an increase in accuracy from 59.5% to 74.8%, a decrease in FAR from 31.1% to 14.3%, and a reduction in detection time from 21 seconds to 0 seconds. In addition, the F1-score increased from 0.459 to 0.638, precision from 0.473 to 0.716, and recall from 0.446 to 0.575. These results show that the dynamic threshold approach is more adaptive and reliable in IoT-based fire detection systems than conventional static threshold methods.

Keywords: IoT, dynamic threshold, Fuzzy Logic, machine learning, Adaptive Z-Score

INTRODUCTION

Fire is a disaster with significant economic and human safety impacts. Fire is a serious threat that can cause major financial losses and significant casualties. According to data from the National Disaster Management Agency (BNPB), more than 600 fire incidents occur each year in Indonesia, resulting in total material losses exceeding IDR 15 trillion. Early detection is an essential factor in reducing the impact of fires; a one-minute delay can increase the burned area by up to 150%.

Internet of Things (IoT) technology offers significant opportunities for real-time fire monitoring systems, but it still faces challenges in terms of accuracy and reliability. One of the main challenges is the high rate of false alarms, which can reach 15–28% in dynamic environmental conditions (Morchid et al., 2024), (Abdullahi et al., 2025). False alarms not only disrupt system operations but also reduce user confidence, which can have fatal consequences if a real fire occurs.

Previous research has identified three significant limitations in IoT-based fire detection systems. First, static threshold approaches use fixed thresholds (e.g., temperature > 60°C or smoke > 400 ppm), which cannot adapt to environmental changes or to human activities such as cooking or daily temperature fluctuations (Li & Sun, 2023). Second, vision-based methods such as Convolutional Neural Networks (CNNs) require line-of-sight and perform poorly in closed or foggy conditions, with accuracy decreasing by up to 40% when visibility is limited (Khan et al., 2025). Third, machine learning algorithms such as Random Forests can improve accuracy but still rely on static features and cannot adapt to sensor dynamics (Ding et al., 2023).

This study aims to develop a dynamic threshold based on a hybrid Fuzzy Logic–Machine Learning approach and to compare its performance with that of the static threshold method using a secondary dataset. The dataset was processed and tested using Jupyter Notebook for algorithm modeling and visualization of results (Vaegae et al., 2024).

This research contributes in three main ways. First, it proposes an innovative dynamic threshold architecture that combines Fuzzy Logic (for uncertainty interpretation), Random Forest (for parameter optimization), and Adaptive Z-Score (for real-time adaptive calculation). Second, it provides a comprehensive comparative analysis between static and dynamic thresholds based on accuracy, false alarm rate, and detection time metrics. Third, it

*name of corresponding author

proves that the dynamic threshold method can reduce false alarms by more than 50% and significantly improve accuracy on realistic data.

LITERATURE REVIEW

Research on early fire detection based on the Internet of Things (IoT) has developed in several main directions, including: (i) sensor-based approaches with static thresholds, (ii) image or video-based approaches (vision-based), (iii) machine learning on continuous sensor data (sensor streams), and (iv) hybrid approaches that combine fuzzy logic, sensor fusion, and time-series-based adaptive thresholds. This literature review examines the strengths and weaknesses of each approach and clarifies the position and contribution of this research within the existing literature.

Sensor-Based Approach with Static Threshold

The fixed threshold-based approach (e.g., temperature > t, PM2.5 > s, or gas > c) is a commonly used classical method because it is inexpensive, simple, and deterministic with a fast response time. However, its weakness lies in the system's inability to adapt to environmental variations, such as daily temperature changes, human activity, or disturbances in air ventilation. These baseline changes often lead to increased false alarms and delayed detection when the same threshold values are applied across different locations (Park et al., 2023). In addition, the univariate (single-sensor-based) approach is unable to capture complex multivariate relationships, such as the combination of a slow rise in temperature, increased particulates, and gas emissions in the early stages of a fire. These limitations have led to the need for adaptive threshold systems that can dynamically adjust to environmental conditions (Chan et al., 2024).

Vision-Based Detection Approach

Vision approaches using images or videos have shown rapid progress with the advent of modern Convolutional Neural Network (CNN) models such as ResNet, YOLO, and Faster R-CNN (Bonilla-Ormachea et al., 2025). These models can visually detect fire or smoke with high accuracy under good lighting and visibility conditions. However, their effectiveness decreases dramatically in enclosed spaces, foggy conditions, or low-light environments. In addition, hardware requirements such as high-resolution cameras and intensive computing make this approach less efficient for large-scale or IoT-based fire detection systems with limited resources. Therefore, in most industrial environments and enclosed buildings, environmental sensor systems are still the preferred choice because they are stable and power-efficient (Li & Sun, 2023).

Machine Learning Approach to Sensor Streams

The application of machine learning algorithms, such as Random Forest and Support Vector Machine (SVM), and sequence-based models, such as Long Short-Term Memory (LSTM), has improved the accuracy of fire pattern detection (Ma et al., 2025). These algorithms can learn correlations between sensors and recognize complex patterns in real-time data. However, most studies still use static threshold-based final decisions on model output scores. This causes the system to remain non-adaptive to concept drift—that is, changes in the data distribution over time due to environmental dynamics (Baek et al., 2021). As a result, model performance declines when applied in the long term or under conditions different from the training data. Thus, integrating adaptive mechanisms is essential to maintain the stability of the IoT fire detection system performance (Liu et al., 2021).

Hybrid Fuzzy–Machine Learning Approach

Fuzzy Logic enables handling uncertainty and ambiguity in data through membership functions and flexible inference rules. Several studies have combined fuzzy logic with machine learning to improve the system's ability to cope with dynamic environments. For example, hybrid systems can adjust detection thresholds based on the degree of membership of variables such as temperature, smoke, or gas. However, a common weakness of this approach is the lack of a real-time threshold update mechanism (online adaptive threshold). The output scores from fuzzy or ML systems are usually static, so control over the false alarm rate (FAR) is not yet optimal. This limitation indicates the need for a hybrid model that not only combines fuzzy and ML but also includes an adaptive mechanism to the environmental baseline (Abdullahi et al., 2025).

Adaptive Thresholds and Change/Anomaly Detection in Time Series

Statistical methods such as $\mu \pm k\sigma$, Exponentially Weighted Moving Average (EWMA), Cumulative Sum (CUSUM), and Page–Hinkley have been widely used for real-time anomaly detection. These techniques are effective at detecting changes in data patterns, but controlling false alarm rates can be challenging when the smoothing process is too aggressive. In addition, some methods are black box in nature, making it challenging to interpret detection results and calibration processes. In the context of IoT, this approach needs to be adjusted to be able to work adaptively while maintaining transparency and computational efficiency (Desikan et al., 2025).

Synthesis and Position of Contribution

From the literature review, a consistent research gap was found, namely the absence of an end-to-end dynamic threshold framework that: (a) utilizes Fuzzy Logic to handle uncertainty, (b) applies Random Forest-based machine learning for sensitivity (k) tuning, (c) uses Adaptive Z -Score based on a sliding window to adjust the local baseline, (d) adds a trend signal (slope) to accelerate detection, and (e) provides explicit control over the false alarm rate (FAR) during calibration. The method proposed in this study is designed to address these gaps. Test results on public datasets show that this approach provides significant improvements in F1-score and recall, decreases in FAR, and faster detection times compared to conventional static threshold methods.

Table 1. Summary of Previous Studies Related to IoT Fire Detection

No	Researcher & Affiliation (Year/Venue)	Title/Brief Topic	Operational Definition & Analysis Tools	Main Results (Summary)
1	Chan et al., various universities (2024, IEEE Access – survey)	Ground-sensing IoT survey for early fire detection	Systematic study of temperature/PM/gas sensors, network topology, metrics (Accuracy, FAR, latency)	Emphasizing the need for adaptive thresholds and FAR control in dynamic environments, sensors remain crucial in indoor areas.
2	Baek et al., Korea (2021, Fire Safety Journal)	Real-time fire detection based on multichannel sensors (DTW)	Multi-sensor time series; Dynamic Time Warping, simple fusion	Detection is improved over a single sensor, but it does not yet include adaptive thresholding or FAR control; this is the basis of our method.
3	Waworundeng, Indonesia (2020, *CogITo Smart J.*)	IoT prototype for smoke/fire detection (static threshold)	Temperature/smoke/CO sensor node; fixed threshold OR rule	Inexpensive and quick to implement, yet produces high FAR in real-world conditions, it motivates the development of dynamic thresholds.
4	Li & Sun (2023, *Mathematics*)	Image-based detection: segmentation + ResNet TL	Vision (camera); dynamic threshold segmentation grayscale + CNN	High accuracy when the line of sight is good, but limited in low-visibility or enclosed spaces; higher computational costs.
5	Bhattacharjee et al. (2024, *arXiv – CNN Forest Monitoring*)	CNN-based *real-time forest monitoring*	CNN vision for smoke/fire detection in forests	High accuracy but highly dependent on visual conditions; complements IoT sensor systems.
6	Bifet & Gavalda (2007, *KDD – ADWIN*)	Drift detection for data streams	ADWIN: adaptive window based on statistical testing	Maintaining adaptation to concept drift is relevant for adjusting the baseline, although not specific to fires.
7	Montgomery (SPC/Quality Control) & Page (*Biometrika – CUSUM*)	Process control: μ ± kσ, EWMA, CUSUM	*Statistical threshold:* $Threshold = \mu + k\sigma$, EWMA/CUSUM	A solid basis for adaptive thresholds; requires k tuning and latency compromise — addressed through RF→k and FAR calibration.
8	Desikan et al. (2025, *Sensors*)	ML-based hybrid sensor fusion for IoT fire risk mitigation	*Sensor fusion* + ML (*fault-tolerant*)	False alarms are reduced through feature fusion, but adaptive thresholds are not yet tied to FAR control → a gap that this research fills.

METHOD

This chapter describes the systematic research stages in developing an Internet of Things (IoT)-based early fire detection system using a computational approach. This research does not involve laboratory experiments or direct physical sensor testing but instead uses publicly available secondary datasets for digital testing using predetermined algorithms. The entire implementation was carried out in a Jupyter Notebook running on a local computer, with dataset calls from the following directories: Indoor Laboratory Fire Dataset and Smoke Detection

Dataset. This approach allows testing, training, and evaluation to be carried out in a controlled, replicable manner. The research method consists of four main stages: system architecture design, data preprocessing, feature extraction, and the application of two comparative approaches: a static threshold baseline and a dynamic threshold based on the integration of Fuzzy Logic, Random Forest, and Adaptive Z-Score.

System Architecture Design

The fire detection system in this study is designed as a fully computational IoT-based framework utilizing secondary sensor data, without direct hardware deployment. The architecture is hybrid, integrating three core components : Fuzzy Logic, Random Forest, and Adaptive Z-Score, to generate the final Fire or No Fire decision. The workflow includes raw data preprocessing, feature extraction, fuzzy-based hazard scoring, adaptive parameter tuning via Random Forest, and dynamic threshold calculation using the Adaptive Z-Score method. A Decision Engine then verifies the output through a k-of-n voting mechanism to enhance stability and reduce false alarms. The system architecture design is depicted in Figure 1.

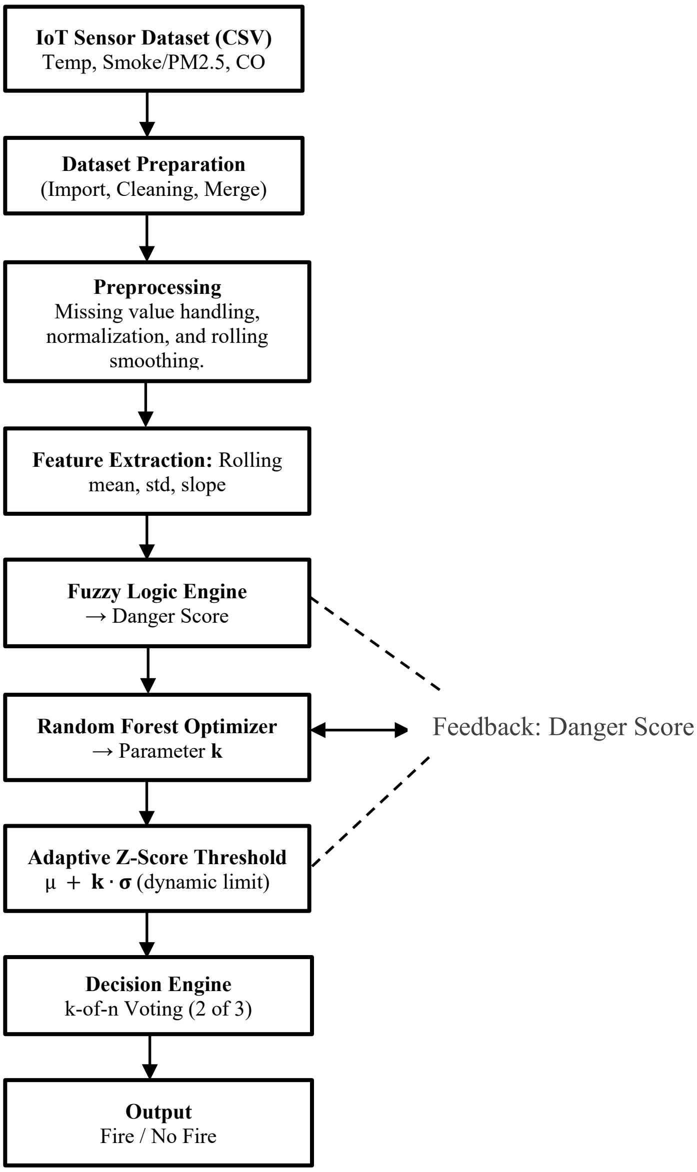

Fig 1. Block Diagram of the Dynamic-Threshold IoT-Based Fire Detection System

Data Collection and Preprocessing

This study uses a combination of laboratory data and realistic data from Kaggle. The research dataset consists of two sources: (1) Indoor Laboratory Fire Dataset, which contains sensor data from laboratory experiments that record changes in temperature, gas, and smoke under various fire conditions, (2) Smoke Detection Dataset, which contains sensor data under realistic conditions for detecting smoke and fire in indoor spaces. Both datasets were downloaded in CSV format and processed using Python in Jupyter Notebook. Data preprocessing steps included: (1) Handling missing values using linear interpolation and forward/backward fill methods to maintain data continuity. (2) Normalizing data using the Min–Max normalization method to a range of [0,1] to ensure uniform reading scales between sensors. (3) Noise reduction using rolling mean or median filters. (4) Standard deviation stabilization to prevent zero division in adaptive z-score calculations. The result of this process is clean processed data, as illustrated in Figure 2.

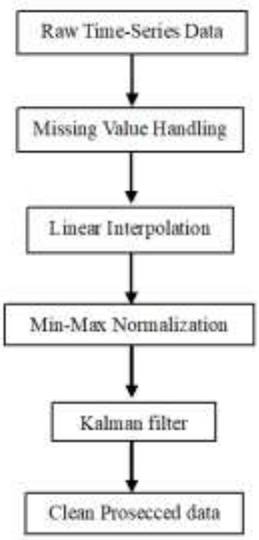

Fig 2. Visualization of the MQ-139 sensor data preprocessing with rolling mean

Feature Extraction

Each data point represents periodic readings from three main sensors (temperature, PM2.5, and CO), along with labels indicating Fire and No-Fire conditions. The extracted features include Mean and standard deviation values (rolling mean and rolling standard deviation), Fuzzy score (danger level), Adaptive Z-score (based on local baseline), Slope score (signal increase level), and Z-max value (absolute maximum Z-score across sensors). All these features are used as input into a Random Forest model to determine the optimal value in the dynamic threshold formula. The results are used in the real-time detection phase to determine potential fire conditions (Aparcana-Tasayco et al., 2025).

Implementation of Dynamic Threshold

The dynamic threshold approach is designed to overcome the rigidity of conventional methods that use fixed thresholds (Ehsan et al., 2022). This system integrates three main modules: (1) Fuzzy Logic, which translates sensor values into hazard scores based on membership functions and inference rules. (2) Random Forest, which receives fuzzy scores and historical features as input to estimate sensitivity parameters k. Adaptive Z-Score, which adjusts thresholds contextually based on average values (μ) and the current standard deviation (σ). $Threshold = \mu_t + k_t \cdot \sigma_{t.}$ The k_t parameter is generated by RF and updated adaptively. The final decision is made by the Decision Engine module, which checks whether the sensor value exceeds the dynamic threshold and confirms the result through a voting mechanism. This system can adjust thresholds to environmental changes, such as daily temperature fluctuations or human activity, making it more resilient and reducing detection errors (Toledo-Castro et al., 2021).

Implementation of Static Threshold (Baseline)

The static threshold model serves as a benchmark (baseline). The system will only activate an alarm if one of the sensor parameters exceeds a fixed threshold value, for example: $(Temperature\ T > Tthr, PM2.5\ S > Sthr, or\ CO\ C > Cthr)$. The threshold value is determined based on the percentile distribution in the validation data. Although computationally efficient (O(1) complexity per sample), this method is unable to adapt to environmental dynamics, thus tending to increase false alarms.
Performance Evaluation
System performance is measured using five key metrics:
Accuracy

*name of corresponding author

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Measures the proportion of correct predictions from all data.
False Alarm Rate (FAR)

$$\mathrm{FAR} = \frac{FP}{FP + TN} \tag{2}$$

Shows how often the system gives false alarms.

F1-Score

$$F1 = 2\,x\,\frac{Precision\ x\ Recall}{Precision + Recall} \tag{3}$$

With

$$Precision = \frac{TP}{TP + FP}\ \text{and}\ \mathrm{Recall} = \frac{TP}{TP + FN} \tag{4}$$

The time difference between the start of the fire and when the system sounds the alarm. Computational complexity comparing the efficiency of the static method (O(1) per sample) with the dynamic method (O(n) per window), both of which remain space-efficient because they are implemented in streaming.

Validation Protocol

The validation protocol is designed to ensure that the system testing process is objective, measurable, and replicable. The dataset used is a combination of two data sources: the Indoor Laboratory Fire Dataset (70%) and the Smoke Detection Dataset (30%). These sources represent laboratory fire conditions and realistic indoor situations. Data division was carried out using the stratified sampling method to ensure balance between the Fire and No-Fire classes.

The training stage used a 5-fold cross-validation scheme to prevent overfitting and ensure model generalization. Next, the testing process was carried out with 70% of the data allocated for training and 30% for testing. All stages of the experiment were run using Python in the Jupyter Notebook environment. Performance evaluation was conducted by comparing two main approaches: the static threshold method and the dynamic threshold, which integrates Fuzzy Logic, Random Forests, and Adaptive Z-Score (Alatawi, 2025). System performance was measured using key metrics: Accuracy, Precision, Recall, F1-Score, False Alarm Rate (FAR), and Detection Time.

To test the robustness and reliability of the model, the system was evaluated in three realistic test scenarios, namely (1) gradual fire, a condition in which the increase in temperature and smoke occurs slowly, simulating a fire that develops gradually, (2) fast ignition, describing a situation of rapid fire due to an electrical short circuit or small explosion, and (3) non-fire interference conditions, simulating non-fire activities such as cooking or dust exposure, to test the system's ability to avoid false alarms.

Statistical analysis of the test results was performed using a paired t-test at a significance level of p=0.05, supplemented with a 95% confidence interval obtained via bootstrap resampling with 1,000 iterations. This protocol describes the complete flow from data preparation, model training, and validation to the analysis of results, allowing other researchers to replicate the entire procedure and reflect realistic system operating conditions.

RESULT

This chapter presents the results of experiments and performance analysis of an IoT-based early fire detection system using two approaches: static and dynamic thresholds. The study was conducted based on six main metrics: Accuracy, False Alarm Rate (FAR), Detection Time, F1-Score, Recall, and Precision. The discussion focuses on the effectiveness of dynamic threshold adaptivity to variations in real environmental conditions.

Dataset Description

This research dataset was obtained from an open data source (Kaggle) containing IoT sensor data for detecting fire and smoke events. This dataset comprises two main groups: the Indoor Laboratory Fire Dataset and the Smoke Detection Dataset. The two datasets were combined to increase data diversity and support model generalization.

The fire dataset comes from digitally documented laboratory experiments, while the smoke dataset contains simulations and indoor sensor measurements. All data were processed in Python using a Jupyter Notebook environment, with preprocessing steps including normalization, imputation, and rolling smoothing to reduce noise. Next, the data were split into training, validation, and test sets using stratified sampling to maintain a balanced proportion of Fire/No-Fire classes.

Table 2. Summarizes The Number Of Samples, Features, and Description Of Each Dataset Subset

Dataset	Sample	Features	Description
Cardboard 1 & 2	786	9	Cardboard sensor data
Clothing 1 & 2	3.217	9	Data sensor clothing
Electricity1–4	7.754	9	Electricity sensor data
Fire Data	10.000	5	Laboratory fire dataset
Processed Fire Data	84.997	5	Fire data after preprocessing
Synthetic Indoor IoT Data	20.000	5	Indoor IoT simulation data
Combined Fire Data Set	126.794	24	Combined fire dataset
Smoke Dataset	62.630	16	Smoke dataset
Combined Data Set	189.424	39	Fire + Smoke

Sensor Data Visualization and Preprocessing Results

This section displays several visualizations that illustrate the effects of preprocessing and compare performance between static and dynamic thresholds.

Bar Chart Comparison of Static vs. Dynamic

To clearly demonstrate the performance improvement achieved by the dynamic threshold approach, a comparative bar chart is presented. This visualization highlights differences across six key evaluation metrics used in the study.

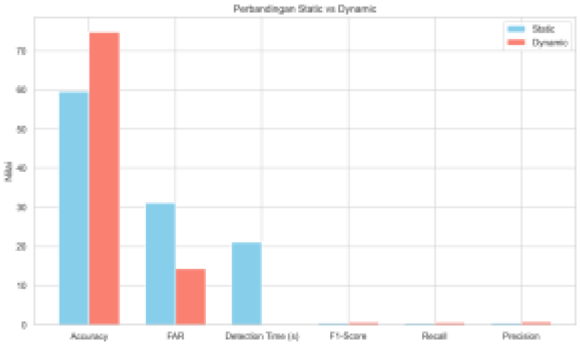

Fig 3. Comparison of static and dynamic threshold performance across six evaluation metrics.

The results in Figure 3 show that the dynamic threshold method consistently performs better than the static threshold approach. Accuracy improves from 59.5% to 74.8%, while the False Alarm Rate (FAR) drops significantly from 31.1% to 14.3%. In addition, the detection time is reduced from 21 seconds to near-zero latency, indicating faster system response to fire events. Improvements in F1-Score, precision, and recall further confirm that dynamic thresholding provides more reliable and adaptive fire detection in varying environmental conditions.

Boxplot of Prediction Probability Distribution

To further analyze the confidence level of each detection approach, a boxplot is used to illustrate the distribution of prediction probabilities generated by the static and dynamic threshold methods.

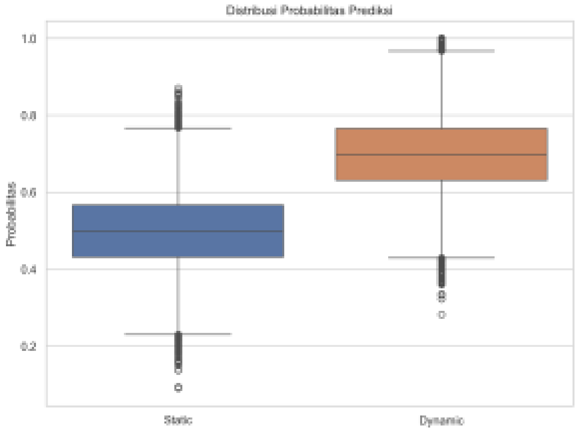

Fig 4. Boxplot of Accuracy Results

As shown in Figure 4, the dynamic threshold method produces a more concentrated and stable probability distribution compared to the static method. The median probability value for the dynamic approach is approximately 0.748, which is higher than the static method's median of around 0.595. Additionally, the dynamic method exhibits fewer outliers, indicating a more consistent prediction pattern. This suggests that the dynamic threshold model operates with higher confidence when identifying potential fire events, thereby improving overall reliability and detection accuracy.

Prediction Probability Histogram

To evaluate classification performance in detail, confusion matrices are used to compare the static and dynamic threshold approaches. This visualization highlights the distribution of True Positive, True Negative, False Positive, and False Negative outcomes for each method.

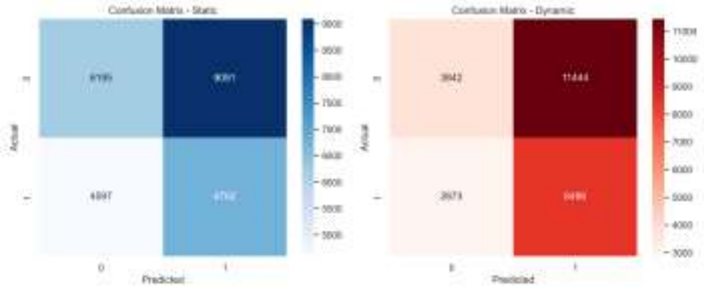

Fig 5. False Alarm Rate Distribution Histogram – FAR

As shown in Figure 5, the dynamic threshold method achieves higher True Positive and True Negative values compared to the static approach. Specifically, the dynamic model correctly identifies a larger number of fire events while also reducing false alarms. The decrease in False Positive and False Negative instances indicates a more balanced and accurate detection response. Overall, these results demonstrate that the dynamic threshold system is more effective and reliable in distinguishing fire conditions from normal environmental variations.

Prediction Probability Histogram

To further evaluate model confidence and prediction behavior, a probability distribution histogram is used to compare the static and dynamic threshold methods.

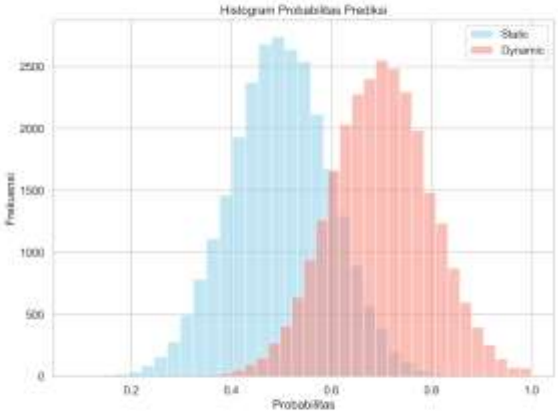

Fig 6. Histogram of Probability Distribution Predictions in Static and Dynamic Methods

As illustrated in Figure 6, the dynamic threshold model produces a tighter probability distribution centered at higher values, indicating stronger confidence in fire event classification. In contrast, the static threshold method shows a wider spread with lower peak probabilities, reflecting greater uncertainty in its predictions. This distribution pattern confirms that the dynamic approach not only enhances detection accuracy but also provides more consistent and reliable decision-making in varying environmental conditions.

Line Plot of Prediction Probabilities for the First 100 Samples

To observe model behavior over time, a line plot is used to display the prediction probabilities for the first 100 data samples under both static and dynamic threshold approaches.

*name of corresponding author

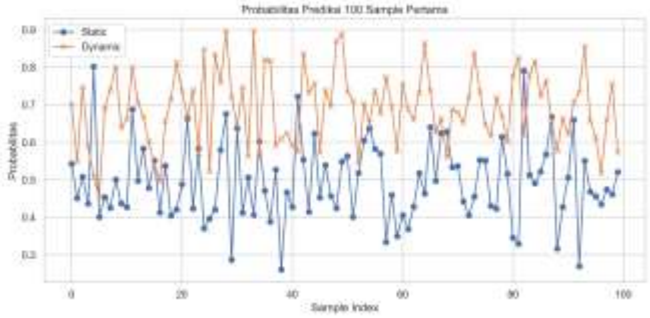

Fig 7. Line Plot of Prediction Probabilities for the First 100 Samples

As shown in Figure 7, the static threshold method demonstrates high variability and inconsistent probability values, indicating unstable confidence levels and a higher likelihood of misclassification. In contrast, the dynamic threshold method maintains a more stable and elevated probability pattern across the same sample range. This stability reflects the model's ability to adapt to sensor data fluctuations in real time, resulting in more reliable fire detection and fewer erratic predictions.

Performance Evaluation

The following table compares performance evaluation results for static and dynamic threshold methods. Based on the results in Table 3, the dynamic threshold method consistently outperforms the static threshold method. An increase in accuracy of 26%, a reduction in the False Alarm Rate (FAR) by more than 50%, and a reduction in detection time from 21 seconds to 0 seconds demonstrate a significant improvement in the system's speed and accuracy. In addition, the increase in F1-score, precision, and recall values confirms that the dynamic threshold approach has better adaptive capabilities to changes in environmental conditions, making the system more efficient, reliable, and responsive in detecting potential fires.

Table 3. Static and Dynamic Threshold Evaluation Results

Metrics	Static	Dynamic	Improvement (the lower the FAR value, the better)
Accuracy	59.5%	74.8%	+26%
FAR	31.1%	14.3%	+54% (the smaller the value, the better)
Detection Time(s)	21.0	0.0	+-100%
F1	0.459	0.638	+39%
Recall	0.446	0.575	+29%
Precision	0.473	0.716	+51%

DISCUSSIONS

The results of the study show that dynamic thresholds provide a significant performance improvement over static threshold methods in IoT-based early fire detection systems. The developed system can adaptively adjust sensor thresholds in response to changes in environmental conditions, making it more responsive to early indications of fire without generating excessive false alarms.

The effectiveness of this approach is achieved through the integration of three main techniques, namely Fuzzy Logic, Random Forest, and Adaptive Z-Score. Fuzzy Logic helps manage sensor value uncertainty by converting it into flexible fire risk levels. Random Forests optimize classification and identify historical patterns in environmental data, while Adaptive Z-Score adjusts the detection threshold in response to real-time fluctuations in sensor values. The synergy of these three techniques enables the system to automatically adapt to changes in temperature, smoke concentration, and carbon monoxide levels, with a more balanced sensitivity between early detection and reduced false alarms.

Based on testing on the combined dataset, the system with an adaptive or dynamic threshold achieved 74.8 percent detection accuracy, a 14.3 percent reduction in False Alarm Rate (FAR), and a detection time of 0 seconds. These results confirm that the adaptive mechanism can provide an optimal balance between system sensitivity and stability. This model also has real application potential in industrial environments, densely populated residential areas, and public facilities that require fast, accurate, and reliable fire detection.

However, the dynamic threshold approach still has several limitations. One of them is its dependence on sensor quality and calibration, as extreme reading variations can affect the stability of the adaptive model. In addition, external factors such as humidity, air circulation, and sensor position can cause variations in the data, affecting prediction accuracy. Therefore, additional field tests and adaptive parameter adjustments are needed to ensure the system operates consistently across various real-world conditions.

CONCLUSION

This study concludes that applying a dynamic threshold that integrates Fuzzy Logic, Random Forests, and Adaptive Z-Score significantly improves the effectiveness of an IoT-based early fire detection system. The developed system automatically adjusts sensor thresholds in response to changes in environmental conditions, thereby improving detection accuracy, reducing false alarm rates, and speeding up response times. These results show that the adaptive approach is more efficient than static methods, which tend to be less flexible in the face of field sensor data variations.

The main advantage of this research lies in its ability to deliver an intelligent, flexible, and efficient detection system. Dynamic threshold models can be applied in high-risk areas, such as warehouses, factories, and densely populated urban areas. In addition, the adaptive concept can be further developed for various other IoT-based monitoring applications, such as air quality monitoring systems, gas leak detection, and real-time environmental monitoring.

Although promising results have been achieved, this study still has limitations, particularly in the number of sensors used and the scope of testing, which do not fully represent field conditions. Therefore, further research is recommended to add more sensor types and numbers to expand detection coverage and improve model accuracy. In addition, large-scale field testing with system integration into real-time monitoring dashboards and automatic mitigation mechanisms such as smart alarms or smart sprinklers will further improve the efficiency and reliability of the system in dealing with emergency situations.

Overall, this research makes a real contribution to the development of IoT-based security systems with an adaptive approach that can adapt to environmental dynamics. These findings demonstrate the great potential of using dynamic thresholds as the basis for developing smarter, faster, and more accurate early fire detection systems in the future.

REFERENCES

Abdullahi, U. I., Zhang, W., Cao, Y., & Irankunda, G. (2025). Integrating IoT Technology for Fire Risk Monitoring and Assessment in Residential Building Design. *Buildings*, *15*(8). https://doi.org/10.3390/buildings15081346

Alatawi, M. N. (2025). Optimizing security and energy efficiency in IoT-Based health monitoring systems for wireless body area networks. *Scientific Reports*, *15*(1), 1–20. https://doi.org/10.1038/s41598-025-11253-x

Aparcana-Tasayco, A. J., Deng, X., & Park, J. H. (2025). A systematic review of anomaly detection in IoT security: towards quantum machine learning approach. *EPJ Quantum Technology*, *12*(1). https://doi.org/10.1140/epjqt/s40507-025-00414-6

Baek, J., Alhindi, T. J., Jeong, Y. S., Jeong, M. K., Seo, S., Kang, J., Shim, W., & Heo, Y. (2021). Real-time fire detection system based on dynamic time warping of multichannel sensor networks. *Fire Safety Journal*, *123*(April). https://doi.org/10.1016/j.firesaf.2021.103364

Bonilla-Ormachea, K., Cuizaga, H., Salcedo, E., Castro, S., Fernandez-Testa, S., & Mamani, M. (2025). *ForestProtector: An IoT Architecture Integrating Machine Vision and Deep Reinforcement Learning for Efficient Wildfire Monitoring*. 70–75. https://doi.org/10.1109/icara64554.2025.10977677

Chan, C. C., Alvi, S. A., Zhou, X., Durrani, S., Wilson, N., & Yebra, M. (2024). A Survey on IoT Ground Sensing Systems for Early Wildfire Detection: Technologies, Challenges, and Opportunities. *IEEE Access*, *12*(October), 172785–172819. https://doi.org/10.1109/ACCESS.2024.3501336

Desikan, J., Singh, S. K., Jayanthiladevi, A., Bhushan, S., Rishiwal, V., & Kumar, M. (2025). Hybrid Machine Learning-Based Fault-Tolerant Sensor Data Fusion and Anomaly Detection for Fire Risk Mitigation in IIoT Environment. *Sensors*, *25*(7). https://doi.org/10.3390/s25072146

Ding, Y., Wang, M., Fu, Y., Zhang, L., & Wang, X. (2023). A Wildfire Detection Algorithm Based on the Dynamic Brightness Temperature Threshold. *Forests*, *14*(3). https://doi.org/10.3390/f14030477

Ehsan, I., Mumtaz, A., Khalid, M. I., Iqbal, J., Hussain, S., Ullah, S. S., & Umar, F. (2022). Internet of Things-Based Fire Alarm Navigation System: A Fire-Rescue Department Perspective. *Mobile Information Systems*, *2022*. https://doi.org/10.1155/2022/3830372

Khan, T., Singh, K., Bhati, B. S., Ahmad, K., Al-Rasheed, A., Getahun, M., & Soufiene, B. O. (2025). Trust-driven approach to enhance early forest fire detection using machine learning. *Scientific Reports*, *15*(1), 1–22. https://doi.org/10.1038/s41598-025-99032-6

Li, H., & Sun, P. (2023). Image-Based Fire Detection Using Dynamic Threshold Grayscale Segmentation and Residual Network Transfer Learning. *Mathematics*, *11*(18). https://doi.org/10.3390/math11183940

Liu, H. H., Chang, R. Y., Chen, Y. Y., & Fu, I. K. (2021). Sensor-Based Satellite IoT for Early Wildfire Detection. *2021 IEEE Globecom Workshops, GC Wkshps 2021 - Proceedings*. https://doi.org/10.1109/GCWkshps52748.2021.9682098

Ma, M., Xu, C., & Han, J. (2025). Application of an intelligent electrical fire monitoring system based on the EC-IOT framework in high-rise residential buildings. *Systems and Soft Computing*, *7*(April), 200257.

https://doi.org/10.1016/j.sasc.2025.200257

Morchid, A., Oughannou, Z., Alami, R. El, Qjidaa, H., Jamil, M. O., & Khalid, H. M. (2024). Integrated internet of things (IoT) solutions for early fire detection in smart agriculture. *Results in Engineering*, *24*(September), 103392. https://doi.org/10.1016/j.rineng.2024.103392

Park, S. H., Kim, D. H., & Kim, S. C. (2023). Recognition of IoT-based fire-detection system fire-signal patterns applying fuzzy logic. *Heliyon*, *9*(2), e12964. https://doi.org/10.1016/j.heliyon.2023.e12964

Sharma, A., Nayyar, A., Singh, K. J., Kapoor, D. S., Thakur, K., & Mahajan, S. (2024). An IoT-based forest fire detection system: design and testing. *Multimedia Tools and Applications*, *83*(13), 38685–38710. https://doi.org/10.1007/s11042-023-17027-9

Toledo-Castro, J., Rodríguez-Perez, N., Caballero-Gil, P., Santos-González, I., Hernández-Goya, C., & Aguasca-Colomo, R. (2021). Detection of forest fires outbreaks by dynamic fuzzy logic controller. *Logic Journal of the IGPL*, *29*(6), 936–950. https://doi.org/10.1093/jigpal/jzaa036

Vaegae, N. K., Annepu, V., Bagadi, K., & Dahan, F. (2024). Multisensor Fuzzy Logic Approach for Enhanced Fire Detection in Smart Cities. *Journal of Optimization*, *2024*(1). https://doi.org/10.1155/2024/8511649