Edu Komputika 12 (2) (2025)

**Edu Komputika Journal**

https://journal.unnes.ac.id/journals/edukom/

# Multisensor Integration in Early Fire and Gas Detection Based on Artificial Neural Network

**Khoirudin Fathoni[1)]✉, Alfa Faridh Suni[2)], Nur Iksan[3)], and Mohammad Sofyan Aziz[3)]**

[1]Electrical Engineering Education Study Program, Faculty of Engineering, Universitas Negeri Semarang, Sekaran, Gunungpati, Semarang, 50229, Indonesia

[2]Computer Engineering Study Program, Faculty of Engineering, Universitas Negeri Semarang, Sekaran, Gunungpati, Semarang, 50229, Indonesia

[3]Electrical Engineering Study Program, Faculty of Engineering, Universitas Negeri Semarang, Sekaran, Gunungpati, Semarang, 50229, Indonesia

## Article Info

**Article History:**
Received: 7 November 2025
Revised: 20 December 2025
Accepted: 9 April 2026

**Keywords:**
Artificial Neural Network, False Alarm, Fire Detection, Hazardous Gases.

## Abstract

Fire detection systems commonly rely on a single variable, such as smoke or flame, to detect fires. However, using a single sensor or threshold-based method for early detection is prone to false alarms. To address this issue, this study proposes an early fire detection system using an artificial neural network based on the Radial Basis Function Network (RBFN) architecture. The aim of this research is to minimize false alarms by implementing an early fire detection system that not only detects flames but also hazardous gases, temperature, and humidity as potential sources of fire. A multisensor system comprising an IR flame sensor, gas sensors MQ-9, MQ-2, and MQ-4, as well as a DHT-11 temperature and humidity sensor, was integrated and processed using an RBFN-based ANN on a Raspberry Pi 3. The ANN processes a series of datasets trained to generate a model that determines fire conditions. Testing results showed that the proposed method did not produce any false alarms, with a response time of 2.1 seconds from the ignition of a fire source to the issuance of a warning.



✉ Correspondence Address:
E8 Building, UNNES Kampus Sekaran, Gunung Pati, Semarang City, Central Java 50229
E-mail: khoirudinfathoni@mail.unnes.ac.id

ISSN 2460-0741
E-ISSN 2548-9364

## INTRODUCTION

Fire is an essential element for human life. However, it can become dangerous when it causes fires, which may result in not only material losses but also fatalities. Fires can be triggered by various factors, such as electrical short circuits, gas leaks, and sparks (Noorfirdaus et al., n.d.). Numerous fire detection systems have been developed using different detection methods. Modern fire detection systems employ various techniques aimed at minimizing the main issue in fire detection, i.e., false alarms (Muharam et al., 2020). A false alarm refers to a condition in which the system either fails to alert during a fire or mistakenly triggers an alert when there is no fire.

Several studies have been conducted on the development of hardware and control systems for fire and hazardous gas detection. Among them, they utilized various types of sensors to detect fire directly without applying any input data processing methods. Study (Kusuma Dewi et al., 2022) used an infrared (IR) sensor and an Arduino microcontroller. This study tested the reliability of the IR sensor in detecting fire. The outcome showed that the IR sensor was not optimal in detecting fire, resulting in errors in the output. Next, the study (Iskandar Alam et al., 2019) used an IR sensor and an MQ-2 sensor. The IR sensor was used to detect fire, while the MQ-2 sensor was used to detect smoke. The test results were then transmitted to the controller, which activated a water sprinkler if the output indicated a fire. Study (Ferdyansyah et al., 2022) used a DHT22 sensor to detect temperature and an MQ-2 sensor to detect smoke. This study evaluated both sensors by triggering them to detect fire. When the sensors were triggered, the controller sent a signal to activate a water pump to spray water onto the fire source. Study (Suwarjono et al., 2021) used a threshold algorithm with temperature and gas sensors. This research determined sensor thresholds through experimentation. The system's output was then forwarded to an SMS gateway. Study (Getu et al., 2022) used a temperature sensor to obtain readings, implementing a threshold algorithm with a limit of 50 degrees Celsius. If the sensor reading exceeded this threshold, the data was sent to the Arduino, which then activated the pump. Study (Rehman et al., 2021) used multiple parameters to determine the appropriate response, including temperature, humidity, and smoke. If the output indicated a hazardous situation, the system would activate the ventilation and water pump. This study used MATLAB software to simulate the system. In a study (Nopriadi & Fajrin, 2023), a microcontroller was used to build a robot capable of detecting fire. The robot relied solely on an infrared sensor for fire detection. It also moved autonomously using ultrasonic sensors to identify nearby objects, allowing it to navigate freely. This study did not use any data classification method; it only relied on real-time data received by the IR sensor. Study (Kharisma & Setiyansah, 2021) employed temperature and gas parameters, which were sent to an Arduino microcontroller to generate an SMS notification to the user. The focus of this study was on how quickly the user received the SMS when a fire occurred. This study used only a threshold-based method. Although threshold-based detection using sensors without any classification methods is effective, it has limitations. The primary weakness is that if only one sensor is triggered, the system activates, potentially causing a false alarm. Therefore, this research integrates five sensors to address this issue.

Several studies on fire detection have also employed the fuzzy logic method. Among them, some studies investigated fire detection using fuzzy logic classification methods. A study (Irfanianingrum et al., 2023) examined the use of fuzzy logic for early fire detection. In this study, the fuzzification values were categorized into several classes. The result of the study showed that the fuzzy method was successfully used for early fire detection based on the fuzzified outputs from the flame sensor, DHT11 sensor, and MQ-2 sensor. Next, the study (HW et al., 2021) focused on the application of fuzzy logic to detect fire using a real-time database. This study used MQ-2, DHT22, and IR sensors. The fuzzy membership values were determined through a trial-and-error process. The results indicated that the fuzzy method produced no errors. Study (Rachman et al., 2020) applied fuzzy logic to detect fire using smoke, flame, and temperature sensors. This research used various labels to define the fuzzy inputs. It compared the fuzzification results obtained through practical implementation and MATLAB simulation. The result showed a 0.99% error rate. A study (Ren et al., 2021) implemented fuzzy logic to detect fires in greenhouses. A greenhouse is a building that uses very minimal energy. Simple fire detectors often encounter problems when installed in greenhouses due to the very low voltage received by the system. Therefore, this study proposed the development of a fire detection system using fuzzy logic to address this issue. However, the system was only tested under non-extreme conditions, which means that prediction errors under extreme conditions are still highly likely.

Based on studies of fire detection systems using fuzzy logic, it can be concluded that while the method performs well, it still has several limitations. One of the main weaknesses is that the accuracy of fire detection can be influenced by the selection of input variables and the rules applied, which may lead to false alarms or failure to detect actual fires. Additionally, in cases where there are significant environmental or condition changes, fuzzy systems may not effectively adapt or produce accurate outputs due to the lack of training to accurately recognize such conditions. Therefore, this study employs the Artificial Neural Network (ANN) method, which can overcome these issues.

Subsequent research explored fire detection using the YOLO (You Only Look Once) method. YOLO is a machine learning technique that processes images into matrix data, which is then analyzed to identify objects. Studies (Salman Farhan et al., 2022). The study conducted tests during both daytime and nighttime. During the day, the system achieved only 80% accuracy, while at night, the accuracy increased to 97%. This difference was due to backlighting during the day, which prevented the camera from properly detecting fire objects. Additionally, the distance between the camera and the fire also affected the system's accuracy and precision. Therefore, camera-based systems are considered less reliable and only accurate under certain conditions. Study (Talaat & ZainEldin, 2023) used YOLOv4 for fire detection with three different models, each having different layers and datasets. Among the three models, the first model achieved the highest accuracy at 90%. Study (Abdusalomov et al., 2021) employed YOLOv3 for fire detection using surveillance cameras during both day and night. This study identified several issues with the YOLO method, one of which was misprediction when the background color lacked contrast with the color of the fire. A study (Bahhar et al., 2023) investigated the use of the YOLO method for detecting forest fires. Like the other YOLO-based studies, it encountered several problems. One key issue was false fire detection due to the nature of fire, which lacks a consistent shape. If the fire resembled another object, the system produced a misprediction. From these studies, it can be concluded that the YOLO method has several limitations: inconsistent accuracy, high dependence on camera placement, long training time, and difficulty detecting small or clustered objects.

Research on fire detection using neural networks has been conducted in several studies. Study (Cheng et al., 2011) used only three input variables: temperature, smoke, and CO gas, with an error rate of 2.3%. It was concluded that adding IR and methane sensors would improve fire detection performance. Study (Nakip et al., 2021) employed a Recurrent Neural Network (RNN) architecture and achieved an accuracy of 96%. However, due to the complexity of RNN, the Radial Basis Function (RBF) network architecture was preferred in this study.

Based on the aforementioned studies, this research aims to improve the performance of fire and hazardous gas detection systems by reducing false alarms and improving accuracy. This objective is achieved by using five sensors—DHT11, IR, MQ-2, MQ-4, and MQ-9—integrated with a Raspberry Pi 3B microcontroller and employing an ANN with an RBF network architecture for data processing.

## RESEARCH METHODS

This study proposes the development of a fire and hazardous gas detection system using an Artificial Neural Network (ANN) with a Radial Basis Function (RBF) architecture. The system uses a Raspberry Pi 3B as the controller, with a DHT11 sensor for temperature and humidity, an IR sensor for infrared detection, an MQ-2 sensor for $CO_2$ detection, an MQ-4 sensor for methane detection, and an MQ-9 sensor for CO detection. The data and outputs will be displayed on an LCD in real-time.

Although the DHT11 sensor has limited accuracy compared to high-performance industrial-grade temperature sensors, it is not used as the primary fire detection parameter. Instead, temperature and humidity serve as supporting environmental inputs that provide contextual information. The ANN-based multisensor fusion approach can tolerate sensor noise and reduce dependency on a single sensor's accuracy.

### A. Design System

The early fire and hazardous gas detection system is divided into three parts: input, process, and output, as shown in Figure 1.

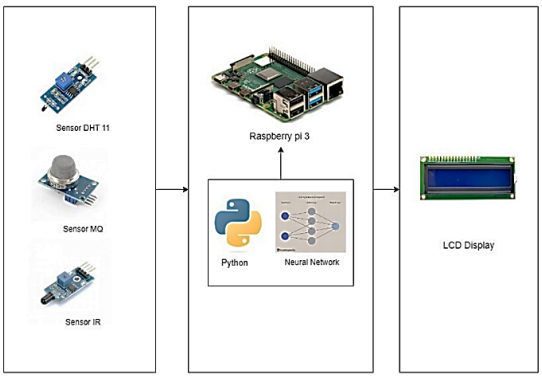


Figure 1. Block diagram

In the processing section, a Raspberry Pi serves as the central processing unit that collects data from five sensors, i.e., three gas sensors, an IR sensor, and a temperature-humidity sensor. The data are processed using an Artificial Neural Network (ANN) implemented in the Python programming language. After processing, the data are sent to the output section, which is an LCD display. Figure 3 shows the prototype design of the fire and hazardous gas detection system.

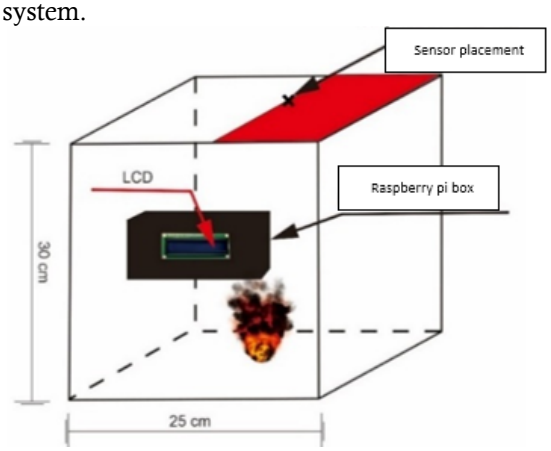


Figure 2. Prototype design system

The prototype design of the fire and hazardous gas detection system uses a transparent acrylic box measuring 25 cm x 30 cm. This size is intended to represent a room measuring 2.5 m x 3 m, making the prototype built to a 1:10 scale. This dimension was chosen because it reflects the typical size of a bedroom in Indonesian homes. However, this acrylic enclosure was designed as a functional representation rather than a physical model of fire dynamics. The purpose of this prototype is to evaluate sensor response patterns and ANN classification performance under controlled and repeatable conditions. This scaling does not aim to replicate real-scale fire growth or heat transfer behavior but focuses on multisensor data fusion and decision-making performance.

In terms of design, on the front of the box, a plant box is attached, which houses the Raspberry Pi 3B and the Analog-to-Digital Converter (ADC). An LCD screen is installed on the front of the plant box to serve as the user interface, displaying real-time temperature readings and gas concentration percentages inside the box. Figure 3 presents the schematic diagram of the fire and hazardous gas detection system.

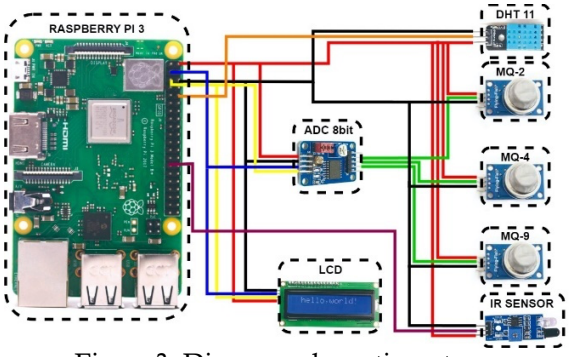


Figure 3. Diagram schematic system

In this fire and hazardous gas detection system, an Analog-to-Digital Converter (ADC) is required. The ADC functions to convert the analog signal output from the MQ sensors into digital signals so that they can be read as numeric values by the Raspberry Pi. The ADC used in this system has an 8-bit specification. Based on the prototype design and the system schematic diagram, the system was then implemented into the device as shown in Figure 4.

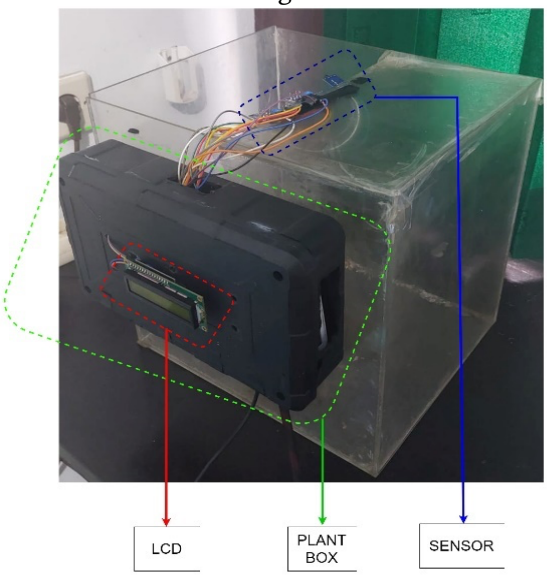


Figure 4. Device implementation

The plant box system contains the Raspberry Pi, 8-bit ADC, and breadboard. The contents inside the plant box are shown in Figure 5.

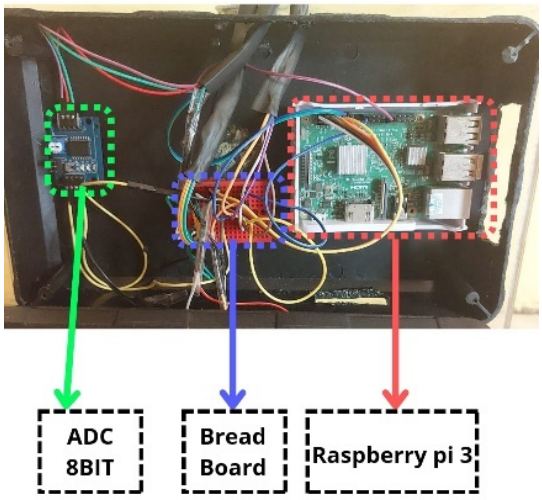


Figure 5. Device implementation

B. Software

The fire detection flowchart is shown in Figure 6. The first step in the early fire and hazardous gas detection system is to detect temperature, flame, and gas using sensors. The sensor used to detect fire is the IR sensor. The sensor used to detect temperature is the DHT11, and the sensors used to detect gases are the MQ series sensors.

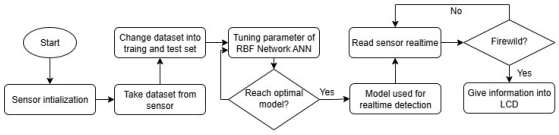


Figure 6. Program flowchart

Once the data is collected, it is processed using the Artificial Neural Network (ANN) algorithm. The obtained data is processed to generate rules. These rules are derived by inputting setpoint data, which the ANN then compares with real-time data. The resulting output is in the form of a label—either "normal" or "fire." This output label is generated by the ANN process.

The dataset labeling process was conducted based on controlled experimental conditions. The label "Normal" represents indoor conditions without any ignition source, where temperature and gas concentrations remain within ambient levels. The label "Fire" represents conditions where a combustion source was intentionally introduced, resulting in simultaneous increases in temperature and combustible gas concentrations. The labeling was performed manually based on experimental observation rather than predefined threshold values, allowing the ANN to learn nonlinear relationships among multiple sensor inputs.

There are several types of ANN methods depending on their applications. In this study, the Radial Basis Function (RBF) network is used. The RBF network method was chosen for its strong capabilities in prediction and data classification. One of the main advantages of the RBF network is that it has only one hidden layer, which makes it faster in predicting outputs. Figure 7 illustrates the basic scheme of the RBF Network.

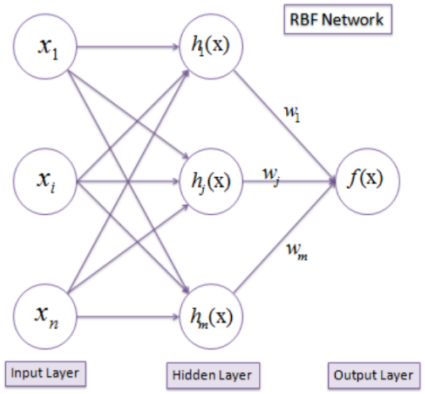


Figure 7. RBF network architecture

$$h(x) = exp\left(\frac{(x-c)^2}{r^2}\right) \quad (1)$$

In the hidden layer (1), where *h(x)* represents the hidden layer, *x* is the input, *c* is the center, and *r* is the data radius. The value of *x* is obtained from sensor readings, while *c* is determined by a clustering algorithm that randomly selects several datasets. The clustering algorithm identifies the center point of these datasets. Meanwhile, *r* is obtained using the K-Nearest Neighbors algorithm, which calculates the distance between each data point and the center (*c*). Then, the output layer is represented in (2).

$$f(x) = \sum_{j=1}^{m} w_j \, h_j(x) x_j \quad (2)$$

Where f(x) is the output layer, w represents the weights, h(x) is the result of the hidden layer computation, m is the number of nodes, and j denotes the index of the equation being calculated.

The RBF network needs to be trained in order to distinguish between fire and normal conditions. It requires a dataset as input, which is

collected from sensors. The dataset was recorded under indoor temperature conditions and consists of 165 data entries. After collecting the dataset, it is divided into two parts: training data and validation data. The training data is used to train the system so it can develop a model and pattern to produce an output—either fire or normal. The validation data is used to validate the results of the RBF network's training. From this training process, a function model is obtained that can predict the system's output. RBF networks require parameter tuning to achieve optimal model performance.

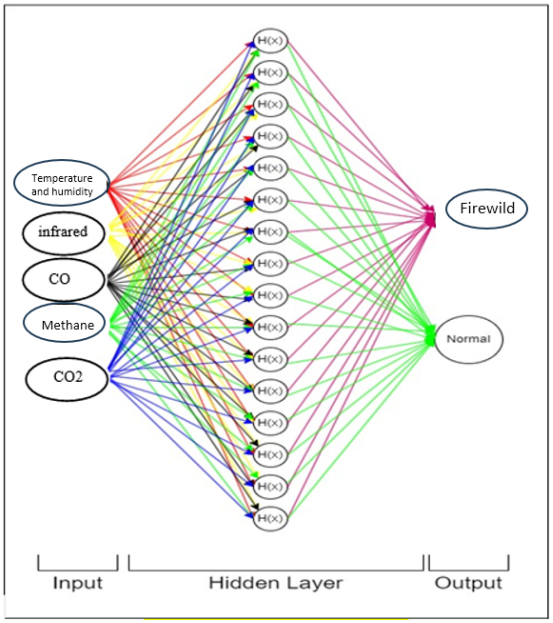


Figure 8. RBF Network model for fire and hazardous gas detection using multisensors

The parameters that are tuned include the number of hidden layer nodes, the number of epochs, and the learning rate. The number of nodes in the hidden layer affects the final prediction results; the more nodes there are, the better the model quality. However, it also increases training time and risks missing the optimal model. The epoch refers to how many times the dataset passes through the algorithm. If the number of epochs is too low, the resulting model may be undertrained; if it is too high, it can lead to longer training times. The learning rate controls the optimization of the model by adjusting the weight changes in the nodes when weight adjustment is necessary due to data loss. The analysis in this study considers several indicators: accuracy, loss, and prediction success. The next step is to split the collected data into two parts: training data and test data. From the validation process, the model's accuracy and data loss can be identified. In this study, the RBF network was tuned with 16 hidden layer nodes, 150 epochs, and a learning rate of 0.001. These parameter values were determined based on trial and error. The final tuned model is illustrated in Figure 8.

The output of the Artificial Neural Network (ANN) is expressed in equation (3), where $f(x)$ represents the output, $w_x$ denotes the weights on the input layer, and $w_h$ refers to the weights on the hidden layer.

$$f(x) = \sum_{j=1}^{m} x_j w_{ij} h_j(x) w_{oj} \quad (3)$$

Here, $w_{ij}$ is the weight associated with input $x_j$, while $w_{oj}$ is the weight associated with the hidden node $h_j(x)$. After calculating the value of $f(x)$, the result is passed through an activation function—specifically, the sigmoid activation function. This function is used to convert the continuous output $f(x)$ into two categorical output codes: 0 indicating no fire and 1 indicating the presence of fire. The equation for the sigmoid activation function is represented in equation (4), where $z$ is the result of $f(x)$.

$$g(z) = \frac{1}{1+e^{-z}} \quad (4)$$

Finally, the model accuracy is determined using the test data. Accuracy is crucial for determining the potential of false alarms in the system. Figure 9 presents the accuracy of the model obtained through testing.

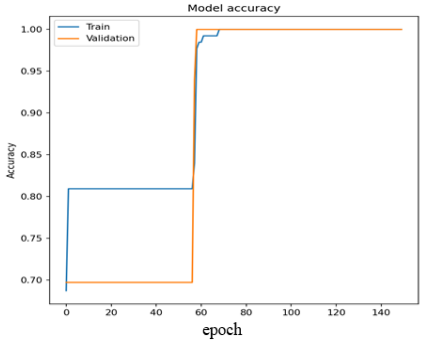


Figure 9. Model accuracy

The graph illustrates the model's accuracy in relation to the number of epochs required by the Artificial Neural Network (ANN). An ideal model is one that achieves an accuracy value of 1 or 100%. As shown in the graph, the model reaches 100% accuracy after approximately 80 epochs, indicating that the selected 150 epochs are sufficient to obtain an accurate model. There are two key parameters in the graph: training data

and validation data. Training data is a subset of the dataset that includes both input and output values and is used to identify patterns that allow the model to generate output based on input.

Validation data, on the other hand, is a separate subset of the dataset used to evaluate the generalizability of the pattern learned from the training data. From the beginning of training until around the 60th epoch, the model accuracy remained at approximately 0.80. The accuracy then increased and reached a value of 1.0 from epoch 61 to epoch 150. This indicates that the RBF Network algorithm requires at least 70 epochs to achieve optimal accuracy. The resulting model's accuracy is considered highly suitable for the fire and hazardous gas detection system developed in this study.

Figure 10 shows the model's data loss graph. Data loss represents the difference between the predicted values and the actual values. Similar to the accuracy graph, the data loss graph compares results obtained from the training data and validation data.

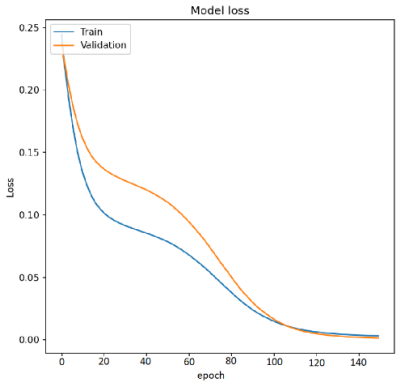


Figure 10. Model loss

A good model is characterized by a data loss value approaching zero, as a smaller data loss indicates that the model's predicted values are closer to the actual values. At epoch 0, the data loss is considerably high, but it decreases significantly by epoch 20. The training loss then continues to decline steadily, approaching zero up to epoch 120. After epoch 120, the data loss stabilizes near zero through epoch 140. This behavior indicates that the system's prediction performance is highly accurate, with minimal deviation between predicted and actual values. Consequently, this model is selected for implementation in the fire and hazardous gas detection system.

In this study, 80% of the dataset was used for training and 20% for testing. The evaluation results show that no misclassification occurred on the test data under the experimental conditions. However, considering the relatively small dataset size, the obtained accuracy reflects experimental performance and may still carry a potential risk of overfitting. The reported 100% accuracy refers strictly to the test dataset under controlled experimental conditions and does not imply generalization to broader or unseen environments. Therefore, further validation using larger datasets and cross-validation techniques is required in future work.

## RESULT AND DISCUSSION

Table 1 presents the threshold values set for each parameter. These threshold values were determined based on testing each sensor under fire conditions. The threshold limits will be tested using several conditional logics: OR, AND, and a combination of AND-OR.

The humidity parameter does not directly affect the fire detection process; however, it plays an important role in minimizing false alarms, as high or low humidity levels can influence the behavior of fire as detected by the system. Therefore, the humidity parameter is still required in this system.

Table 1. Threshold Parameter

| Parameter Name | Parameter Value |
|---|---|
| Temperature | 53 celcius |
| Humidity | 37% |
| CO | 0,62 ppm |
| Metana | 0,026 ppm |
| CO2 | 0,556 ppm |

Based on the threshold values determined through individual sensor testing under fire conditions, the fire and hazardous gas detection system was tested using the OR logic threshold algorithm, starting with normal room temperature conditions. The testing process is shown in Figure 12.

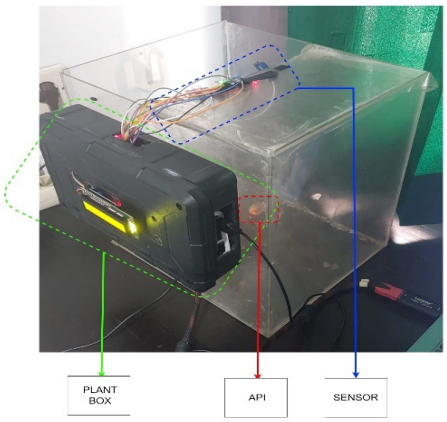


Figure 12. Testing process

The results of the testing process for the fire and hazardous gas detection system are presented in Table 2. Based on Table 2, the system was tested using the threshold-based method. The first test involved the use of OR logic. Initially, the system did not produce any false alarms. However, when the humidity level dropped to 35%, the system triggered a false alarm. This occurred because the predefined threshold for the humidity parameter was set at 37%. As a result, the system classified the condition as fire. However, in the actual condition, no fire had yet occurred. The temperature increased, and the humidity decreased due to the ignition of a heat source before testing, which can raise the temperature even though combustion had not yet taken place. The system's output returned to reflect the actual condition once the test object was fully ignited to simulate a fire situation at a temperature of 53°.

Table 2. Result of OR Logic Test

| Temp ($^0$C) | Humidity (%) | CO (ppm) | methane (ppm) | CO2 (ppm) | System Output | Real Condition |
|---|---|---|---|---|---|---|
| 44 | 50 | 0,0204 | 0,0001 | 0,006 | Normal | Normal |
| 48 | 35 | 0,0204 | 0,0001 | 0,006 | Api | Normal |
| 50 | 29 | 0,0204 | 0,0001 | 0,006 | Normal | Normal |
| 53 | 20 | 0,0204 | 0,0001 | 0,006 | Api | Api |
| 55 | 17 | 0,0204 | 0,0001 | 0,006 | Api | Api |

Next, the fire and hazardous gas detection system was tested using the threshold algorithm with AND logic. The threshold values used were based on the parameters listed in Table 2. The test results using the threshold algorithm with AND logic are presented in Table 3.

Table 3. Result of AND Logic Test

| Temp ($^0$C) | Humidity (%) | CO (ppm) | methane (ppm) | CO2 (ppm) | System Output | Real Condition |
|---|---|---|---|---|---|---|
| 46 | 45 | 0,0204 | 0,0001 | 0,006 | Normal | Normal |
| 46 | 36 | 0,0204 | 0,0001 | 0,006 | Normal | Api |
| 56 | 25 | 0,0204 | 0,0001 | 0,006 | Api | Api |
| 60 | 8 | 0,0204 | 0,0001 | 0,006 | Api | Api |
| 60 | 3 | 1,67 | 0,039 | 0,84 | Api | Api |

Table 3 presents the test results of the fire and hazardous gas detection system using the threshold algorithm with AND logic only. The results show that a false negative occurred when the temperature reached 46°C and the humidity was at 36%. The system produced a "normal" output, whereas in the actual condition, a fire had already been present. This was due to the fact that the AND logic threshold algorithm requires at least 2 parameters to reach their respective thresholds before identifying a fire condition.

Table 4. Result of AND-OR Logic Test

| Temp ($^0$C) | Humidity (%) | CO (ppm) | methane (ppm) | CO2 (ppm) | System Output | Real Condition |
|---|---|---|---|---|---|---|
| 40 | 44 | 0,0204 | 0,0001 | 0,006 | Normal | Normal |
| 44 | 38 | 0,0204 | 0,0001 | 0,006 | Normal | Api |
| 54 | 35 | 0,0204 | 0,0001 | 0,006 | Api | Api |
| 58 | 20 | 0,0204 | 0,0001 | 0,006 | Api | Api |
| 60 | 3 | 1,67 | 0,039 | 0,84 | Api | Api |

The next test was conducted using a threshold with a combination of AND-OR logic. The AND logic was applied to the CO, $CO_2$, and methane gas parameters, while the OR logic was applied to the temperature and humidity parameters. The test results are presented in Table 4. In the results of the system test using the combined AND-OR logic, a false alarm occurred when the temperature reached 44°C and the humidity was 38%, even though an actual fire was already present. The system failed to detect the fire because the output values did not reach the predefined thresholds.

Subsequently, the fire and hazardous gas detection system was tested using an Artificial Neural Network (ANN) with a Radial Basis

Function (RBF) network architecture. This test utilized the trained model based on raw data, as shown in Table 5. The test conditions were kept the same as those used in the threshold algorithm testing.

Table 5. Result of ANN RBF Network

| Temp ($^0$C) | Humidity (%) | CO (ppm) | methane (ppm) | CO2 (ppm) | System Output | Real Condition |
|---|---|---|---|---|---|---|
| 40 | 58 | 0,0204 | 0,0001 | 0,006 | Normal | Normal |
| 50 | 41 | 0,0204 | 0,0001 | 0,006 | Normal | Normal |
| 53 | 37 | 0,0204 | 0,0001 | 0,006 | Api | Api |
| 59 | 31 | 0,0204 | 0,0001 | 0,006 | Api | Api |
| 60 | 5 | 1,201 | 0,026 | 0,556 | Api | Api |

Based on the testing results presented in Table 5, it can be observed that no false alarms were detected. The output labels produced by the system matched the actual conditions. This outcome confirms that the model trained using the RBF network is indeed accurate.

In the testing of the basic fire and hazardous gas detection system, a total of 165 data samples were used. Among these, 14 output labels were incorrect. This indicates that the accuracy of the basic fire and hazardous gas detection system is 91.51%.

In contrast, when testing the fire and hazardous gas detection system using an Artificial Neural Network (ANN) with a Radial Basis Function (RBF) architecture, no false alarms were found. This means the ANN-based system achieved an accuracy rate of 100%.

Table 6 presents a comparison of the accuracy between the basic fire and hazardous gas detection system and the system utilizing the ANN-based RBF network.

Table 6. Comparative Accuracy System

| | Threshold detection system | ANN detection system |
|---|---|---|
| **Number of taken data** | 165 | 165 |
| **False alarm** | 14 | 0 |
| **Accuracy (%)** | 91,515 | 100% |

Based on the accuracy comparison between the two methods in the fire and hazardous gas detection system, the system using an Artificial Neural Network (ANN) with a Radial Basis Function (RBF) architecture performs better, as it produces no false alarms. In terms of detection time, the proposed system can detect fire and hazardous gases in an average of 2.1 seconds, whereas the threshold algorithm takes an average of 0.96 seconds. Although the ANN-based system requires slightly more time, the difference is not significant when considering its zero false alarm performance.

When compared to YOLO, which relies heavily on camera input and is affected by environmental conditions such as lighting (day/night) and the physical shape of the flame, the proposed method offers a clear advantage. It does not require a camera, making it suitable for all lighting conditions and unaffected by flame variations.

Furthermore, compared to fuzzy logic, where detection performance is highly dependent on the number of membership functions and the design of fuzzy rules, the proposed ANN-based method is more adaptive and reliable. Since it has been trained on actual sensor data, it provides more accurate and robust predictions, making it a superior choice for early fire and hazardous gas detection.

It is important to note that the objective of this study is not to build a large-scale fire prediction model, but to evaluate the performance of an ANN-based multisensor detection system under representative early fire conditions. Therefore, the dataset is considered sufficient to demonstrate system behavior and decision-making capability within the defined experimental scope.

## CONCLUSION

In this study, an Artificial Neural Network (ANN) with a Radial Basis Function Network (RBFN) architecture was implemented for a fire and hazardous gas detection system based on multisensor integration. The system incorporates gas, IR, temperature, and humidity sensors, all processed using a Raspberry Pi 3. The ANN-trained system successfully produced a model that can accurately predict fire occurrences with

an average response time of 2.1 seconds, and without false alarms under experimental conditions, outperforming conventional threshold-based methods in this study. Although the dataset size is relatively small, the collected data were obtained under carefully controlled experimental conditions and cover representative early-stage fire and non-fire scenarios.

**ACKNOWLEDGEMENTS**

This research was funded by Universitas Negeri Semarang (UNNES) through the Faculty of Engineering (FT) research scheme, as stated in the Budget Implementation Document (DPA) of FTUNNES Number: 96.22.3/UN37/PPK.05/ 2024.

**REFERENCES**

Abdusalomov, A., Baratov, N., Kutlimuratov, A., & Whangbo, T. K. (2021). An improvement of the fire detection and classification method using YOLOv3 for surveillance systems. *Sensors*, *21*(19). https://doi.org/10.3390/s21196519

Bahhar, C., Ksibi, A., Ayadi, M., Jamjoom, M. M., Ullah, Z., Soufiene, B. O., & Sakli, H. (2023). Wildfire and Smoke Detection Using Staged YOLO Model and Ensemble CNN. *Electronics (Switzerland)*, *12*(1). https://doi.org/10.3390/electronics12010228

Cheng, C., Sun, F., & Zhou, X. (2011). One fire detection method using neural networks. *Tsinghua Science and Technology*, *16*(1), 31–35. https://doi.org/10.1016/S1007-0214(11)70005-0

Ferdyansyah, F., Suhradi Rahmat, R., Education Park, J., Ki Hajar Dewantara, J., & Cikarang, N. (2022). Alat Pendeteksi Kebakaran dan Pemadam Api Otomatis Menggunakan Kontrol Arduino. In *Journal of Mechanical Engineering and Mechatronics* (Vol. 7, Issue 2).

Getu, B. N., Alahmed, A., Alshamsi, A., Alzaabi, A., Ali, M. Al, & Alshamsi, N. (2022). A GSM and Arduino Based Fire Detection and Control System. *International Journal of Emerging Technology and Advanced Engineering*, *12*(5), 8–14. https://doi.org/10.46338/ijetae0522_02

HW, E. A., TULLOH, R., HADIYOSO, S., & RAMADAN, D. N. (2021). Sistem Pemantauan dan Pendeteksi Kebakaran berbasis Logika Fuzzy dan Real-time Database. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, *9*(3), 577. https://doi.org/10.26760/elkomika.v9i3.577

Irfanianingrum, I., Rizal Chaidir, A., Sumardi, S., Aditya Rahardi, G., & Wahyu Herdiyanto, D. (2023). Sistem Pendeteksi Dini Kebakaran Hutan Berbasis Logika Fuzzy dengan Integrasi Telegram. *Emitor: Jurnal Teknik Elektro*, *22*(2), 120–127. https://doi.org/10.23917/emitor.v22i2.22019

Iskandar Alam, T. H., Soekarta, R., & Ramadhan, W. (2019). Rancang Bangun Prototype Alat Pendeteksi Kebakaran Menggunakan Arduino Uno Dilengkapi Pemadam Dan Notifikasi Sms Gateway. *Insect (Informatics and Security): Jurnal Teknik Informatika*, *5*(1), 21. https://doi.org/10.33506/insect.v5i1.1280

Kharisma, R. S., & Setiyansah, A. (2021). Fire early warning system using fire sensors, microcontroller, and SMS gateway. *Journal of Robotics and Control (JRC)*, *2*(3), 165–169. https://doi.org/10.18196/jrc.2372

Kusuma Dewi, A., Riesty Wijayanti, A., & Friandi Ramadhan, R. (2022). Prototype Pendeteksi Api Menggunakan Arduino Uno R3 ATMega 328P. *Journal of Systems Information Technology and Electronics Engineeering*, *2*(1), 1–5. http://e-journal.ivet.ac.id/index.php/jsitee

Muharam, M., Latif, M., Baharuddin, B., & Richaflor, I. (2020). Pencegahan Kesalahan Alarm dalam Sistem Pendeteksi Dini Kebakan dan Pemadaman Berbasis Internet of Things. *JITCE (Journal of Information Technology and Computer Engineering)*, *4*(02), 53–62. https://doi.org/10.25077/jitce.4.02.53-62.2020

Nakip, M., Guzelis, C., & Yildiz, O. (2021). Recurrent Trend Predictive Neural Network for Multi-Sensor Fire Detection. *IEEE Access*, *9*, 84204–84216. https://doi.org/10.1109/ACCESS.2021.3087736

Noorfirdaus, J. R., Virgian, D., & Sakti, S. Y. (n.d.). Sistem Pendeteksi Kebakaran Dini Menggunakan Sensor Mq-2 Dan Flame Sensor Berbasis Web. *Konferensi Nasional Ilmu Komputer*, 2020.

Nopriadi, N., & Fajrin, A. A. (2023). Design Of Automatic Fire Detection and Extinguishing Devices Using Arduino. *Sinkron*, *8*(1), 496–504.

https://doi.org/10.33395/sinkron.v8i1.12047

Rachman, F. Z., Yanti, N., Hadiyanto, H., Suhaedi, S., Hidayati, Q., Widagda, M. E. P., & Saputra, B. A. (2020). Design of the early fire detection based fuzzy logic using multisensor. *IOP Conference Series: Materials Science and Engineering*, *732*(1). https://doi.org/10.1088/1757-899X/732/1/012039

Rehman, A., Qureshi, M. A., Ali, T., Irfan, M., Abdullah, S., Yasin, S., Draz, U., Glowacz, A., Nowakowski, G., Alghamdi, A., Alsulami, A. A., & Węgrzyn, M. (2021). Smart fire detection and deterrent system for human savior by using internet of things (IoT). *Energies*, *14*(17). https://doi.org/10.3390/en14175500

Ren, X., Li, C., Ma, X., Chen, F., Wang, H., Sharma, A., Gaba, G. S., & Masud, M. (2021). Design of multi-information fusion based intelligent electrical fire detection system for green buildings. *Sustainability (Switzerland)*, *13*(6). https://doi.org/10.3390/su13063405

Salman Farhan, M., Sthevanie, F., & Nur Ramadhani, K. (2022). Video Based Fire Detection Method Using CNN and YOLO Version 4. *Indonesia Journal on Computing*, *7*(2), 65–78. https://doi.org/https://doi.org/10.34818/INDOJC.2022.7.2.654

Suwarjono, S., Wayangkau, I. H., Istanto, T., Rachmat, R., Marsujitullah, M., Hariyanto, H., Caesarendra, W., Legutko, S., & Glowacz, A. (2021). Design of a Home Fire Detection System Using Arduino and SMS Gateway. *Knowledge*, *1*(1), 61–74. https://doi.org/10.3390/knowledge1010007

Talaat, F. M., & ZainEldin, H. (2023). An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Computing and Applications*, *35*(28), 20939–20954. https://doi.org/10.1007/s00521-023-08809-1