© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712252

Design And Development of an Artificial Neural Network-Based Fire Detection System

BOLUWATIFE OLAYIWOLA DEMOKUN[1], AKINDELE SEGUN AFOLABI[2], KEHINDE USMAN ADEWUYI[3], TIMILEYIN DANIEL AJAYI[4], ABDUL-QUADRI HUJATULLAHI ALIYU[5], ABDULMALIK ADEWALE AJAYI[6], DANIEL OLUWATOBI AJIBOLA[7], AZEEZ OPEYEMI AZEEZ[8]

[1, 2, 3, 4, 5, 6, 7, 8] *Department of Electrical and Electronics Engineering, University of Ilorin*

Abstract- ***Fire outbreaks are often catastrophic, destroying lives and property. It is essential to develop an accurate and reliable fire detection system to safeguard lives and protect assets. Traditional fire detection approaches, such as using common sensors, are inaccurate and usually trigger false alarms. To address this issue and significantly improve the precision, accuracy, response time, and dependability of fire detection systems, we designed and developed an Artificial Neural Network (ANN)-based system. Our ANN-based fire detection system integrates six (6) sensors, including temperature, humidity, smoke, gas, flame, and light sensors, with an ESP32s microcontroller into a sensing node to maximize the properties of each sensor and reduce false alarms. Four sensing nodes were developed to capture environmental dimensions during data acquisition. A Central data Station (mainly comprising Raspberry Pi 3 B+ microcomputer, real-time clock, display screen, buzzer, and indicators) was also developed to serve as the processing device and the central hub for decision making. The ground truth was established using a manual switch attached to the Central data station (i.e., 'ON' or a '1' for fire scenarios and 'OFF' or a '0' for no fire situations), and data were collected on a Google sheet. The collected data was processed and used to train ANN models of different architectures and hyperparameters on the Central Data Station. The best model was selected using the F1-score evaluation metric. The trained model was deployed to make predictions in real-time. Compared to many conventional systems, the system demonstrated exceptional accuracy of 95% with a false alert rate of less than 3%. Additionally, the system's relative response time is good; on average, fires were detected within 10 seconds of their start. By providing enhanced security and ensuring a prompt response in the event of a fire, this state-of-the-art fire detection system offers a competitive alternative to traditional methods.***

Index Terms- ***Fire Detection, Artificial Neural Networks, Sensors, Microcontroller, Accuracy, Response Time.***

I. INTRODUCTION

Artificial Neural Networks (ANN), inspired by the Human brain, are known for their ability to learn their own useful features (from input features) for effective prediction [1]. This property of ANNs enables ANNs to fit complex data better in classification problems (as compared to other classification supervised learning algorithms like Logistic regression), such as Fire or no-fire binary classification. An ANN network basically consists of an Input Layer, Hidden layer(s), and an Output Layer [1], which all comprise units also called neurons. Each layer is interconnected in such a way that each unit in the next layer is connected to all units in the previous layer [2], as shown in Figure 1.1. The number of units in the input layer is usually fixed to the number of features x; hidden layers can be one or more with multiple units; output layer units are usually fixed to the number of classes, with one unit used in binary classification [3] and more units in multi-class classification. [4]

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712252

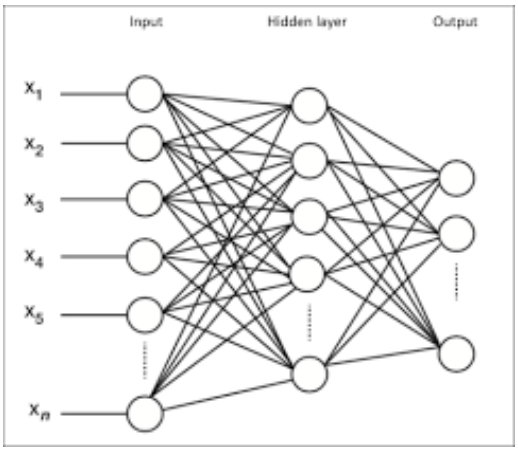


Figure 1.1 A simple Artificial Neural Network Architecture [2]

Each connection between neurons is associated with a weight [5], which is adjusted during the training process [5], [6]. The weighted sum of an input to a unit is calculated using a dot product operation [7]. Given an input vector x and a weight vector w, the weighted sum z of a unit can be calculated as:

$$z = x \cdot w + b \qquad (1.1)$$

Where:
z is the weighted sum, also called the logit
x is the input vector of the unit
w is the weight vector of the unit
b is the bias term of the unit.

The logits (z) are inputted to an activation function to determine the output of a neuron. Common activation functions [8] include Linear, sigmoid, ReLU (Rectified Linear Unit), and hyperbolic tangent, represented mathematically by:

$$\text{Linear activation } (a) = g(z) = z \qquad (1.2)$$

$$\text{Sigmoid activation } (a) = g(z) = 1/\,1 + exp(-z) \quad (1.3)$$

$$\text{ReLu activation } (a) = g(z) = Max\,(0, z) \quad (1.4)$$

$$\text{Hyperbolic tangent activation } (a) = tanh\,(z)\,(1.5)$$

To determine the final output of a Neural network, each layers up to the output layer are activated vectorially [9] using Matrix operations given by:

$$\text{Layer } (L) = g\,(WX^T + b) \quad (1.6)$$

Where:
g is the activation function used.
X is an m by n input feature matrix (m is the number of training examples and n is the number of input features)
W is a weight matrix of layer L with dimension j by n (j is the number of units in the layer)
b is the bias vector of the layer L.
The outputs of Layer (L-1) serve as the input X to Layer (L).

The process of computing prediction (activating the layers) from input to output is called Forward propagation [10]. In training an artificial neural network, a loss function (also called the cost function) measures the error between the predicted output and the true target values [11]. The Binary Cross-Entropy loss function [11], [12] was used. It is expressed as:

$$\text{Binary Cross-Entropy Loss} = -1/m \sum_{i=1}^{m} y_i \log p_i + (1-y_i) \log (1 - p_i) \qquad (1.7)$$

Where:
y is the true target (0 or 1)
p is the predicted probability value
m is the number of data samples.
The closer the loss value is to zero, the closer the predictions of the model are to the targets [12].

Back propagation algorithm [10], [13] is used in computing the gradient of the loss function with respect to the network's weights and biases, which is required in training an ANN model.

II. METHODOLOGY

This work was actualized in two phases, which are the development phase and the deployment phase.

2.1 The development phase
This phase includes:

I. System Design and Implementation
II. Data Collection and Processing
III. Model training and Performance evaluation

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712252

I. System Design and Implementation:

Four sensing nodes and a Central data Station were designed and implemented. Figure 2.1 shows a block diagram depicting the entire designed system.

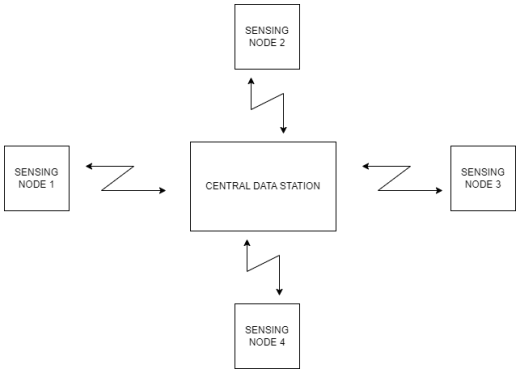


Figure 2.1 Block diagram of the entire system

Each sensing node comprises sensors (smoke detecting sensors, flame sensors, temperature sensors, gas sensors, light sensors), indicators, and an ESP32s microcontroller integrated to serve as an input device for data acquisition in the data collection phase and a real-time data input in the system deployment phase. Figures 2.2, 2.3, and 2.4 show the block diagram of a sensing node, the circuit diagram, and the constructed sensing node, respectively

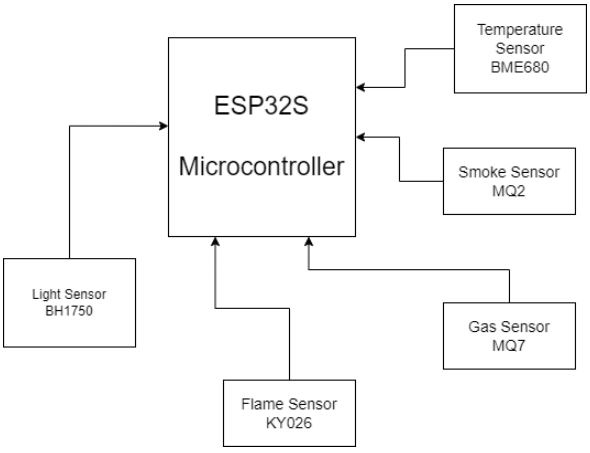


Figure 2.2 Block diagram of a sensing node

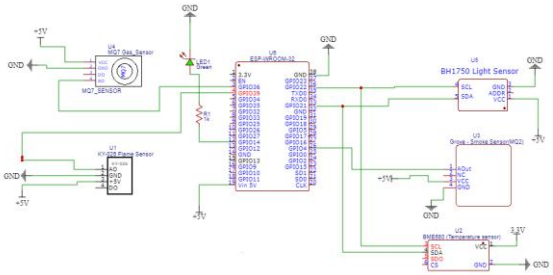
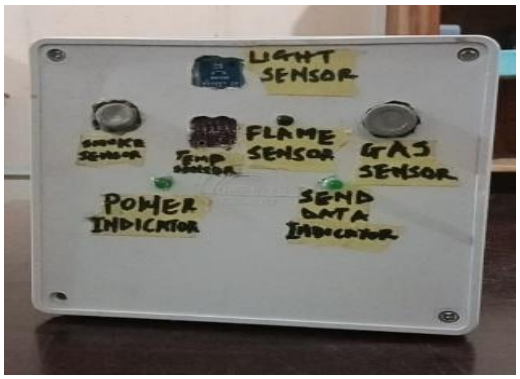


Figure 2.3 Circuit diagram of a sensing node



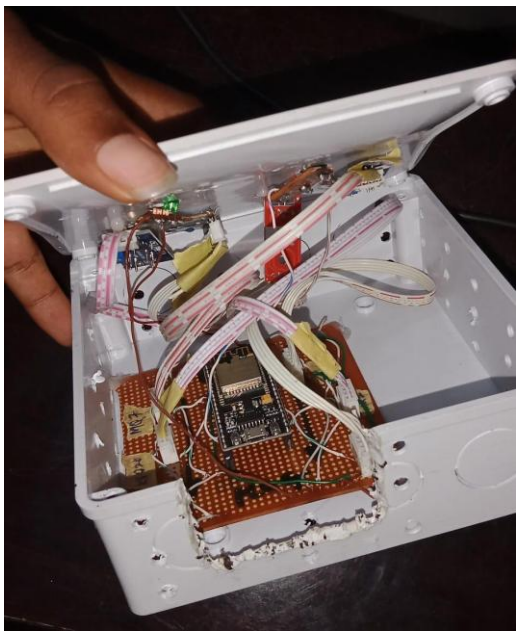


Figure 2.4 Constructed sensing node

The Central data station, comprising the Raspberry Pi 3 B+ microcomputer, a ground truth switch, indicators, a buzzer, a display screen, and a real-time clock, serves as the processing hub. Figures 2.5, 2.6, and 2.7 show the block diagram, circuit diagram, and constructed Central data station.

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712252

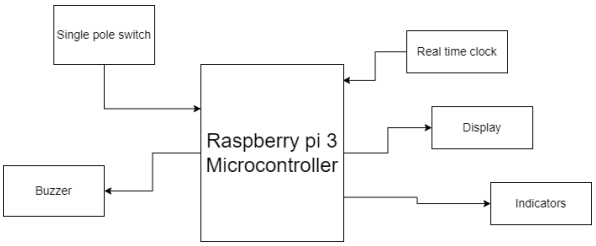


Figure 2.5 Block diagram of the central data station

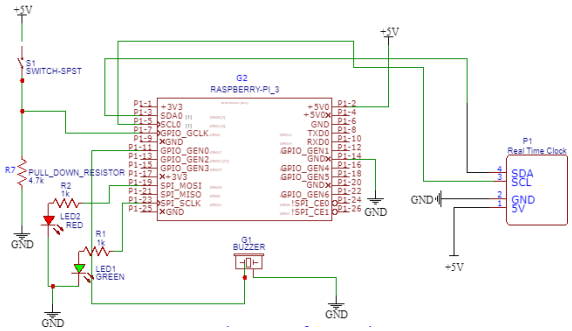


Figure 2.6 Circuit diagram of the central data station

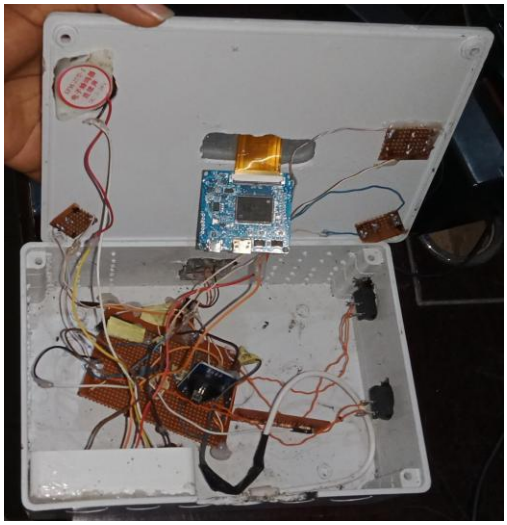


Figure 2.7 Constructed central data station

II. Data Collection and Processing:

The designed sensing nodes were placed around a fire source strategically (all at a distance between 1m and 5m at different times to form different patterns) in both open and confined spaces and data using three different fire scenarios (gas fire, fire from burning soft solids like papers, and wild fire created from burning woods) and a no fire scenario were gathered across seasons in 6 months. The sensing nodes were programmed to stack data from each calibrated sensor of a sensing node into one row and each row from all four sensing nodes into a single tracked list. The data list was then sent to the central data station every 2 seconds through the MQTT communication protocol [14]. The central data station reads the real time clock and the class label digital signal (1 or On indicating fire scenario and 0 or Off indicating no fire scenario) from the manual class label switch and automatically append the date, time and the class label or ground truth to the data list after which it then logs the data list in a Google sheet row after row in every time instance of 5 seconds. Figures 2.8, 2.9, and 2.10 show a snippet of the logged data sheet, a fire scenario, and a no-fire scenario.

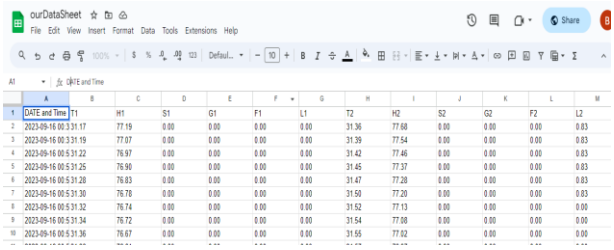


Figure 2.8 Snippet of the logged data sheet

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712252

Figure 2.9 fire scenario

Figure 2.10 A no-fire scenario

The data collected was visualized to better understand the data, randomized to avoid any potential biases in the dataset, cleaned and processed to remove outliers, duplicates, and handle missing values. Data from the light sensors were observed to have a random pattern across different fire and no fire scenarios. These values were removed from each row. Figures 2.11 and 2.12 show a snippet of the processed data and a heatmap of the data after processing.

	Temperature1	Humidity1	Smoke1	Gas1	Flame1	Temperature2	Humidity2	Smoke2	Gas2	Flame2	...	Humidity3	Smoke3	Gas3	Flame3	Temper
0	29.11	71.40	0.0	0.0	0.0	29.54	68.48	0.0	0.0	0.0	...	67.09	0.0	58.0	0.0	
1	40.49	50.47	0.0	0.0	0.0	41.45	41.83	0.0	0.0	0.0	...	44.87	0.0	7.0	0.0	
2	28.93	71.79	0.0	0.0	0.0	29.17	68.45	0.0	0.0	0.0	...	68.76	0.0	59.0	0.0	
3	32.08	60.59	0.0	16.0	0.0	62.17	57.22	0.0	265.0	0.0	...	62.77	0.0	26.0	0.0	
4	33.49	57.16	0.0	17.0	0.0	60.46	52.44	0.0	0.0	0.0	...	57.40	0.0	33.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4659	38.99	54.41	0.0	0.0	0.0	40.60	43.97	0.0	0.0	0.0	...	46.00	0.0	12.0	0.0	
4660	30.45	67.77	0.0	7.0	0.0	29.98	68.54	0.0	0.0	0.0	...	66.53	0.0	53.0	0.0	
4661	30.45	68.30	0.0	8.0	0.0	30.00	68.91	0.0	0.0	0.0	...	66.43	0.0	56.0	0.0	
4662	30.50	68.36	0.0	10.0	0.0	30.04	69.20	0.0	0.0	0.0	...	66.45	0.0	58.0	0.0	
4663	30.38	67.80	0.0	9.0	0.0	29.92	68.49	0.0	0.0	0.0	...	66.56	0.0	55.0	0.0	

Figure 2.11 A snippet of the cleaned data

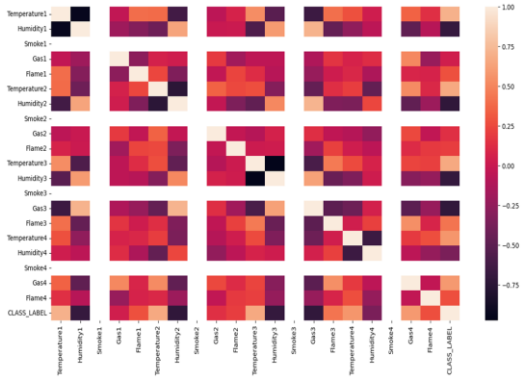


Figure 2.12 Heat map of the data set. Feature Correlation: A Heat Map shows correlation between features. Very light indicates high positive correlation and while dark indicates high negative correlation

III. Model Training and Performance Evaluation:

An ANN-model of different architectures were trained on different train/test split dataset (60-40, 70-30, and 80-20) the input layer was fixed to the number of features and the output layer have a single unit for binary classification (fire event or no fire event), the sigmoid activation function (equation 1.3) was used in all units. The dataset features were scaled using the scikit-learn standard scalar library. (This was done to ensure that the features have comparable ranges, thus making the model training faster and more effective.) TensorFlow and other relevant Python libraries on a Jupyter notebook were used in training the models on a personal computer. The F1-score metric was used to evaluate the performance of each model on the test sets.

F1-Score = 2 × Precision × Recall / (Precision + Recall) ----------(2.1)

Where:

Precision = True Positives / (True Positives + False Positives) ----------(2.2)

Recall (Sensitivity) = True Positives / (True Positives + False Negatives) --------- (2.3)

In this formula:
True Positives (TP) are the number of correct positive predictions.
False Positives (FP) are the number of incorrect positive predictions.
False Negatives (FN) are the number of actual positive instances that were incorrectly predicted as negative.

Table 2.1 shows the performance of each architecture on each train/test split dataset.

Data set	Hidden layer units	Train/test split (%)	Accuracy (%)	F1-score (decimal)
Unprocessed	10	60/40	99.7	0.997
		70/30	98.6	0.986
		80/20	100	1.0
	15	60/40	100	1.0
Processed	1	60/40	87.90	0.935
		70/30	87.88	0.935
		80/20	100	1.0
	3	60/40	100	1.0
		80/20	100	1.0
	5	60/40	100	1.0

Table 2.1 Performance of the models

Figure 2.13 is a flowchart showing the development phase.

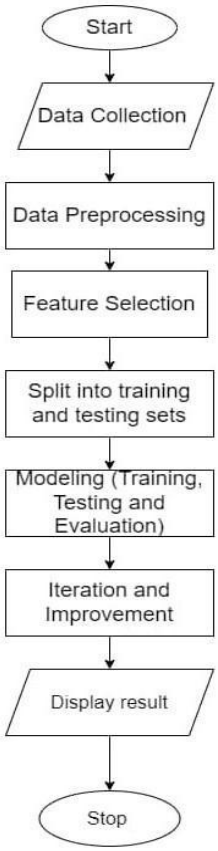


Figure 2.13 The development phase flowchart

2.2 The deployment and testing phase

On the processed data, different model architectures achieved a perfect F1-score, as shown in Table 2.1. The trained artificial neural network model with 5 hidden layers was saved and exported in PKL format using the joblib library in Python and Jupyter Notebook. This model was deployed on the Central data station. The deployed model was tested on real-time data from the sensing nodes in both fire and non-fire scenarios. To avoid false alerts, the central data station was programmed to give a prediction every 3 seconds by first processing the in-stream data from the sensing nodes to the model standard and combining them with the deployed model to give a prediction. prediction samples within every 3 seconds are compared, after which the system gives a final prediction based on the majority output. If the final prediction is a 1 (i.e, fire detected), it automatically activates the buzzer, displays fire on the screen, and blinks the red indicator to alert the user. The system was tested 1012 times in an open and confined space

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712252

environment with 717 fire scenarios and 295 no-fire scenarios. The system gave a correct prediction 972 times with 683 correct fire predictions and 289 correct no-fire predictions, indicating an accuracy of 95% and a false alert rate of 2.03%. Figures 2.14 and 2.15 show a flowchart of the system in operation during testing and the system in operation. The system was able to detect fire in an average of 10 seconds from start time.

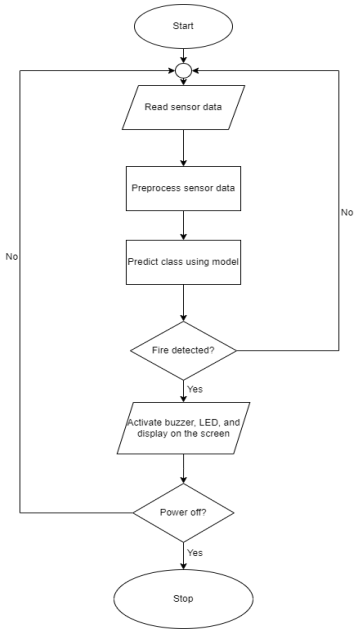


Figure 2.14 Operation flowchart

Figure 2.15 System in Operation

III. RESULTS, DISCUSSION, AND FUTURE DIRECTIONS

The following results were obtained from the testing phase:

I. Accuracy: The deployed artificial neural network model achieved an accuracy of 95% in detecting fires, indicating a high success rate.
II. False Positives: There were a small number of false positives (6 times), resulting in a false alert rate of 2.03%, mainly caused by factors such as dust or temperature fluctuations.
III. Response Time: The system can detect fire in an average time of 10 seconds from the start time.

Table 3.1 summarizes the results from the testing phase.

	Number of Events
Testing	1012
Fire Scenario	717
No-Fire Scenario	295
Total Correct Prediction	972
Correct Fire Prediction	683
Correct No-Fire Prediction	289
Detection Accuracy (%)	95.2
System Average Response Time (s)	10
False Alert rate (%)	2.03

Table 3.1 System Testing Result

The sensors were found to have an average sensing range of approximately 1 m, which limits the ability of the sensing nodes to capture environmental dimensions effectively. This limitation can be addressed by either deploying more sensing nodes throughout the controlled environment or by using sensors with better sensing ranges. Gathering data across seasons, in different environments, and with different fire sources ensures robustness of the system to seasonal changes, environmental changes, and different types of fire. The data gathering process is tedious and time-consuming, and there was

insufficient data to encompass various environmental seasons, conditions, and changes. This hindered the model's ability to generalize effectively. This issue can be addressed by allocating more time to data collection and ensuring data is captured across different seasons, which will, in turn, increase the accuracy of the system. In the future, the system can be trained with a combination of data from classes A, B, C, D, and K fires [15] to ensure more robustness. The model's response time is not swift, averaging up to 10 seconds for predictions. This delay can potentially be reduced by incorporating components with faster response times and by increasing the number of sensors to capture additional features, thereby improving the system's response time.

IV. CONCLUSION

In the effort to improve fire detection systems for swiftness in safeguarding lives and property in the event of fire, this work, which utilized an Artificial Neural Network algorithm in fire detection, achieved a good response time, accuracy, and false alert rate. Integrating an Artificial neural network algorithm in traditional systems redefined the effectiveness and efficiency of fire detection systems, increasing the reliability of these systems. This developed system has the potential to achieve better results with more training data gathered across different seasons, fire sources, and environmental conditions.

REFERENCES

[1] R. Dastres and M. Soori, “Artificial neural network systems,” International Journal of Imaging and Robotics (IJIR), vol. 21, no. 2, pp. 13–25, 2021.

[2] T. Tyagi, H. Sharma, K. Jain, and C. Mittal, “Handwritten Digit Recognition System using Machine Learning,” International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 12, no. 5, pp. 3150–3154, May 2024. doi: 10.22214/ijraset.2024.62254

[3] S. A. Abdullah and A. Al-Ashoor, “An artificial deep neural network for the binary classification of network traffic,” International Journal of Advanced Computer Science and Applications, vol. 11, no. 1, pp. 402–408, 2020.

[4] C. Joshi, R. K. Ranjan, and V. Bharti, “ANN based multi-Class classification of P2P Botnet,” pp. 1319–1325, 2021.

[5] W. F. Schmidt, M. A. Kraaijveld, and R. P. W. Duin, “Feed forward neural networks with random weights,” in Proc. Int. Conf. on Pattern Recognition, IEEE Computer Society Press, 1992, pp. 1–1.

[6] F. Günther and S. Fritsch, “neuralnet: Training of neural networks,” 2010.

[7] A. Krenker, J. Bešter, and A. Kos, “Introduction to the artificial neural networks,” in Artificial Neural Networks: Methodological Advances and Biomedical Applications, InTech, 2011, pp. 1–18.

[8] J. Lederer, “Activation functions in artificial neural networks: A systematic overview,” arXiv preprint arXiv:2101.09957, 2021.

[9] A. Lichtner-Bajjaoui, “A mathematical introduction to neural networks,” 2021.

[10] M. Cilimkovic, “Neural networks and back propagation algorithm,” Institute of Technology Blanchardstown, no. 1, pp. 18, 2015.

[11] N. Zhang, S.-L. Shen, A. Zhou, and Y.-S. Xu, “Investigation on performance of neural networks using quadratic relative error cost function,” IEEE Access, vol. 7, pp. 106642–106652, 2019.

[12] A. Buja, W. Stuetzle, and Y. Shen, “Loss functions for binary class probability estimation and classification: Structure and applications,” Working Draft, Nov. 2005, pp. 13.

[13] M. H. Sazlı, “A brief review of feed-forward neural networks,” Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering, vol. 50, no. 1, 2006.

[14] D. Soni and A. Makwana, “A survey on MQTT: a protocol of Internet of Things (IoT),” in Proc. Int. Conf. on Telecommunication, Power Analysis and Computing Techniques (ICTPACT-2017), vol. 20, Apr. 2017.

[15] R. Ponomarenko et al., “Review of the environmental characteristics of fire

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712252

extinguishing substances of different composition used for fires extinguishing of various classes," 2019.