

ARDUINO BASED FIRE DETECTION WITH GSM ALERT USING ML

P.Sindhu Priya¹, K.Pushpa², CH.R.Dhana Sri³, M.B.M.S Gupta⁴, P.J Mahesh Babu⁵
K.Nagaraju⁶

^{1,2,3,4,5}B.Tech Students, ⁶Assistant Professor, Department of Electronics and Communication Engineering, Sri Vasavi Engineering College, Tadepalligudem, Andhra Pradesh, India

Abstract - Fire accidents can cause damage to life and property, especially in forest areas, homes, industries, and public areas. Detecting fire at an early stage is very crucial to reduce risks and take timely action. In this project, a fire detection system is developed using multiple sensors based on ARDUINO UNO. The system continuously monitors temperature, gas levels, and flame presence to identify fire conditions. To improve the decision-making process, simple logic inspired by decision tree algorithms is used in machine learning, allowing the system to evaluate sensor data and determine the likelihood of a fire based on predefined thresholds. When a fire is detected, the system immediately activates a buzzer and sends an alert message using the GSM module. The proposed system is simple, cost-effective, and suitable for real-time fire safety applications.

Key Words: Fire detection, ARDUINO, sensors, machine learning, decision tree algorithm, and GSM module.

1. INTRODUCTION

Fire accidents are one of the most common hazards that can lead to severe damage and loss of life. In many cases, the damage becomes worse because the fire is not detected early. Traditional fire detection systems usually depend on fixed limits, such as temperature or smoke levels. These systems usually depend on fixed limits, such as temperature and smoke levels. These systems may sometimes give false alarms or fail to detect fire in the early stages. With the help of modern embedded systems, it is possible to design more reliable solutions. In this project, a system is developed using a microcontroller based on ARDUINO UNO along with multiple sensors. By combining data from different sensors and applying simple logic inspired by machine learning, the system can identify fire conditions more accurately and respond quickly.

2. LITERATURE REVIEW

Many fire detection systems have been developed using sensors such as smoke detectors and temperature sensors. While these systems are useful,

they often rely on a single parameter, which can lead to incorrect results in some situations. Some recent systems use internet-based technologies to monitor fire conditions remotely. However, these systems depend on stable network connections. Machine learning-based systems have also been introduced to improve accuracy and reduce false alarms, but they can be complex and expensive. This project focuses on a simple and practical approach by combining multiple sensors with basic intelligent logic to improve reliability without increasing complexity.

Recent advancements in fire detection systems focus on improving early detection, accuracy, and real-time alert mechanisms. Sharma et al [1] developed a GSM-based fire alert using ARDUINO to send SMS notifications during fire accidents. CHINCHMALATPURE et al. [2] proposed an IOT-based system integrating multiple sensors for real-time monitoring and control. Additionally, UMOH et al. [3] introduced an ml approach using SVM to reduce false alarms. More recent studies use IOT and deep learning techniques for efficient fire detection. AI-based models such as convolutional neural networks (CNN) and YOLO have shown high accuracy in detecting fire smoke in real-time environments [4], [5]. Overall, existing systems demonstrate improved performance.

3. PROBLEM STATEMENT

Existing fire detection systems may not always be monitoring accurate values because they depend on fixed threshold values. This can lead to false alarms or to detecting fire too late. In addition, some advanced systems require internet connectivity, which may not always be available on the internet. Therefore, there is a need for a reliable, low-cost system that can detect fire accurately and work independently.

4. OBJECTIVES OF THIS PROJECT

- * To detect fire at an early stage
- * To reduce false alarms
- * To improve accuracy

5. PROPOSED SYSTEM

The proposed system focuses on developing a simple and efficient fire detection solution using a combination of sensors and a microcontroller. Unlike traditional systems that depend on fixed threshold values, this system uses multiple sensors and intelligent decision logic to improve detection accuracy. In this system, a flame sensor and temperature sensor are connected to the ARDUINO UNO. These sensors continuously monitor environment conditions such as heat, smoke, and flame presence. The collected data is processed by the microcontroller to identify abnormal situations. To enhance decision-making, a machine learning model is trained using sensor data collected under different conditions. The model generates decision rules that help in distinguishing between fire and non-fire scenarios. These rules are then implemented in the ARDUINO program. When a fire condition is detected, the system activates a buzzer and LED for immediate alert. At the same time, a GSM module sends an SMS notification to a predefined mobile number. This ensures both local and remote alerting. The system is designed to be low-cost, reliable, and suitable for areas where internet connectivity is not available.

6. SYSTEM ARCHITECTURE

The system consists of three main parts: sensing unit, processing unit, and alert unit. The sensors collect environmental data and send it to the microcontroller. The microcontroller processes the data and decides whether a fire condition exists. If a fire is detected, the alert unit activates a buzzer and sends a message to the user.

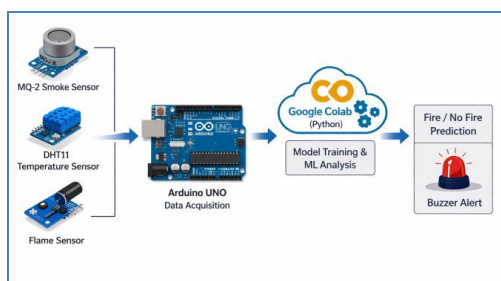


Figure1: BLOCK DIAGRAM

7. METHODOLOGY

The system follows a step-by-step approach for fire detection. First, sensors are used to collect environmental data such as temperature, gas levels, and flame presence. This data is continuously

monitored by the ARDUINO. Next, the collected data is used to train a machine learning model. The model learns patterns that represent fire and normal conditions. After training, the decision rules are extracted and implemented in the ARDUINO code. During real-time operation, the ARDUINO compares sensor values with these rules. If the conditions indicate fire, the system activates alerts using a buzzer and GSM module. Otherwise, it continues monitoring. This approach improved accuracy and reduced false alarms while keeping the system simple and efficient.

8. FLOWCHART

The flowchart explains the working process of the system. It starts with initializing the system and sensors. Then, the system continuously reads sensor values and checks them against predefined conditions. If a fire condition is detected, the system activates the buzzer and LED and sends an alert message to mobile through the GSM module. Otherwise, it continues monitoring.

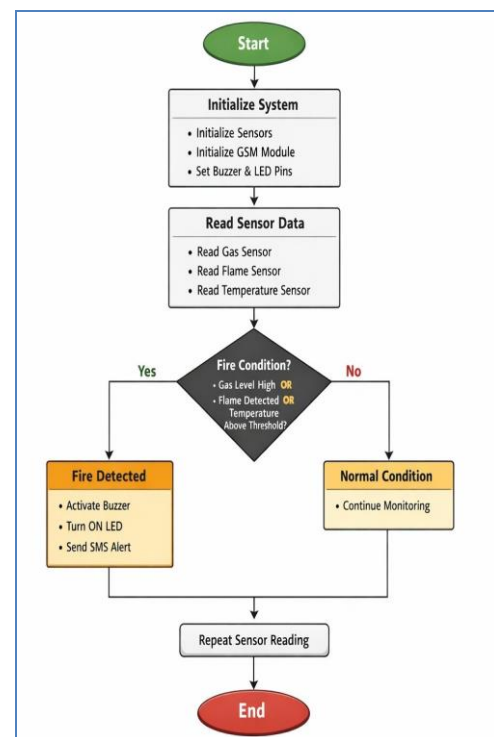



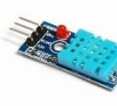




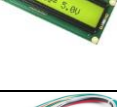




Figure2: FLOWCHART

9. HARDWARE REQUIREMENTS

Sl. No	Components Name	Specification	Figure
1.	Arduino UNO	ATmega328P	
2.	Flame Sensor (IR-Based)	Ir-Based Flame Detection Module	
3.	Gas Sensor	MQ-2	
4.	Temperature sensor	DHT11	
5.	Buzzer	5V active buzzer	
6.	GSM Module	SIM900L	
7.	Breadboard	Standard Bread board	
8.	Led indicator	5V LED	
9.	LCD	16x2 Display	
10.	Connecting wires	Jumper wires	
11.	power supply or battery	5V DC	

10. SOFTWARE REQUIREMENTS

1. Arduino IDE

-Used to program Arduino NANO
-Reads sensor data and controls buzzer, LED, and GSM

2. Google Colab using Python

-Used to train decision tree ML model
-Processes collected sensor data
-Used for data processing & ML implementation

11. IMPLEMENTATION

The fire detection system is implemented using an Arduino Nano and multiple sensors to detect fire conditions effectively. In this project, a flame sensor, gas sensor (MQ-2), and temperature sensor (DHT11) are used to monitor the environment.

First, all the components are connected properly. The sensors are connected to the input pins of the Arduino, while the buzzer and LED are connected to the output pins. The GSM module is connected using serial communication to send alert messages.

After completing the hardware connections, the Arduino is programmed using the Arduino IDE. The program is written to read data from all sensors continuously. The sensor values are checked using certain conditions to identify fire situations.

A simple logic based on machine learning is used in this project. The decision tree model is trained separately, and its rules are converted into conditions in the Arduino program. For example, if the temperature is high and gas is detected or if a flame is present, the system identifies it as a fire condition.

When a fire is detected, the buzzer starts ringing and the LED turns on to give an alert. At the same time, the GSM module sends a warning message to a predefined mobile number.

The system continuously monitors the environment and responds quickly when abnormal conditions are detected. The project is tested under different conditions, and it works properly by detecting fire and sending alerts without delay.

12. RESULTS AND DISCUSSION

The proposed fire detection system was successfully implemented using both hardware components and software techniques. The system was tested under different conditions to verify its performance. During testing, under normal conditions, the system displayed a safe status, and no alert was generated.

Data Set:

Gas	Temperature	Flame	Fire
200	28	1	0
250	30	1	0
300	32	1	0
350	27	1	0
400	33	1	0
450	29	1	0
500	31	1	0
150	26	0	0
280	30	1	0
380	28	0	0
800	55	0	1
850	58	0	1
900	60	0	1
750	52	0	1
780	57	0	1

When fire conditions were simulated by increasing temperature, introducing gas, or detecting flame, the system responded immediately. The LED indicator turned on, and the buzzer produced an alarm sound. At the same time, the GSM module sent an alert message to the registered mobile number. The LCD also displayed a fire warning message. The software model was tested using a dataset of 15 samples. The decision tree algorithm correctly classified all samples into fire and non-fire conditions. The system achieved an accuracy of 100%, indicating correct prediction for all test inputs. For example, when the input values were gas=800, temperature=54 centigrade, and flame=0, the system successfully predicted FIREDETECTED.

```

Dataset
  gas  temperature  flame  fire
0  200           28      1      0
1  250           30      1      0
2  300           32      1      0
3  350           27      1      0
4  400           33      1      0
5  450           29      1      0
6  500           31      1      0
7  150           26      0      0
8  280           30      1      0
9  380           28      0      0
10 800           55      0      1
11 850           58      0      1
12 900           60      0      1
13 750           52      0      1
14 780           57      0      1

Dataset size: 15 rows

Accuracy: 1.0
Enter Gas value: 800
Enter Temperature: 54
Enter Flame value (0 or 1): 0
🔥 FIRE DETECTED
    
```

Figure3: Machine learning code output

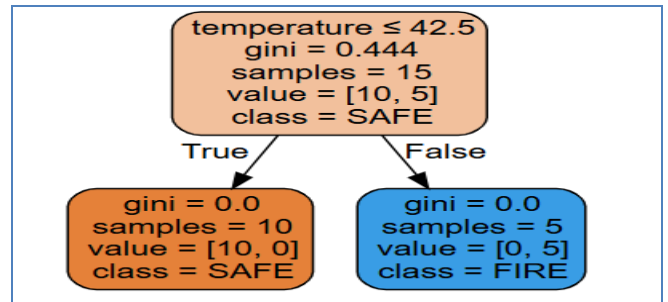


Figure4: Decision Tree Diagram from ml code

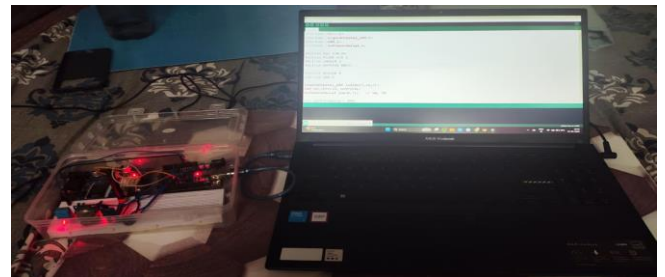


Figure5: Hardware output

Accuracy Calculation:

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

- Fire (1) - Positive class
- Non-fire (0)-Negative class
- From Dataset have 15 samples:
- Non-fire (0) samples=10
- Fire (1) samples=5
- *TP (True Positive) = 5
- fire correctly detected
- *TN (True Negative) = 10
- safe condition correctly detected
- *FP (False Positive) = 0
- No false alarms
- *FN (False Negative) = 0
- No fire missed

CONFUSION MATRIX TABLE:

Parameters	Predicted Fire	Predicted safe
Actual Fire	TP=5	FN=0
Actual Safe	FP=0	TN=10

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

$$\text{Accuracy} = \frac{5+10}{5+10+0+0}$$

$$\text{Accuracy} = \frac{15}{15} = 1.0$$

ACCURACY=100%

DISCUSSION:

The results show that the system performs efficiently in detecting fire conditions. The use of multiple sensors improves reliability compared to single-sensor systems.

The decision tree model helps in making better decisions about sensor values. It is simple and suitable for implementation in Arduino, as it uses rule-based logic instead of complex computations.

The system provides fast responses and real-time alerts through a buzzer, LED, and GSM communication. However, the accuracy is calculated using a small dataset, so performance may vary in real-world conditions.

Overall, the system is effective, low-cost, and suitable for practical fire detection applications.

13. CONCLUSION

In this project, a fire detection system was developed using ARDUINO UNO along with gas, temperature, and flame sensors. The system is designed to continuously observe environmental conditions and identify possible fire situations at an early stage.

To improve the decision-making process, a simple machine learning technique (decision tree) was used. The rules obtained from the model were applied in the ARDUINO program, which helped the system to make more accurate decisions based on multiple sensor inputs. During testing, the system responded quickly whenever abnormal conditions were created. The buzzer and LED provided immediate local alerts, and the GSM module successfully sent warning messages to the user. The system was able to detect fire conditions correctly without unnecessary alerts in normal situations. Overall, this project demonstrates a practical, cost-effective approach to fire detection. It can be used in different environments to improve safety and reduce the risk of fire-related damage.

REFERENCE

1. IJARIE-ISSN (0)-2395-4396
2. A.Sharma, R. Gupta, and S. Verma, "GSM Based Fire Alert System Using ARDUINO , " International journal of Engineering Research and Technology (IJERT), vol. 10, no. 5, pp. 123-126, 2022.
3. S. CHINCHMALATPURE, A. SAGAR, and R. LOHE " IoT-based smart fire detection and control system using GSM module", International

Journal for Research in Applied Science and Engineering Technology (IJRASCT), vol. 11, no. 3, pp. 456-460, 2023.

4. U. UMOH, E. UDO, and N. EMMANUEL, "support vector machine-based fire outbreak system," in proc. int. conf. Computing and communication systems, 2019, pp. 1-5
5. K.Sathish Kumar, R. Priya and M. Karthik, "Wireless Sensor Network Based Fire Monitoring System," International Journal of Advanced Research in Electrical and Electronics and Instrumentation Engineering (IJAREE), vol. 5, no. 4, pp.3021-3026, 2016