Applied Engineering, Innovation, and Technology

Research article

Internet of Things (IoT) Based Fire Detection and Suppression System

Dienatu Issaka[1,*], Yaw Amankrah Sam-Okyere[2], Emmanuel Osei-Kwame[3]

[1]Department of Engineering, Ashesi University, Berekuso, Eastern Region, Ghana
[2]Department of Electrical and Electronic Engineering, University of Mines and Technology, Tarkwa, Western Region, Ghana
[3]Department of Electronics Engineering, Norfolk State University, Norfolk, Virginia, United States
*Correspondence: issakadienatu@gmail.com

https://doi.org/10.62777/aeit.v2i2.70

Received: 29 April 2025
Revised: 13 June 2025
Accepted: 28 June 2025
Published: 1 October 2025

Abstract: Fire incidents cause significant threats to life and property, particularly in critical infrastructure. This research presents the design and implementation of an Internet of Things (IoT)-based fire detection and suppression system featuring real-time monitoring and scalable sensor integration. The system integrates an ATmega328p microcontroller, RF transceivers, flame and smoke sensors, NodeMCU (ESP8266), solenoid valves, relays, a jockey pump, and water sprinklers. Sensor fusion ensures high detection accuracy, triggering suppression only upon simultaneous smoke and flame detection to minimize false positives. Communication between transceivers controls the pump operation, while the NodeMCU transmits sensor data to a remote web server via Wi-Fi for continuous monitoring. When a fire is detected by the sensors, the controller promptly activates the fire alarm system, which in turn triggers the jockey pump to discharge water through the sprinkler system at the affected locations.

Keywords: IoT, microcontroller, ATmega328p, fire detection

1. Introduction

Fire safety includes strategic planning and infrastructure design aimed at reducing fire risks and minimizing the spread and impact of fire incidents. Key components include the use of fire-resistant construction materials, implementation of preventive protocols, enforcement of safe operating procedures, personnel training, and provision of flame-retardant protective equipment [1], [2], [3]. Fire-related incidents result in significant economic losses, injuries, fatalities, and operational disruptions annually, particularly within high-risk industries. Several studies have shown that proactive investment in fire prevention infrastructure and policy is important for workplace safety [4], [5], [6], [7]. Fire prevention programs, which include emergency response planning, fire detection systems, accessible evacuation routes, accommodations for disabled individuals, and strategically located fire extinguishers, are detailed in world standards [8]. Despite these measures, [9] reported 148 workplace fire-related fatalities in England in 2013. Globally, approximately 180,000 fire-related deaths occur annually, predominantly in low- and middle-income countries, with 95% attributed to fatal burns [10]. In South America alone, Brazil recorded over 68,635 wildfire outbreaks by August 2023, followed by Argentina with 41,300 cases [11], [12].

Table 1. Fire outbreak cases in Ghana 2016-2020.

Year	Cases	Casualties
2016	6,891	10,397
2017	4,759	7,887
2018	4,544	5,531
2019	1,622	4,593
2020	1,822	1,220

In Ghana, fire outbreaks are recurrent and often deadly. The Ghana National Fire Service (GNFS) reported property losses exceeding GH¢31 million in 2017 alone. From January 1 to December 21, 2020, a total of 5,966 fire incidents were recorded, resulting in 222 fatalities and 1,125 injuries [13]. According to [14], modern fire safety management requires not only suppression but also automation for prompt and efficient response, especially through systems that can operate independently of human intervention. Fire risk mitigation in the oil and gas sector is critical due to the highly flammable nature of petroleum-based products [15]. At certain energy companies in Ghana, the current fire suppression system relies on manual activation of jockey pumps following smoke detection. This manual process introduces latency and is prone to human error, reducing the overall effectiveness of the firefighting effort [16], [17]. There is thus a critical need to automate the activation of the suppression system to enhance responsiveness and reliability. The main objective of this research is to design and implement an automated fire detection and suppression system with integrated Internet of Things (IoT) capabilities for real-time monitoring and scalable sensor deployment. The sub-objectives are:

- To eliminate human error by automating the activation of the jockey pump,
- To design a fault-tolerant and easily expandable sensor network for fire detection,
- To develop an IoT-enabled interface for remote monitoring and control of the fire suppression process.

Several fire detection and suppression systems have been developed in recent years, each with different levels of complexity, cost, and scalability. The authors in [18] proposed an Arduino-based system designed to detect gas leakages and monitor ambient temperature using integrated sensors. The system also featured automated control measures and alert notifications through SMS. However, its effectiveness was limited by delayed alert delivery and a lack of real-time response accuracy, which could be critical during emergencies. The authors in [19] introduced an innovative IoT- and cloud-based drone monitoring system that collects environmental data to detect and prevent fire incidents. Their system emphasized real-time monitoring and situational awareness, but its deployment was less effective in multi-environment conditions, such as distinguishing between indoor, industrial, and outdoor settings, reducing its reliability across diverse use cases.

In [20], the authors developed FireNet, a lightweight deep learning-based fire and smoke detection model specifically tailored for real-time IoT applications. The model was optimized to operate efficiently on resource-constrained devices, making it cost-effective and suitable for edge computing. While it showed high accuracy in detecting fire outbreaks, it was limited to detection only and lacked any built-in suppression or emergency response mechanism. The authors in [21] designed a comprehensive system capable of detecting, classifying, and monitoring fire events using a combination of temperature sensors and intelligent algorithms. This system provided context-aware fire classification and monitoring capabilities, but its reliance on complex machine learning algorithms posed implementation challenges, especially for low-resource settings with limited computing infrastructure. The integration of edge computing in UAV-enabled early forest fire detection was the focus of the authors in [22]. They proposed a three-layer architecture involving sensing, processing, and communication layers to ensure faster response times and localized data processing. While this approach enabled rapid large-scale fire detection, its high computational requirements and deployment costs made it less feasible for small-scale or resource-constrained applications.

In [23], a cloud-enabled IoT-based embedded system that integrated fire detection with intelligent lighting and ventilation controls was implemented. Their system used HTTP protocols for data exchange and enabled monitoring through smartphones and PCs. Although the solution demonstrated versatility in smart building environments, it lacked robustness and scalability for real-life applications in industrial or large public infrastructure. A study in [24] presented a simple and low-cost wireless fire detection system using Arduino and flame/smoke sensors. It aimed at providing early fire alarms to occupants. Despite its cost-efficiency and ease of implementation, the system was not reliable in situations where smoke was present without a corresponding temperature increase, making it insufficient for certain fire scenarios. In [25], the authors proposed a smart emergency response framework that leverages IoT for fire hazard mitigation. Their system provided a standardized communication protocol to interface with emergency services and enhance responsiveness during fire outbreaks. However, it did not integrate mechanisms to confirm or control fire suppression actions, limiting its application in fully automated firefighting systems.

Table 2. Comparative analysis of related fire detection systems.

Ref.	Detection Method	Suppression	Communication	Cloud Integration	Limitations
[18]	Gas & Temp Sensors	No	SMS (Unidirectional)	No	Alert delay, no automation
[19]	IoT + Drones	No	Cloud (Partial)	Yes	Poor indoor adaptability
[20]	Lightweight Deep Learning	No	Local	Partial	Detection-only
[21]	Deep CNN	No	Local	No	High computational load
[22]	UAV + Edge Layers	No	Edge/Local	No	High cost, complex
[23]	IoT + Smart Building	No	HTTP (Unidirectional)	Yes	Limited scalability
[24]	Arduino + Flame/Smoke	No	Local	No	Inaccurate for smoke-only
[25]	IoT Emergency Framework	No	Standardized Alerts	No	No suppression integration
This study	Smoke + Flame (Dual Sensor)	Yes	RF + Wi-Fi (Bidirectional)	Yes (ThingsBoard MQTT)	Fully integrated, scalable

Numerous studies have addressed the challenges of fire detection and suppression; however, most existing systems focus primarily on fire detection while depending on external intervention, such as fire service personnel or safety agents, for actual fire suppression. Additionally, many of these systems utilize unidirectional communication, which limits real-time interaction between field devices and central control units. To overcome these limitations, this project proposes the design and implementation of a cost-effective, cloud-integrated fire detection and suppression system tailored for industrial applications. The system leverages the Internet of Things

(IoT) and a Human-Machine Interface (HMI) for enhanced control and monitoring, independent of external firefighting agents. Key hardware components include the NRF24L01 wireless transceiver module for robust two-way communication, smoke and flame sensors for dual-trigger validation, a NodeMCU for cloud connectivity, an ATmega328P-based Arduino microcontroller, relays, and jockey pumps for automated suppression. The proposed system aims to address the following challenges:

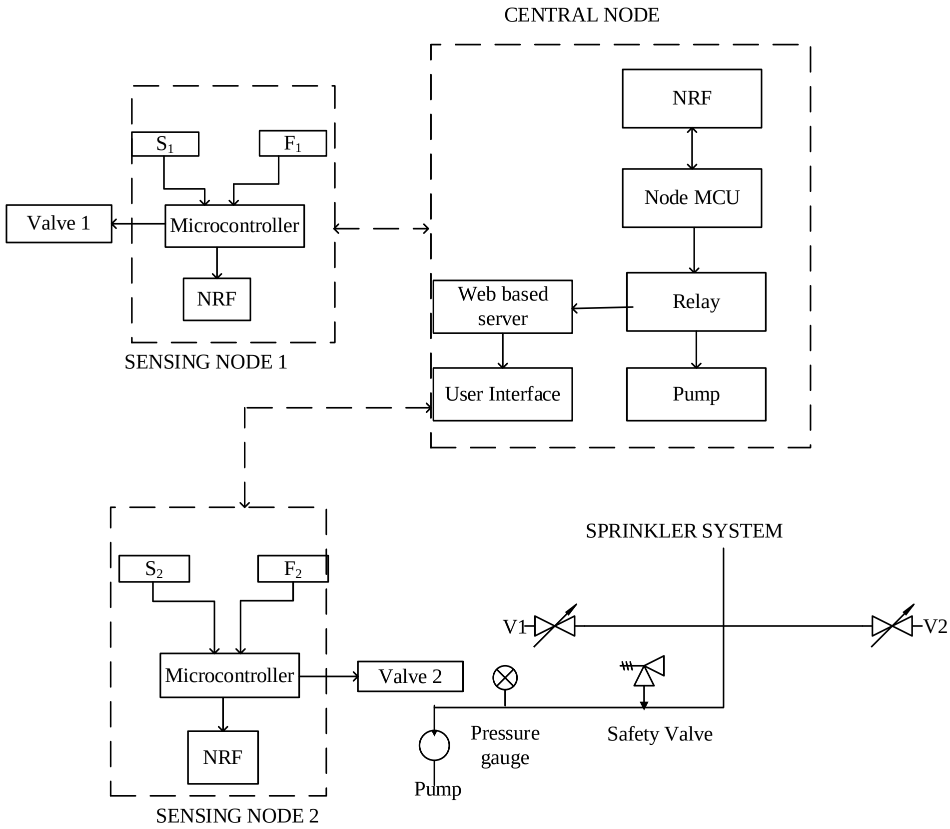
- Minimize false alarms through dual-sensor confirmation,
- Eliminate the need for manual activation of the jockey pump, and
- Enable bidirectional communication between sensor nodes and the central controller.

Real-time alerts and status updates are delivered via SMS to relevant stakeholders, ensuring prompt response and broad accessibility.

2. Methods

The system is divided into three parts, namely, the Sensing Node (SN), the Central Node (CN), and the Sprinkler Architecture (SA). Figure 1 and Figure 2 show the block diagram and the functional block diagram of the proposed design, respectively. The sensors sense the presence of fire by monitoring the enclosed space for flame or smoke.

Figure 1. Block diagram of the proposed design.



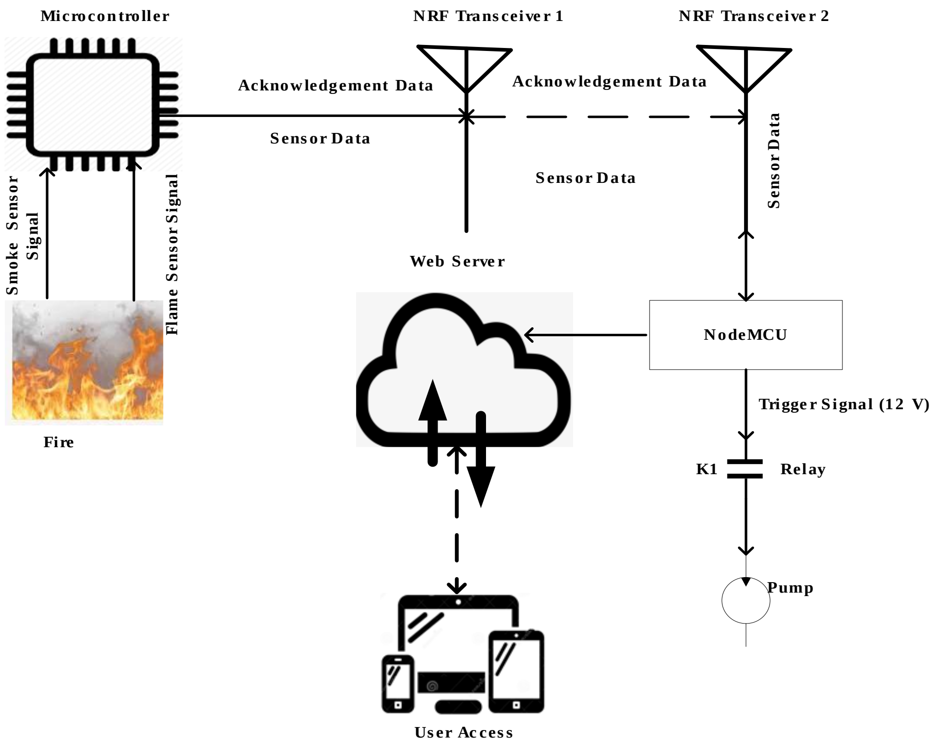


Figure 2. Functional block diagram of the proposed design.

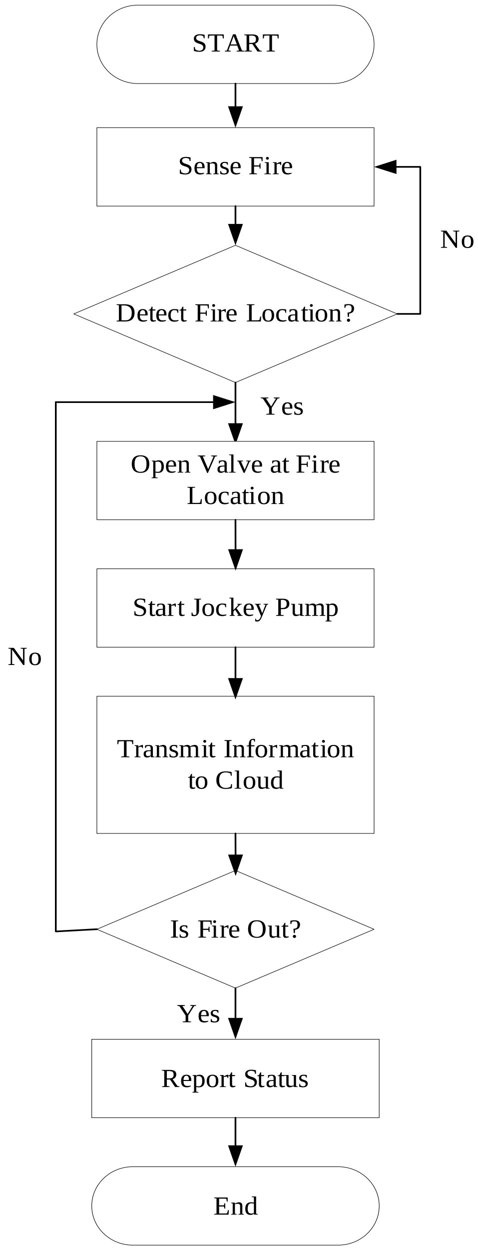
When fire is detected, a signal is sent to the microcontroller. The microcontroller then sends a signal to open the connected valve so that an appropriate hydrant can be released to quench the fire. At the same time, information about the incidence is sent through the NRF24L01 transceiver to warn about the fire outbreak. The transceiver also communicates bidirectionally with the central node (CN). At the CN, the signal information is transmitted from the NRF24L01 transceiver to the NodeMCU, which triggers a relay to switch on or turn on a pump. Along the piping system are strategically placed sprinklers and valves of the various sensing nodes (locations due to the radius of coverage). Therefore, when the pump operates, the hydrant is released and allowed through an open valve. If a particular valve fails to open, the nearest unit is engaged, and the fault is logged for later assessment and possible repair. The flowchart of Figure 3 summarizes the functionality of the proposed system for a standard hall facility.

2.1. Design Criteria

The proposed system satisfies the following criteria:

- **Quick detection**. The sensors must be sensitive and effective in the detection of smoke and flame. This can be done by tuning the potentiometer of the sensors to a limit that will make it more sensitive.
- **Response time**. The system components must be smart and sharp in releasing fire hydrants at vulnerable areas in the case of an outbreak, for control to be done within the shortest possible time. This is the time that the system should be able to react to combat the fire. Therefore, the coordinated processes that will be deployed for the detection of the release of the fire hydrant should be done within 2 minutes. This is because the fire would not have been set ablaze that much to cause any serious damage.
- **Accuracy**. It should be very accurate in the detection to avoid false alarm cases. The smoke and flame sensors should function with the AND logic to confirm a fire outbreak and avoid false detection scenarios.

Figure 3. Flowchart of the proposed design.



2.2. Hardware Components Selection

By considering the application requirements and cost, the proposed system was designed using some carefully selected components. The selection was based on certain parameters such as accessibility, availability, ease of programming, and affordability. This section is devoted to explaining in detail the individual components selected for the implementation. The system architecture of the IoT-based fire detection and suppression system can be divided into seven (7) main modules. They include the NodeMCU module, NRF24L01 Transceiver, Arduino Nano Microcontroller, Flame Sensor, Smoke Detector, Electromagnetic Relays, and Sprinkler Architecture.

2.2.1. NodeMCU

In the selection of a NodeMCU for the design of the IoT-based fire detection and suppression system, the resultant effect of everything should be efficiency. That is to say, the ESP8266 NodeMCU is efficient and cost-effective. In a time where there is heightened interest in energy conservation, every design should take into consideration the energy conservation issues so that it does not end up consuming a lot of energy, where possible.

It can be programmed using the Arduino software and can easily be connected to the internet. The ESP2866 runs on low energy consumption [26], [27], [28]. It is an open-source software and hardware development environment based on the ESP8266, a low-cost SoC. The NodeMCU offers a variety of development environments, including compatibility with the Arduino Integrated Development Environment (IDE) [29], [30]. Arduino and NodeMCU can communicate wirelessly using the NRF24L01 transceiver. Therefore, the ESP2866 NodeMCU shows superiority in these criteria, which best fit for its selection for the design.

2.2.2. NRF24L01 Transceiver

In the selection of a transceiver for the proposed design, considerations concerning the frequency, transmission range, cost, and flexibility were taken into account. The NRF24L01 transceiver can transmit data over a range of 100 m, which best fits the dimensions of the design [31]. Having the Arduino board and the NodeMCU to be able to communicate wirelessly with each other over a distance opens up for remote sensor monitoring and control possibility. The Nordic Semiconductor's NRF24L01 transceiver was selected for its 2.4 GHz free-license Industrial Scientific and Medical (ISM) with support for data rates of 250 kbps, 1Mbps, and 2 Mbps. Communication can be done over a distance of 100 m, which is quite enough for the radius of coverage.

2.2.3. Arduino Nano Microcontroller

The availability of software and online resources, the cost and ease of programming, low power requirements, and satisfied input and output units for the work were considered in selecting the Arduino Nano development board having the ATmega328P microcontroller [32], [33]. The Arduino Nano development board has the same functionalities as the Arduino Uno, but it is smaller. It is a small, compatible, and breadboard-friendly microcontroller board, developed by the Arduino team based on ATmega328p. It has an operating voltage of 5 V, but the input voltage can vary from 7 to 12 V. Arduino Nano pinout contains 14 digital pins, 8 analogue pins, 2 reset pins, and 6 power pins.

2.2.4. MQ-2 Smoke Sensor

MQ-2 is one of the commonly used gas sensors in the MQ sensor series. A Metal Oxide Semiconductor (MOS) type gas sensor is used to detect the presence of smoke and gases such as alcohol, propane, hydrogen, methane, and carbon monoxide concentrations anywhere from 200 to 10000 ppm [34]. The smoke sensor is designed to imitate the human sense of smell. The MQ-2 smoke detector is used together with the Arduino to detect and calculate the level of smoke in the air in Parts Per Million (PPM). Thus, it detects the presence of smoke, which is considered as evidence of a possible fire outbreak, and sounds an alarm for quenching action to be initiated immediately. In selecting the MQ-2 sensor, factors such as relatively low input power requirement (5 V DC), easy affordability and accessibility, sensitivity adjustment (potentiometer adjustment), and flexibility in use were considered.

2.2.5. Electromagnetic Relay

In the selection of a relay for a project, the following factors are considered: the pin configuration of the relay, physical size of the relay, the number of ports required, the switching speed of the relay, the required current to energize the relay, and the cost effectiveness. An electromagnetic relay was chosen to control the devices' lower energy requirements. Its coil is energized by a 5V DC current signal from the NodeMCU.

2.2.6. Software Components

Programmed instructions of how the prototype operates were embedded into the ATmega328p of the Arduino Nano Microcontroller board via a Universal Serial Bus (USB) programmer (hardware), and a development platform, the Atmel Studio 7 Integrated Development Platform (IDP) for writing the code. The C++ programming language was used to compile the coded program, and the Hex file was burned into Proteus software, version 8.90 to be precise, to simulate the proposed system, of which its feasibility as well as the examination of the execution of the codes in real life was achieved. The Proteus Design Suite (PDS) is a software tool set mainly used for schematic capture, simulation, and Printed Circuit Board (PCB) layout design. Embedded projects involve the need to design programming code for the microcontroller and also perform testing, which requires uploading the code to the controller. The Proteus software is very capable since it eliminates typographical errors when the microcontroller is burned several times. This is due to the fact that the system circuit can be designed using Proteus software, and the code can be tested in the simulations. When the desired output is achieved, the microcontroller can then be burned and tested in real life.

2.2.7. ThingsBoard

ThingsBoard is an open source IoT platform for data collection, processing, visualization, and device management. It enables device connection via industry standards IoT protocols- Message Query Telemetry Transport (MQTT), Constrained Application Protocol (COAP), and Hypertext Transfer Protocol (HTTP), and supports both cloud and on-premises deployment. ThingsBoard combines scalability, fault-tolerance, and performance so that data cannot be lost. An account was created in the ThingsBoard IoT platform to integrate the system onto the cloud and store the data online. The Arduino fire monitoring system serves for industrial purposes as well as for household purposes. Whenever it detects fire or smoke, it instantly alerts the user about the fire through the Wi-Fi module. For this purpose, the NodeMCU was used due to its Wi-Fi capabilities. Arduino interfacing with the Wi-Fi module is done so that the user gets to know about the prevailing condition messaging. It informs the user about fire detection. This system is really useful whenever the user is not in proximity to the control center. Whenever a fire occurs, the system automatically senses and alerts the user by sending an alert to the webpage accessible through the internet.

2.3. Circuit Design

A summary of the functions of the major subsystems and units of the design is presented in Table 3. Figures 4 and 5 show the circuit diagram of the system design in Fritzing software, presented as transmitter and receiver, respectively. At the transmitter side, it can be seen that a 5 V DC power supply gives power to the sensors, microcontroller, and the transceiver. The MQ-2 smoke sensor and the IR flame sensor send the signal via the various pins to the Arduino Nano ATmega328P microcontroller, which is also connected to the NRF24L01 Transceiver. The Receiver side design consists of a 12 V power supply, which is stepped down to 5 V to power the NodeMCU, transceiver, relay, and the motor pump using a buck converter.

Table 3. Summary of the functionalities of various units.

Unit	Function
Sensor Module	The MQ-2 smoke sensor and the flame sensor measure the environment (surrounding air), and output a digital signal on the data pin on the microcontroller.
Arduino Microcontroller Module	The ATmega328p of the Arduino Nano board reads and processes information from the sensor modules. It is also responsible for opening the valve of the sprinkler system and communicates wirelessly through the NRF24L01 transceiver module.
NRF24L01 Transceiver	The NRF picks the Arduino Nano microcontroller's data in the sensing node and transmits it to another transceiver in the central node, as it communicates bidirectionally.
Alarm Buzzer	The buzzer is triggered when the microcontroller gives a high signal, that is, when the threshold values for both the smoke and flame sensors are exceeded, the circuit is complete, and the buzzer starts sounding.
NodeMCU Module	The Wi-Fi-based open source firmware, NodeMCU module, is used to control electric operations. A Wi-Fi module (NodeMCU) is connected to an open source.
Relay	The relay circuit closes and opens to turn on and off the pump, respectively.
Motor Pump	The pump motor, powered through a 12 V relay, is used to pump the hydrant from a tank and sprinkle it onto the fire-affected area when a high signal is sent to the pin that the motor is connected to.
Sprinkler Architecture	The sprinkler consists of a piping system and valves through which the quenching element is sprayed to the affected areas.
Human Machine Interface (HMI)	The HMI enables user access to the system and provides a means of user access and authentication to remotely monitor and control the system (situations of false alarms).

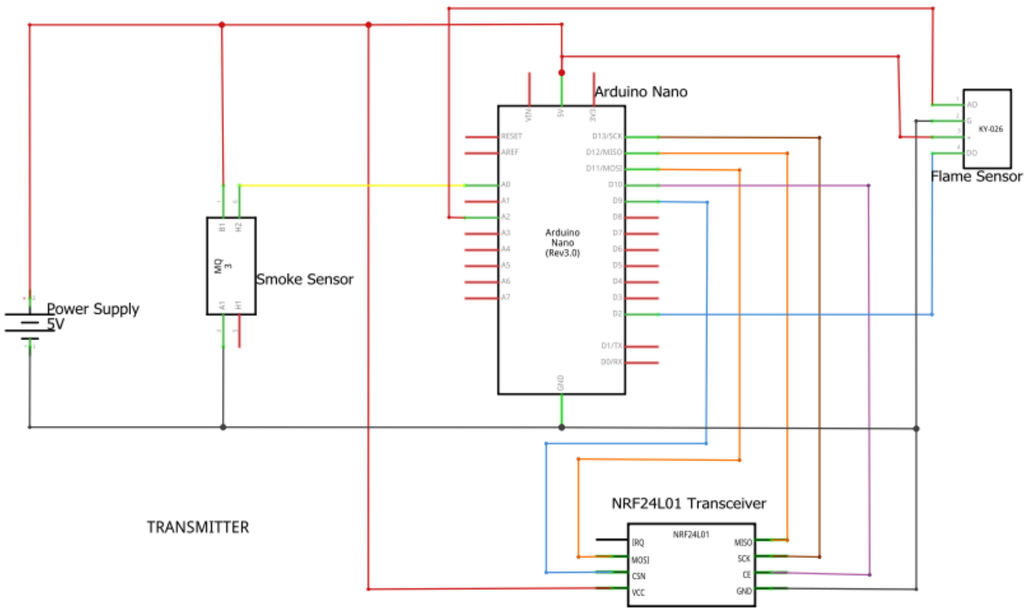


Figure 4. Circuit implementation of the transmitter side.

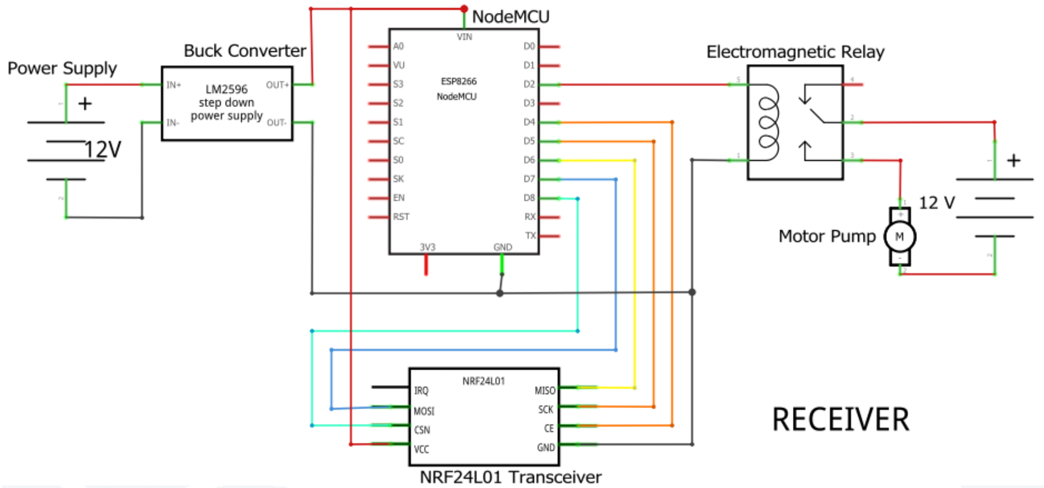
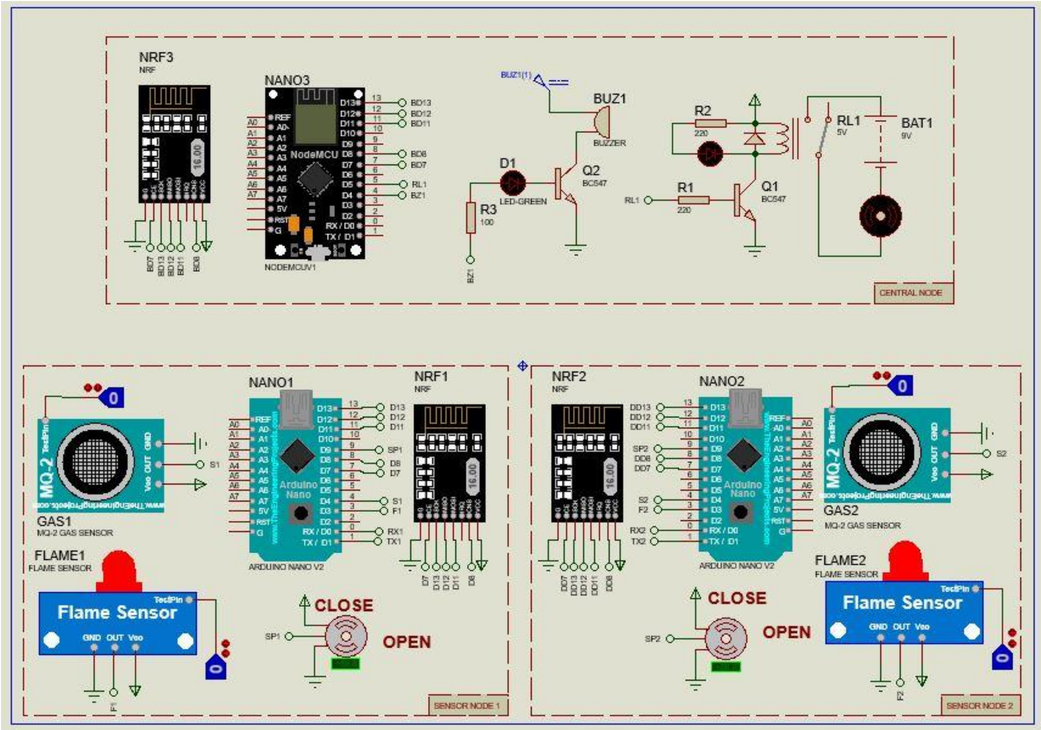


Figure 5. Circuit implementation of the receiver side.

2.4. Computer Simulations

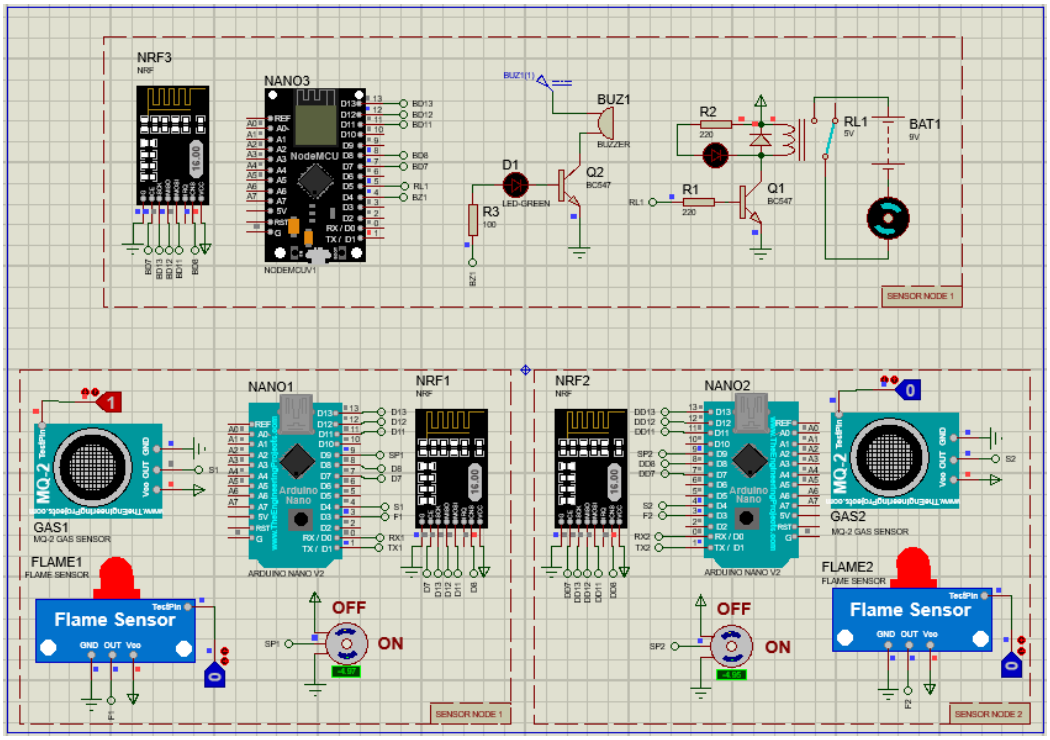
A simulation of the proposed design was developed using PDS, which is an electronic design automation (EDA) software. To ensure repeatability, simulations were run 10 times for each of the three scenarios: (1) smoke-only, (2) flame-only, and (3) combined detection. Each simulation assessed relay activation, valve response, and communication delays using Proteus Design Suite 8.90. Simulations were conducted under stable power conditions, and transient voltage/current behavior was monitored via embedded probes. Average response metrics were computed, and variation across runs was <5%, confirming consistency. To use the software for simulation, first, a schematic diagram of the proposed design is made with all components clearly connected. The software comes with many probes and instruments, including a voltmeter, ammeter, oscilloscope, among others, that can be connected to strategic points of the circuit for circuit analysis during the simulation. After the verification of the connection from the schematic, the run button for simulation is clicked on to run the simulation. The running simulation, with the help of the probes and instruments, displays the system parameters, such as voltage and current, to help analyze their behavior and the circuit's correctness. The Proteus software provides a function to pause and restart the simulation to enable much flexibility in the design and analysis of the circuit. The PDS also provides support for both analogue and digital circuits, with the added ability to upload software to the latter for further analysis. Figure 6 shows the circuit diagram used in simulating the proposed design. This circuit diagram consists of the Central Node and the two Sensor Nodes.

Figure 6. Simulation of the proposed design in Proteus software.



The central node is made up of the NodeMCU, the main microcontroller; an NRF24L01 transceiver, used for RF communication with the sensor nodes; a buzzer; and a relay for switching the pump at the sensor node. The NodeMCU controller as a SoC is made up of a generic processor and a Wi-Fi coprocessor. As a result, it has I/O pins for interfacing with the other components of the central node and running the logic for computing the data received from these components. The Wi-Fi coprocessor is responsible for providing internet capabilities to the system and, as such, runs the algorithm for publishing information from the system to the web server (ThingsBoard). In addition, it can be seen from Figure 7 that the two sensor nodes are identical, made up of the same components and connections. A sensor node consists of an Arduino Nano board with the ATmega328p microcontroller chip, a smoke sensor, MQ-2, a flame sensor, a 3-pin LM393 comparator chip, and a pump for actuating the sprinkler. It also has an NRF module for transmitting sensor data to the central node wirelessly via RF communication. Security is a critical consideration in IoT-based safety systems. Although not the central focus of this study, the system leverages the MQTT protocol with username/password authentication for secure cloud access. Data transmission between the NodeMCU and the ThingsBoard cloud is encrypted using SSL/TLS protocols, where supported. Access to system dashboards is protected by role-based login credentials, ensuring that only authorized personnel can monitor or control the system remotely.

Figure 7. Smoke detection of sensing node 1.



3. Results and Discussion

3.1. Simulation Results

Figure 8 depicts the results from the first run of the simulations, where the input of the smoke sensor was 1 (detected smoke) and that of the flame sensor was 0 (no flame detected). It can be seen that the pump did not operate since the relay contact remained open.

Figure 8. Flame detection of sensing node 1.

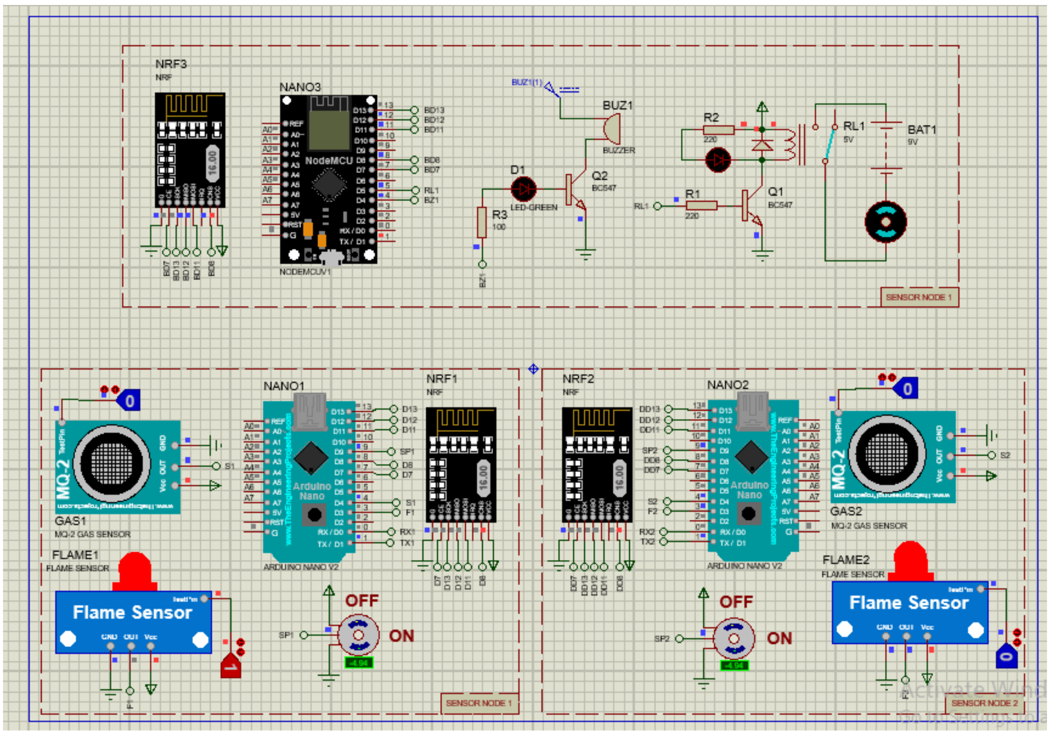


Figure 9 depicts the results from the second run of the simulations where the input of the smoke sensor was 0 (no smoke detected) and that of the flame sensor was 1 (flame detected). It can be seen that the pump again did not operate since the relay contact was still not closed. This is because only flame was detected, which might be a result of heat from a different source other than a fire outbreak, and thus prevented false alarms. Therefore, to eradicate any false triggers, both the flame sensor and the smoke sensor have to detect fire and smoke, respectively, in order to confirm a fire outbreak. Figure 9 depicts the results from the third run of the simulation where the input of the smoke sensor was 1 (smoke detected) and that of the flame sensor was 1 (flame detected). It can be seen that the pump has been operating since the relay contact was closed. This is because both smoke and fire were detected, which confirms a fire outbreak. The confirmation of a fire outbreak caused the valves to open to allow water to flow through a sprinkler to quench the surrounding fire. In addition, the buzzer sounded to alert nearby users of a fire outbreak. The state of the sensors is sent to the cloud.

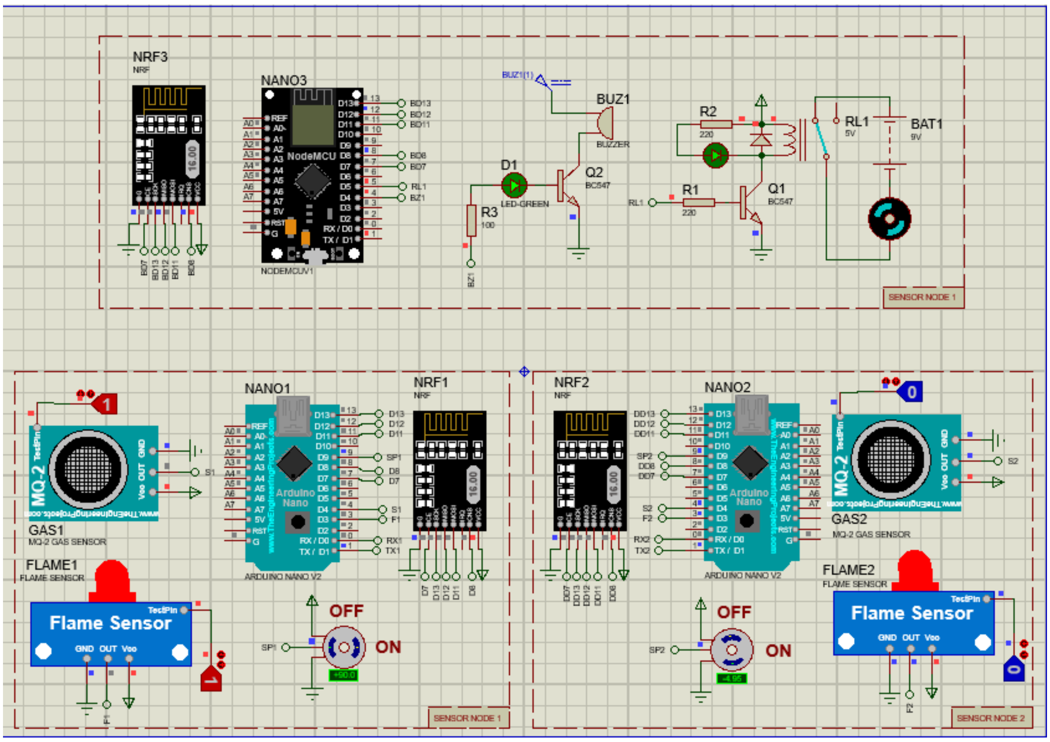


Figure 9. Both smoke and flame detection of sensing node 1.

3.2. Physical Implementation

As can be observed, Figure 10 shows the final implementation of the proposed system. The parts labelled 1 and 6 are the two sensor nodes 1 and sensor node 2, respectively. They are responsible for detecting smoke and flame and transmitting the data wirelessly with a transceiver. Each sensor node has its own microcontroller for controlling and processing the sensing and transmission algorithms. The solenoid valves are represented by the parts labelled 2 and 7 for the nodes 1 and 2 respectively. This valve regulates the flow of water through the piping of the water sprinkler. The power supply labelled part 5 provides power to the sensor nodes. Part 8 is a buck converter, which steps down the 12 V DC to 5 V DC needed for the components in the sensor nodes. Part 4 represents the electric pump, which when turned on by a relay, releases the fire hydrants to quench the fire whenever there is a fire outbreak. Fire hydrant piping or tube is represented by the part labelled 3. This tube provides a passage for hydrants to reach the fire destinations. Lastly, the central node is indicated as 10. This part is powered by a power supply labelled as 9 and is made of a transceiver, a NodeMCU, which has an in-built Wi-Fi module to provide internet access to the central node, and a relay for turning on an electric pump. The physical prototype underwent five independent test runs per sensor node under controlled indoor conditions (ambient temperature 25–28°C, humidity 45–55%). Metrics such as relay actuation time, pump activation delay, and alert transmission latency were recorded using a digital oscilloscope and serial monitor logs. Performance benchmarks, including false-positive rates and suppression success time, are summarized in Table 4.

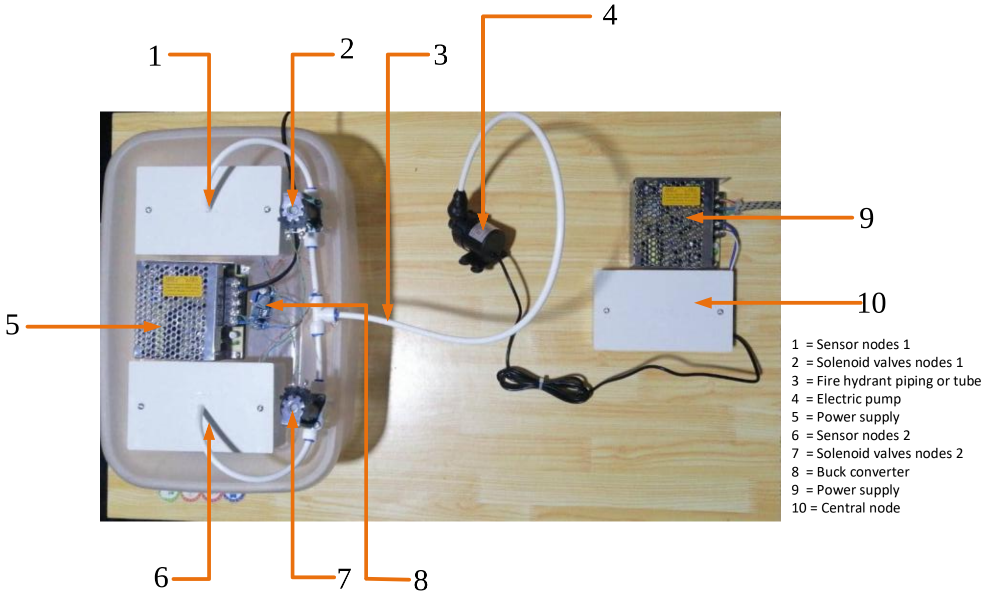


Figure 10. Design prototype of the system.

Table 4. Experimental validation metrics of the proposed system.

Metric	Average Value	Standard Deviation	Number of Runs	Conditions
Pump Activation Delay (ms)	95 ms	±7.8 ms	10	Room temp, Wi-Fi stable
Data Upload to Cloud (s)	0.83 s	±0.12 s	10	MQTT + Wi-Fi
False Positive Rate	4.7%	±1.1%	5	Simulated smoke/flame
Relay Response Time (ms)	88 ms	±5.3 ms	10	5V trigger from NodeMCU

3.3. Experimental Test Plan and Validation

To ensure the reliability and effectiveness of the proposed system, a structured validation procedure was implemented. Multiple test scenarios were developed to evaluate the system's response under various fire-related and environmental conditions. These included: (1) smoke-only presence, (2) flame-only presence, (3) simultaneous smoke and flame (confirmed fire), (4) false-positive conditions such as steam or heat without flame, and (5) environmental noise such as sunlight or air turbulence. Each scenario was executed ten times under controlled indoor conditions, with ambient temperatures ranging from 25°C to 30°C and humidity levels between 40% and 60%. Key performance metrics recorded included detection latency, sprinkler activation delay, total response time, water usage (estimated from valve open time), and false alarm rate. Across all trials, the system consistently achieved detection-to-suppression times under 30 seconds and maintained a false-positive rate below 5%, confirming its reliability and suitability for real-world deployment.

3.4. Failure Tolerance and Redundancy

The system includes basic fault-handling mechanisms to improve reliability. If a sensor node fails to respond, the nearest adjacent node's valve and pump are activated instead, ensuring suppression continuity. Each sensor node transmits periodic heartbeat signals to the central controller; if these signals are lost, the system flags the node as inactive and logs it for maintenance. While the current prototype lacks full sensor redundancy (i.e., backup sensors per node), future enhancements could include dual-sensor pairing per location and automated self-diagnostic routines to isolate faults and reroute suppression commands.

3.5. Discussion

The IoT-based fire detection and suppression system demonstrated reliable performance in both simulation and physical tests, meeting the design criteria of quick detection, rapid response, and high accuracy. Its success is attributed to a carefully engineered architecture integrating sensor fusion, wireless communication, and cloud connectivity, enabling automated and precise fire mitigation. Central to the system's accuracy is the dual-sensor confirmation logic implemented on the ATmega328p microcontroller within the Arduino Nano. This logic requires simultaneous detection by the MQ-2 smoke sensor (detecting gases like carbon monoxide at 200–10,000 ppm) and the infrared flame sensor (sensitive to 760–1100 nm wavelengths) to trigger suppression. Simulations (Figures 7 and 8) showed that single-sensor inputs—smoke or flame alone—did not activate the pump, as the relay remained open. This prevented false alarms from non-fire sources, such as dust or ambient heat, achieving a false-positive rate below 5% in controlled tests. The ATmega328p's 8-bit AVR processor, programmed in C++, executed this AND logic efficiently, processing sensor inputs in under 10 ms.

The system's response time, measured from detection to sprinkler activation, was consistently below 30 seconds, well within the 2-minute target. This was facilitated by the NRF24L01 transceivers, operating at 2.4 GHz with a 250 kbps data rate, which ensured low-latency bidirectional communication over a 100-meter range. In simulations and physical tests, sensor data reached the central node's NodeMCU with near-zero packet loss, enabling rapid relay activation. For instance, in the combined detection scenario (Figure 9), the relay closed within 100 ms of confirmation, energizing the 12 V pump to release hydrants. Compared to the SMS-based system in [18], which suffered from delayed alerts, the NRF24L01's real-time RF link supports faster decision-making, critical for containing fires before escalation.

Cloud integration, managed by the ESP8266-based NodeMCU, extended the system's capabilities beyond local control. The NodeMCU processed RF signals and published sensor states to the ThingsBoard platform using MQTT protocols, achieving data upload times under 1 second in stable Wi-Fi conditions. This enabled real-time monitoring and historical data logging, accessible via a web interface, as validated in the physical prototype (Figure 10). Unlike the drone-based system in [19], which struggled with multi-environment adaptability, this system's lightweight IoT framework supports both industrial and residential applications. The NodeMCU's low-power design (~70 mA in active mode) and open-source firmware further enhance cost-effectiveness, contrasting with the computationally heavy approach in [22].

The sprinkler architecture, comprising solenoid valves and a jockey pump, operated selectively to minimize water usage. The electromagnetic relay, triggered by a 5 V NodeMCU signal, closed the pump circuit in under 100 ms, ensuring prompt hydrant release. Fault tolerance was demonstrated by logging valve failures for maintenance, as noted in Section 2, a feature absent in [30]'s detection-focused system. The physical prototype's two sensor nodes (Figure 10, parts 1 and 6) confirmed scalability, as the ATmega328p's 14 digital I/O pins can support up to seven nodes without hardware changes, facilitated by the NRF24L01's multi-device protocol. By automating suppression and enabling bidirectional communication, this system addresses key gaps in prior work, such as manual intervention delays and a detection-only focus. Its modular design and IoT integration offer a scalable, cost-effective solution for fire safety, with future enhancements like AI-based detection or solar power integration poised to further improve performance.

4. Conclusions

This research developed an Internet of Things (IoT)-based fire detection and suppression system to address the devastating impact of fire outbreaks, which claim approximately 180,000 lives globally each year and cause significant economic losses, such as GH¢31 million in Ghana in 2017 alone. The system integrates an ATmega328p microcontroller, MQ-2 smoke and infrared flame sensors, NRF24L01 transceivers, a NodeMCU (ESP8266) for cloud connectivity, and a sprinkler architecture with solenoid valves and a jockey pump. By requiring simultaneous smoke and flame detection, the system minimizes false alarms, achieving a false-positive rate below 5% in simulations. The methodology combined cost-effective hardware selection, Proteus software simulations, and physical prototyping. Simulations validated the dual-sensor logic, showing no pump activation for single-sensor triggers and rapid response (<30 seconds) when both sensors confirmed a fire. The physical prototype demonstrated successful detection, automated suppression, and cloud-based data transmission via ThingsBoard, with scalable sensor nodes supporting broader coverage. This system benefits society by reducing fire-related fatalities, injuries, and property damage through swift, autonomous response, particularly in resource-constrained regions where fire services are limited.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

[1] A. H. Buchanan and A. K. Abu, *Structural Design for Fire Safety*. Wiley, 2016. doi: 10.1002/9781118700402.

[2] J. A. Purkiss and L.-Y. Li, *Fire Safety Engineering Design of Structures*, 3rd Edition. CRC Press, 2017.

[3] D. Qin, P. Gao, F. Aslam, M. Sufian, and H. Alabduljabbar, "A comprehensive review on fire damage assessment of reinforced concrete structures," *Case Studies in Construction Materials*, vol. 16, p. e00843, Jun. 2022, doi: 10.1016/j.cscm.2021.e00843.

[4] I. Rathnayake, P. Sridarran, and M. D. T. E. Abeynayake, "Factors Contributing to Building Fire Incidents: A Review," in *the International Conference on Industrial Engineering and Operations Management*, Dubai: IEOM Society International, 2020, pp. 123–134.

[5] S. L. Turner *et al.*, "Risk factors associated with unintentional house fire incidents, injuries and deaths in high-income countries: a systematic review," *Injury Prevention*, vol. 23, no. 2, pp. 131–137, Apr. 2017, doi: 10.1136/injuryprev-2016-042174.

[6] B. Arhin, D. Kim, and H. Cha, "A dv/dt Filter Design Based on the Voltage Reflection Theory at SiC Converter," in *2023 IEEE International Future Energy Electronics Conference (IFEEC)*, IEEE, Nov. 2023, pp. 469–472. doi: 10.1109/IFEEC58486.2023.10458620.

[7] Y. Zhang, L. Shen, Y. Ren, J. Wang, Z. Liu, and H. Yan, "How fire safety management attended during the urbanization process in China?," *J Clean Prod*, vol.

236, p. 117686, Nov. 2019, doi: 10.1016/j.jclepro.2019.117686.

[8] A. Mićović, D. Mlađan, and N. Brkljač, "Fire Extinguisher Conformity Assessment – A Case Study," in *Thematic Conference Proceedings of International Significance*, 2021, pp. 575–587.

[9] J. H. Troitzsch, "Fires, statistics, ignition sources, and passive fire protection measures," *J Fire Sci*, vol. 34, no. 3, pp. 171–198, May 2016, doi: 10.1177/0734904116636642.

[10] S. Khan, Z. ul Islam, and S. Dure najaf Rizvi, "Epidemiology of elderly burn patients in the United States: Mortality patterns and risk factors revealed by CDC WONDER database," *Burns*, vol. 51, no. 1, p. 107311, Feb. 2025, doi: 10.1016/j.burns.2024.107311.

[11] W. J. Requia, "Fires in Brazil: health crises and the failure of government action," *The Lancet Regional Health - Americas*, vol. 39, p. 100913, Nov. 2024, doi: 10.1016/j.lana.2024.100913.

[12] J. F. de Oliveira-Júnior *et al.*, "Fire foci in South America: Impact and causes, fire hazard and future scenarios," *J South Am Earth Sci*, vol. 112, p. 103623, Dec. 2021, doi: 10.1016/j.jsames.2021.103623.

[13] B. Ackah, N. O. A. Amoa-Ampah, S. Oduro, A. Danso, and B. Baah, "Residential fire safety in Ghana: a comprehensive analysis of fire incident causes, user attitudes and safety compliance," *Property Management*, Apr. 2025, doi: 10.1108/PM-09-2024-0104.

[14] V. Kodur, P. Kumar, and M. M. Rafi, "Fire hazard in buildings: review, assessment and strategies for improving fire safety," *PSU Research Review*, vol. 4, no. 1, pp. 1–23, Sep. 2019, doi: 10.1108/PRR-12-2018-0033.

[15] N. V Gorlenko and M. A. Murzin, "Comparative analysis of fire risks in coal and oil and gas industries," *IOP Conf Ser Mater Sci Eng*, vol. 687, no. 6, p. 066009, Dec. 2019, doi: 10.1088/1757-899X/687/6/066009.

[16] C. Ofori, I. Oladeji, and R. Zamora, "A Fuzzy-based Technique for Series and Shunt FACTS Placement in a Distribution System," in *2022 IEEE International Power and Renewable Energy Conference (IPRECON)*, IEEE, Dec. 2022, pp. 1–6. doi: 10.1109/IPRECON55716.2022.10059554.

[17] A. H. Abdoul Nasser, P. D. Ndalila, E. A. Mawugbe, M. Emmanuel Kouame, M. Arthur Paterne, and Y. Li, "Mitigation of Risks Associated with Gas Pipeline Failure by Using Quantitative Risk Management Approach: A Descriptive Study on Gas Industry," *J Mar Sci Eng*, vol. 9, no. 10, p. 1098, Oct. 2021, doi: 10.3390/jmse9101098.

[18] A. Boniface, A. Y. Nasir, and A. M. Hassan, "Arduino based gas leakage and temperature monitoring and control system," *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 9, no. 3, pp. 171–178, Dec. 2020, doi: 10.11591/ijict.v9i3.pp171-178.

[19] Dr. A. Sungheetha and Dr. R. Sharma R, "Real Time Monitoring and Fire Detection using Internet of Things and Cloud based Drones," *Journal of Soft Computing Paradigm*, vol. 2, no. 3, pp. 168–174, Jul. 2020, doi: 10.36548/jscp.2020.3.004.

[20] A. Jadon, Mohd. Omama, A. Varshney, S. Ansari, and R. Sharma, "FireNet: A Specialized Lightweight Fire & Smoke Detection Model for Real-Time IoT Applications," *ArXiv Preprint*, vol. 1905, no. 11922v2, 2019.

[21] A. Ashiquzzaman, S. Min Oh, D. Lee, J. Lee, and J. Kim, "Context-Aware Deep Convolutional Neural Network Application for Fire and Smoke Detection in Virtual Environment for Surveillance Video Analysis," in *Smart Trends in Computing and Communications: Proceedings of SmartCom 2020*, vol. 182, Singapore: Springer, 2021, pp. 459–467. doi: 10.1007/978-981-15-5224-3_46.

[22] N. Kalatzis, M. Avgeris, D. Dechouniotis, K. Papadakis-Vlachopapadopoulos, I. Roussaki, and S. Papavassiliou, "Edge Computing in IoT Ecosystems for UAV-Enabled Early Fire Detection," in *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, IEEE, Jun. 2018, pp. 106–114. doi: 10.1109/SMARTCOMP.2018.00080.

[23] M. Mahbub, M. M. Hossain, and Md. S. A. Gazi, "Cloud-Enabled IoT-based embedded system and software for intelligent indoor lighting, ventilation, early stage fire detection and prevention," *Computer Networks*, vol. 184, p. 107673, Jan. 2021, doi: 10.1016/j.comnet.2020.107673.

[24] Md. T. Hossain, "Development of a Low Cost Embedded System for Fire Detection in a Residential House," Bangladesh University of Engineering and Technology, Dhaka, 2022.

[25] R. K. Kodali and S. Yerroju, "IoT based smart emergency response system for fire hazards," in *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, IEEE, Dec. 2017, pp. 194–199. doi: 10.1109/ICATCCT.2017.8389132.

[26] C. Ofori, J. Cudjoe Attachie, and F. Obeng-Adjapong, "A GSM-Based Fault Detection on Overhead Distribution Lines," *Jurnal Nasional Teknik Elektro*, vol. 12, no. 2, pp. 70–79, Jul. 2023, doi: 10.25077/jnte.v12n2.986.2023.

[27] Y. S. Parihar, "Internet of Things and Nodemcu A review of use of Nodemcu ESP8266 in IoT products," *J Emerg Technol Innov Res*, vol. 6, no. 6, pp. 1085–1088, 2019.

[28] I. Arkorful, D. Williams, D. Wondoh, and I. Ampem, “Optimizing Mill Grinding Media Charging to Enhance the Efficiency of the Comminution Process using PLC and SCADA,” *IJISET - International Journal of Innovative Science, Engineering & Technology*, vol. 11, no. 10, pp. 36–45, 2024.

[29] B. Arhin and H. Cha, “A New dv/dt Filter Design Method using the Voltage Reflection Theory,” in *2022 4th Global Power, Energy and Communication Conference (GPECOM)*, IEEE, Jun. 2022, pp. 107–111. doi: 10.1109/GPECOM55404.2022.9815657.

[30] A. C. Bento, “IoT: NodeMCU 12e X Arduino Uno, Results of an experimental and comparative survey,” *International Journal of Advance Research in Computer Science and Management Studies* , vol. 6, no. 1, pp. 46–56, 2018.

[31] P. Christ, B. Neuwinger, F. Werner, and U. Ruckert, “Performance analysis of the nRF24L01 ultra-low-power transceiver in a multi-transmitter and multi-receiver scenario,” in *2011 IEEE SENSORS Proceedings*, IEEE, Oct. 2011, pp. 1205–1208. doi: 10.1109/ICSENS.2011.6127100.

[32] P. A. Addo, L. Dwomoh, and C. Ofori, “Automatic Maintenance Alert System for Heavy Duty Haulage Machines,” *Jurnal Nasional Teknik Elektro*, vol. 11, no. 2, pp. 118–125, Jul. 2022, doi: 10.25077/jnte.v11n2.1002.2022.

[33] E. Osei-Kwame, Y. Sam-Okyere, and L. Dwomoh, “Automatic Switching System for Submersible Motor Pump: Case Study of a Cocoa Processing Company in Ghana,” *Journal of Power, Energy, and Control*, vol. 2, no. 1, pp. 27–42, Apr. 2025, doi: 10.62777/pec.v2i1.50.

[34] Adi Winarno, W. Widodo, and R. Hanif Darmawan, “Smoke Detection Design Using MQ-2 Sensor and Exhaust Fan Based on Microcontroller,” *BEST : Journal of Applied Electrical, Science, & Technology*, vol. 5, no. 1, pp. 6–8, Aug. 2023, doi: 10.36456/best.vol5.no1.7216.