Fire Detection and Prevention System in Residential Kitchens Using IoT for Early Warning

Kristhian A. Silva[1], Victor R. M. Cavalcante[1], Kaciana S. Oliveira[1], William B. B. dos Santos[1], Celso B. Carvalho[1], Waldir S.S. Júnior[1], Andrey R.R. Bessa[1], Diego A. Amoedo[1], Edma V.C.U. Mattos[1]
[1]Federal University of Amazonas and Center for R&D in Elec. and Inf. Tech. (UFAM/CETELI), AM-Brazil
Emails: {kristhiansilva, victorcavalcante, kacianaoliveira, williamsantos, celso, waldir, andreybessa, diegoamoedo, edmamattos}@super.ufam.edu.br

***Abstract*— This paper aims to develop a system that detects potential fires in residential kitchens. The system collects real-time data on changes in temperature, CO2, and cooking gas leaks using sensors and transmits this information to residents through an Android application. The machine learning model embedded in the system achieved an accuracy of 99.5% during training and 99.4% during testing, demonstrating high reliability in detecting anomalies. The system developed in this study will allow for early intervention and fire prevention, significantly reducing potential losses.**

***Keywords*— Fire detection, Fire prevention, Residential kitchens, Internet of Things (IoT), Sensors, Machine learning, Automated system, Real-time monitoring, Android application**

I. Introduction

The kitchen plays a central role in homes, but it also represents an environment with a high risk of accidents using gas stoves and other cooking equipment [1]. Studies highlight that gas leaks, sudden temperature increases, and fires associated with the improper use of utensils are frequent causes of residential fires, with smoke being the leading cause of deaths [1], [2]. Despite the availability of automated fire detection systems using IoT, gaps in accuracy and responsiveness still need to be addressed. For example, existing works mainly focus on the integration of basic sensors, such as the MQ2 and DHT11 sensors, but lack advanced predictive analytics or real-time detection with reliable notifications [3], [4], [5].

Early and effective detection of fire outbreaks is important in preventing fires and protecting lives and property. Since fires in crowded places, such as buildings, tend to increase the number of victims and property damage, it is necessary to take appropriate measures. To this end, there has been a growing effort in developing automated systems using the Internet of Things, capable of monitoring and managing possible occurrences, and there are already several studies conducted in practice [3] [2].

This paper aims to develop an automated system to detect potential fires in residential kitchens early. The system collects real-time data on changes in temperature, CO2, and cooking gas leaks through BMP280: temperature (°C), MQ2: smoke and gas (ppm), and CCS811: CO2 (ppm) sensors installed in the kitchen, sends this information through an ESP32 TTGO T-Beam microcontroller, and if there is any sign of abnormality during monitoring, the system transmits this alert information to residents through an Android application. The system developed in this study can effectively reduce losses resulting from a kitchen fire.

II. Related Work

Automated fire detection systems using IoT have gained prominence in recent years, offering innovative solutions to monitor and prevent disasters in kitchens. Previous works present diverse approaches; however, there is still room for advances in accuracy and integration with predictive technologies. This section presents works related to our research, and Tab. I summarizes the primary sensors, technologies, and limitations of analyzed works, comparing them with the proposed system.

Paper [6] presents a solution for various kitchen problems such as gas leakage, sudden fire, excessive smoke, and sudden temperature rise. The status of these parameters is displayed to the user in real-time, interconnecting the various types of sensors with the Node MCU and integrated Wi-Fi technology that will monitor all the kitchen parameters and can be connected to a mobile application.

Paper [3] presents a smart kitchen project with an automation and monitoring system that allows the user to control their home through their cell phone. For this purpose, the ESP32, DHT11 Sensor, X8 5V Relay, and MQ-135 gas sensors are used, which make up an intelligent kitchen capable of controlling temperature, managing humidity, and detecting gas leaks. The hardware was integrated and programmed using an Arduino, and an Android application was developed, allowing the user to access and control it.

The authors in the paper [1] developed an intelligent fire prevention system for kitchens consisting of sensors capable of detecting flames, high temperatures, or gas leaks, an alarm that emits sound and flashes to alert people, a message notification system and an Internet-enabled camera installed in the kitchen, allowing monitoring via cell phones. The proposed system also includes flame, temperature, and gas detection functions and automatic door unlocking, which allows the management team and rescue personnel to enter the house to fight the fire immediately.

Paper [4] highlights several Internet of Things features and their relevance in innovative kitchens. The system was developed using MQ2 (Gas) sensors, pressure sensors, DHT11 sensors, and IR sensors coupled to an Arduino Uno board. A serial monitor displays information on humidity, temperature, gas level, air quality, sound or alarm status, and human presence. A notification is sent via the mobile application if any accident occurs in the kitchen.

Paper [5] uses Node MCUs combined with gas sensors, temperature sensors, MQ3 sensors, alarm systems, exhaust fans, load cells, and integrated mobile applications. The gas sensor will transmit a warning message to the user upon detecting a gas leak, and as a result of this alert, the stove knob will automatically turn off. When a gas leak is detected, the gas system, which uses an IoT-based sensor, notifies the user via SMS and, as a safety measure, turns off the electricity while sounding an alarm. The temperature of the intelligent kitchen is monitored, and if it exceeds a specific limit, the exhaust fan is activated. The system also monitors all the software functionalities, and based on the results, the system can be used in private homes and public buildings.

Paper [7] presents the prototype of a system that quickly helps homeowners and firefighters detect fires and gas leaks. This residential fire detection system measures a room's ambient temperature and gas levels. Then, the output of this system sends short message information and alarms. The results revealed that the prototype room with scales of 1:25, 1:50, and 1:75, which uses temperature and gas sensors, can work as intended. In 10 tests, the system works according to the designed plan, which means the system can interpret a room's temperature and gas leak, and the system will send a short message and sound the alarm.

Paper [8] aims to implement an intelligent fire detection system that not only detects fire using onboard sensors but also alerts property owners, emergency services, and local police stations. The proposed system employs different onboard detectors, such as heat, smoke, and flame. The signals from these detectors pass through the system's algorithm to check the fire potential and then transmit the predicted result to multiple parties using the GSM modem associated with the system. The experimental results showed the model's superiority in accessibility, effectiveness, and responsiveness, utilizing the Ubidots platform and making data exchange faster and more reliable.

Paper [9] builds an automated kitchen monitoring system using the Internet of Things (IoT) and microcontroller-based sensors. The computerized kitchen monitoring system has a sensor unit (for monitoring kitchen parameters) that is interfaced with a microcontroller development board (Arduino Uno), and the transmission of these parameter values over the Internet is done by NodeMCU 1.0. The system has a mobile application for monitoring kitchen parameters like temperature (in degrees Celsius), humidity percentage, potential LPG gas leakage, accidental kitchen fires, and weighing scales for groceries. The mobile application was developed using an online tool from the inventor of MIT APP, and the database that temporarily stores kitchen parameters is Google's Firebase Database. The mobile application also has the feature of controlling two electronic appliances (connected to a two-channel relay) from anywhere in the world.

Paper [10] presents a novel Internet of Things (IoT)-based fire prevention system that uses the Ethereum distributed ledger (blockchain) to build a verifiable record of fire risk events that operate offline and online and requires no additional electrical installation. The device plugs into the stove's power outlet, and when smoke is detected, the device cuts off electricity to the stove and informs the owner via a smartphone notification. The system uses a Google Firebase-based web service to establish the connection between the IoT device and the smartphone. Applications are available for Android and iOS devices that allow the owner to control and turn off their stove and receive alerts about the state of the sensors.

As will be presented in the following sections, our proposed system advances the analyzed works by incorporating machine learning for highly accurate predictive detection of anomalies. In addition, integrating more accurate sensors, such as BMP280 and CCS811, increases the reliability of identifying changes in the kitchen environment.

III. Proposed System

A. Architecture of the proposed system

In Fig. 1, we present the architecture of the proposed system. The architecture is composed of three stages: (i) Data Capture, (ii) Data Processing, and (iii) Data Display.

1) Data Capture: This step consists of acquiring data using sensors capable of detecting temperature (BMP280 sensor: temperature (°C) with range: -40 to 85 °C), smoke and gas (MQ2 sensor: smoke and gas (ppm) with range: 300 to 10.000 ppm) and CO2 (CCS811 sensor: CO2 (ppm) with range: 400 to 32768 ppm). The data collection environment was set up in a residential kitchen, as shown in Fig. 2, where the data collection device was placed at the height of an air purifier, 75 to 80 cm above the stove. The sensors are connected to an ESP32 microcontroller via the I2C protocol, which receives the collected data and sends this information to the Firebase database via Wi-Fi.

2) Data Processing: The Firebase database receives timestamp information (date and time of collection) and data provided by sensors. This data is accessed in Firebase through Google Colab. Based on the access to the collected data, training is performed, which will classify the data into two classes: anomaly (presence of fire) and non-anomaly (absence of fire).

A threshold is defined by mean and standard deviation to detect the presence or absence of an anomaly. For this project, the mean plus 25% of the standard deviation was calculated to calculate the threshold. When the measurements exceed this value, an anomaly is considered to have occurred.

In training, two hidden layers of a sequential neural network were used, and the data was divided as follows: 70% of the data was used for training, 30% for testing and validation, and 15% for testing and 15% for validation. The model ported to the ESP32 was trained with 50 epochs. The accuracy, that is, how much the model could correctly predict the values, was 0.995 for training and 0.994 for testing.

3) Data Display: The user application displays sensor information in the background, with data being read every second. When the value collected by the sensors is above the threshold and is considered an anomaly according to the microcontroller's prediction, the application displays an alert notification, such as a suspected fire.

IV. Experimental Procedures

The experiment consisted of two distinct phases. In the first phase, the objective was to acquire data using the most appropriate sensors for implementing the system, taking into account the need for temperature sensors, gas and smoke sensors, and CO2 sensors, as well as a microcontroller capable of capturing the data obtained by the sensors. In the second phase, the collected data is classified, trained, and tested to detect the presence or absence of anomalies.

TABLE I: Comparison of IoT-based kitchen fire detection systems.

Paper	Sensors Used	Technology	Results and Limitations
[4]	MQ2, NodeMCU	Wi-Fi Monitoring	Displays real-time status, but not includes machine learning for predictive detection.
[5]	MQ2, DHT11, IR	Arduino Uno with serial monitor	Detects gas leaks and changes temperature, but needs real-time notifications and integration with mobile applications.
[6]	MQ3, NodeMCU	Alarm system and exhaust fans	Includes automatic alarms and shutdown of stoves, but does not offer support for predictive integration or analysis of data for custom alerts.
[9]	NodeMCU, Firebase	Mobile App	Efficient integration with Firebase, but presents limitations in data collection in real-time and doesn't use machine learning.
Proposed System	BMP280, MQ2, CCS811	ESP32 with Firebase and ML	Uses machine learning to classify data on anomalies and non-anomalies with accuracy greater than 99%, but depends on Wi-Fi connectivity and faces challenges in noisy environments.

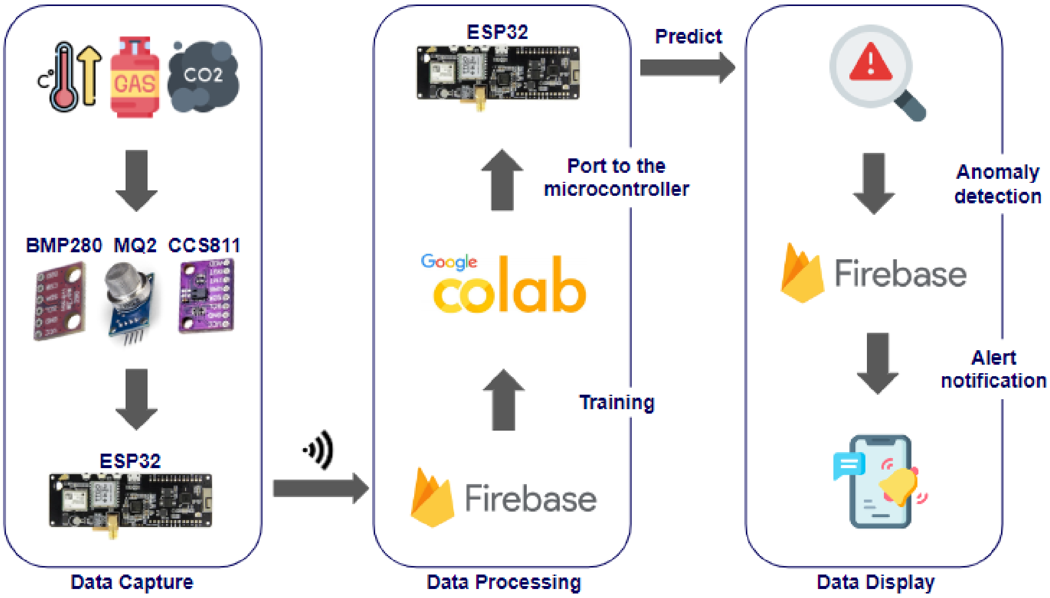

Fig. 1: Diagram of the architecture of the proposed system.

Fig. 2: Data Collection Environment: The Residential Kitchen.

1) Acquisition e Preprocessing: For this sensing stage, the BMP280 temperature sensor was used: temperature (°C) with a range of - 40 to 85 °C; the MQ2 smoke and gas sensor: smoke and gas (ppm) with a range of 300 to 10.000 ppm; and the CCS811 CO_2 sensor: CO_2 (ppm) with range: 400 to 32768 ppm; and the ESP32 TTGO T-Beam Microcontroller that captures the data obtained by the sensors and sends it via Wi-Fi to the next stage of data storage.

For data acquisition, the following setup was set up in a residential kitchen: the data collection device was initially placed next to the stove, but it was found that the data collected were inconsistent, so the data collection device was placed at the height of an air purifier, at a distance of 75 to 80 cm above the stove. Data were collected during meal preparation throughout the day. Collections were performed as follows: At breakfast, a collection of 16 minutes and 16 seconds, while at lunch, three collections were performed: the first of 30 minutes and 3 seconds, the second of 40 minutes and 26 seconds, and the third of 50 minutes and 4 seconds; at dinner a collection of 28 minutes and 59 seconds was performed, totaling 9596 samples. The values obtained were the values of the BMP280, CCS811, and MQ2 sensors and the timestamp that shows the date and time the collection was performed. The sensors were connected to an ESP32 microcontroller that receives the collected data and sends this information to the Firebase database via Wi-Fi. Firebase has an efficient integration with ESP32 and facilitates communication with the application.

2) Training and Testing: After data capture and storage in the Firebase Database, two classes are assigned for anomaly classification: (0) Non-anomaly (less than or equal to the

Threshold) and (1) Anomaly (greater than the Threshold). To determine the Threshold T, we use two variables: the mean of the data (μ) and the standard deviation (σ). The formula is given by:

$$T = x\ \mu + 1.5\ x\ \sigma$$

Where x is the instance to be classified, the captured data can be accessed through Google Colab, allowing access to Firebase data and Machine Learning training. Two hidden layers of a sequential neural network were used.

To carry out the training, the dataset needed to be increased by 10.404 synthetic data samples, resulting in a total of 20.000 samples. The data was divided into 70% training (14.000 samples), 15% testing (3.000 samples), and 15% validation (3.000 samples).

3) Porting to Microcontroller: To perform the model prediction on the edge, it was first necessary to convert the TensorFlow model to TensorFlow Lite and then add it to the project as the file “Anomaly_Model.h", having only 4852 bytes. After this process, the model is embedded in the ESP32 using the EloquentTinyML library. Based on the data from the Serial port, it takes about 510 to 530 µs for each prediction.

V. Experimental results

Based on the training data, we can observe the Confusion Matrix shown in Fig. 3, and by evaluating the model with the training data, we obtain an accuracy of 0.995. Based on the test data, we can observe the Confusion Matrix shown in Fig. 4, and by evaluating the model with the test data, we obtained an accuracy of 0.994. The results demonstrate superior performance, but it is essential to compare with similar works to validate the model's effectiveness. For example, works such as [4] and [9] report similar approaches for anomaly detection in kitchens, but with different sensors and without machine learning, obtaining lower accuracies due to the lack of advanced predictive methods. This result reinforces the differential of the proposed system by incorporating machine learning directly into the ESP32 microcontroller.

It should be added that the use of synthetic data to augment the training set deserves further discussion. While synthetic data helped to balance the set and improve the learning ability of the model, it is essential to evaluate the impact on generalization to real data. Future studies could investigate strategies to validate the model exclusively with real data or with more balanced combinations of synthetic and real data to ensure that the system can accurately identify anomalies even in varied scenarios, such as different kitchen sizes, the presence of drafts, or external interference on the sensors. This additional validation would ensure that the model performs in practical applications, reducing the likelihood of false positives or negatives in real-world scenarios.

1) Confusion Matrix: Based on the training data, we can observe the Confusion Matrix shown in Fig. 3, and evaluating the model with the training data, we obtain an accuracy of 0.995.

Based on the test data, we can observe the Confusion Matrix shown in Fig. 4, and evaluating the model with the test data, we obtain an accuracy of 0.994.

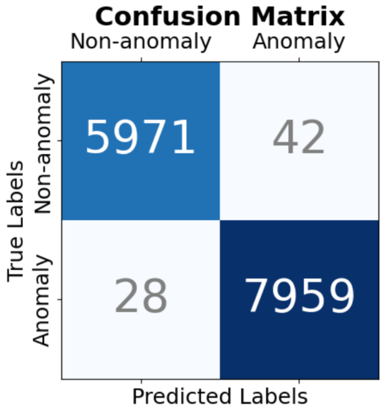

Fig. 3: Training Data Confusion Matrix.

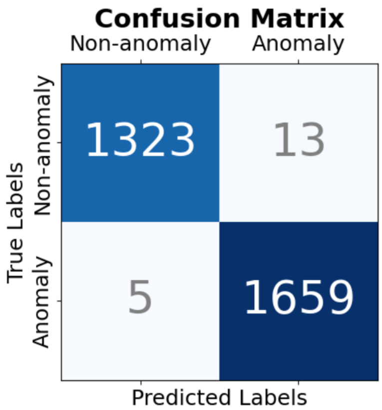

Fig. 4: Test Data Confusion Matrix.

2) Anomaly Detection: Fig. 5 and Fig. 6 show the data obtained by the Serial port during system operation. On the left side, it can be observed that the reading data is sent to Firebase in JSON format, containing the fields referring to the reading of each sensor and the timestamp. On the right side, there is the prediction data of the model in ESP32: input, a vector containing the normalized values of the readings of the BMP280, MQ2, and CCS811 sensors, respectively; benchmark, containing the duration of each prediction in µs; and classification, being 0 for Non-anomaly (Fig. 5) and 1 for Anomaly (Fig. 6).

In Fig. 6, the red markings indicate that the MQ2 sensor readings (gas and smoke) are high, resulting in the Anomaly classification. Thus, the alert notification (Fig. 7) is sent to the user.


```
{
  "fields": {
    "bmp280": {
      "doubleValue": 27.43
    },
    "mq2": {
      "doubleValue": 1431
    },
    "ccs811": {
      "doubleValue": 400
    },
    "timestamp": {
      "stringValue": "2024-04-18 02:23:10"
    }
  }
}
```


```
input: [0.185750, 0.240638, 0.000000]
It takes 529us for a single prediction
classification: 0
>>> NON-ANOMALY DETECTED <<<
```

Fig. 5: Serial port data: Non-Anomaly.

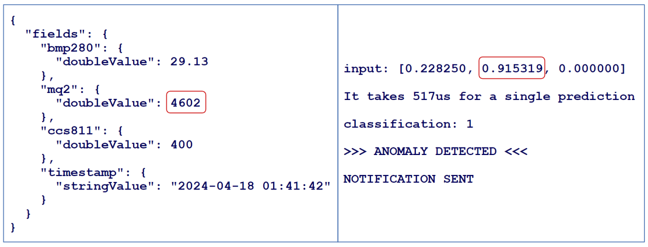

Fig. 6: Serial port data: Anomaly.

3) Alert Notification: For the user to obtain system updates, an Android application was developed in Java, containing user authentication, current readings, and alert notification if an anomaly is detected by the ESP32, as shown in Fig. 7. The notification delivery service works in the background, requiring only an internet connection. That is, the application does not need to be open for the notification to be received.

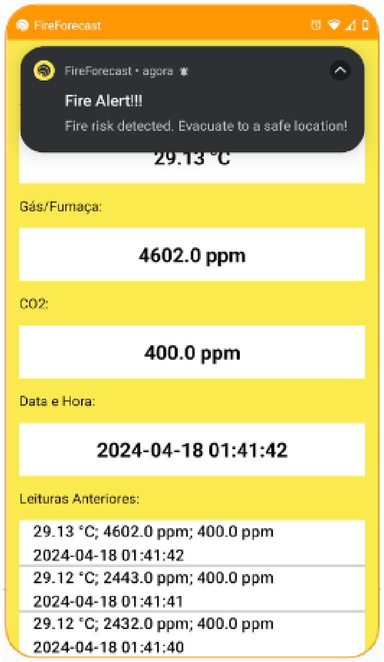

Fig. 7: Alert Notification.

VI. Research Limitations

Although the results presented are promising, the system has limitations that may affect its application in complex scenarios, such as commercial kitchens, homes with atypical layouts, or environments with unstable connectivity and environmental interference. The MQ2 and CCS811 sensors have a limited range, detecting gases only in the immediate area of installation, which may make it difficult to identify leaks in large spaces or complex layouts. In addition, environmental interference, such as air currents and sudden changes in temperature and humidity, may compromise the accuracy of the MQ2, BMP280, and CCS811 sensors, requiring initial calibration or periodic adjustments to ensure reliability. The system also relies on Wi-Fi connectivity for real-time data transmission, making it vulnerable in locations with unstable or nonexistent signals. Another limitation is using synthetic data in model training, which requires more extensive validation with real data to ensure its generalization in different scenarios. Finally, anomaly detection based on fixed thresholds, calculated from the mean and 25% of the standard deviation, may be insufficient for scenarios with more significant variability, suggesting dynamic methods, such as machine learning, to improve accuracy. These limitations list important areas for system improvement, guiding future research and development efforts.

VII. Conclusion

At the end of the study, it was possible to develop a system capable of monitoring, in real-time, risk conditions in residential kitchens, such as increased temperature, gas leaks, and the presence of CO2 at abnormal levels, using BMP280, MQ2, and CCS811 sensors connected to an ESP32 microcontroller. If anomalies are detected, the system alerts residents via an Android application, helping to prevent fires and minimize damage to people and property.

Future work should investigate using long-range sensors, multiple networked devices, and connectivity alternatives such as LoRa or GSM networks. In addition, we suggest investigating the use of dynamic anomaly detection thresholds through statistical methods or machine learning, aiming to adapt the system to variable conditions.

Acknowledgment

The authors would like to thank the Center of Electronic and Information Technology Research and Development (CETELI/UFAM) for the financial and institutional support to the authors of the article.

References

[1] W.-L. Hsu, J.-Y. Jhuang, C.-S. Huang, C.-K. Liang, and Y.-C. Shiau, “Application of internet of things in a kitchen fire prevention system,” *Applied Sciences*, vol. 9, no. 17, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/17/3520

[2] G. Shukla, A. Saini, S. Rawat, A. Upadhyay, G. Gupta, and M. Pal, “A smart iot and ai based cooking system for kitchen,” in *2023 International Conference on Disruptive Technologies (ICDT)*, 2023, pp. 543–548.

[3] L. M and J. J. Jeya Sheela M.E, “Designing an iot based kitchen monitoring and automation system for gas and fire detection,” in *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, 2022, pp. 346–353.

[4] B. Bharti, I. Bharadwaj, and A. Bhardwaj, “Smart kitchen using iot,” *Global Journal of Innovation and Emerging Technology*, vol. 1, pp. 27–31, 01 2023.

[5] M. E. Seno, A. A. Abed, Y. A. Hamad, U. M. Bhatt, B. Ravindra Babu., and S. Bansal, “Cloud based smart kitchen automation and monitoring,” in *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, 2022, pp. 1544–1550.

[6] D. K. Shah, R. Singh, A. Gehlot, S. Khantwal, A. J. Ahmad, and S. V. Akram, “Smart kitchen: Real time monitoring of kitchen through iot,” in *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, 2022, pp. 718–722.

[7] S. Suwarjono, I. Wayangkau, T. Istanto, R. Rachmat, M. Marsujitullah, H. Hariyanto, W. Caesarendra, S. Legutko, and A. Glowacz, “Design of a home fire detection system using arduino and sms gateway,” *Knowledge*, vol. 1, pp. 61–74, 11 2021.

[8] H. Alqourabah, A. Muneer, and S. M. Fati, “A smart fire detection system using iot technology with automatic water sprinkler,” *International Journal of Electrical and Computer Engineering (IJECE)*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233538800

[9] N. Umapathi and S. Sabbani, “An internet of things (iot)-based approach for real-time kitchen monitoring using nodemcu 1.0,” in *Futuristic Communication and Network Technologies*, A. Sivasubramanian, P. N. Shastry, and P. C. Hong, Eds. Singapore: Springer Nature Singapore, 2022, pp. 35–43.

[10] J. Yépez Rodríguez and S.-B. Ko, “Iot-based intelligent residential kitchen fire prevention system,” *Journal of Electrical Engineering Technology*, vol. 15, 08 2020.