ISSN 2278-3091
Volume 13, No.1, January - February 2024
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse041312024.pdf
https://doi.org/10.30534/ijatcse/2024/041312024

# An Event-Driven Mobile Application for Gas Leakage Monitoring and Detection using Firebase and IoT

**John D. Valle[1], Fe Yara[2]**
[1]University of Mindanao - Professional Schools, Davao City Philippines 8000, j.valle.488230@umindanao.edu.ph
[2] University of Mindanao - Professional Schools, Davao City Philippines 8000, fe_yara@umindanao.edu.ph

Received Date : December 21, 2023 Accepted Date: January 21, 2024 Published Date: February 06, 2024

**ABSTRACT**

Gas leakage detection devices have been on the market to address the detrimental effects of gas leaks. However, most of these devices are industry-scale, expensive, and unsuitable for household purposes. As a result, researchers carried out studies for alternative devices, yet certain gaps were still found and presented. With this, the study proposes an improved alternative gas leakage monitoring and detection system that addresses gaps in existing studies. The system would run in an event-driven architecture with additional sensors on top of an MQ5 gas sensor. The IoT component is powered using a Wemos D1 Mini with a built-in ESP8266 WiFi module that allows MQTT communication between a cloud server and a mobile application. By leveraging technologies such as Firebase, Webhooks, and Cloud Functions, users are alarmed of a gas leak remotely while triggered actuators alarm users locally. The study carried out several tests using a cigarette lighter to simulate a gas leak, and the system successfully sent out alarms when fire was detected or when gas levels exceeded the set threshold. The study employed an algorithm for minimizing false alarms using an Exponential Moving Average (EMA) and results show that EMA is not only effective at reducing false alarms but as well as false negatives. This research offers a promising solution, addressing gaps in existing alternatives for effective household gas leakage detection and monitoring.

**Key words**: Gas leak, Internet of Things, Real-time, Detection, Monitoring, Exponential Moving Average

## 1. INTRODUCTION

Liquefied Petroleum Gas (LPG) leakage has been known to cause suffocation, fire, explosion, injuries, and even death [1,2]. These unfortunate events are usually linked to substandard cylinders, old valves, worn-out regulators, and the lack of general knowledge in handling LPG cylinders [3]. Due to these circumstances, devices have been created and made available in the market to prevent the possible effects of gas leaks.

Traditional gas leak detection devices, such as the HandheldGD01, AmprobeGSD600, and Analox Sensor Technology are available on the market. These devices can be affixed to a fixed position of the suspected leak source or hand-held to point and allow continuous spot readings [4,5]. Although these devices are helpful, they tend to be bulky, expensive, and difficult to source, often sold for factories or large-scale manufacturers.

In light of these challenges, various research has been conducted to produce improved alternative gas leakage devices utilizing modern technologies such as the Internet of Things (IoT). The concept of IoT has been widely used in various fields. Its applications are found in smart farming systems [6] and distance learning [7], and it is commonly used in smart home systems [8]. In the context of monitoring and detection, utilizing IoT has also proven to be effective. In telehealth monitoring, a person's body temperature and location can be remotely monitored through a non-contact temperature sensor and a mobile application [9]. Meanwhile, in a closely related study, a remote pH sensor was used to monitor acid leaks in an underground pipeline. This system served as an early warning system that reduced the errors from manual pipe inspections [10].

In the scope of gas leakage monitoring and detection, the general approach taken by related studies is using an MQ2, MQ5, or MQ6 gas sensor. These sensors are proven effective in detecting the presence of LPG and are commonly paired with a microcontroller such as an Arduino UNO or a NodeMCU [11,12,13].

The microcontroller is used to process the values from the gas sensor, and it is where a threshold value for indicating a leakage is predefined. In the context of gas leakage detection, most studies alarm its users locally using a buzzer. In other studies, users are notified through an SMS or an email. Meanwhile, in the context of gas leakage monitoring, an LCD is often connected to the microcontroller or a mobile application is developed to display the values of the gathered data [14,16].

All of the aforementioned studies have concluded that utilizing IoT to develop an alternative gas leakage-detecting system is feasible and effective. However, a few technical limitations were noted by the researchers, who suggested several areas for improvement.

Some researchers observed that their system immediately alerts users of a gas leak, reducing the time to act and potentially preventing a fire in remote settings. However, the system uses raw, unfiltered sensor data, which increases the risk of false alarms due to noise or fluctuations. Ideally, warnings could be initially sent before an alert [6].

Moreover, existing studies only address remote notifications through SMS or email and local notifications using a buzzer. Some studies also take remote monitoring into account using a mobile application, and some use an LCD display for local monitoring. However, no actual study addresses monitoring and detection in both local and remote contexts [13,14,15].

This is a significant oversight, as it leaves users vulnerable in both scenarios. For example, if a gas leak occurs in a remote location, users may not receive a notification in time to take action. Additionally, if a gas leak occurs in a local location, users may be unable to hear the buzzer or see the LCD display. A comprehensive gas leak detection system should address both local and remote scenarios, ensuring that users can monitor gas values or get alerted regardless of their location.

Lastly, it was observed that these studies commonly use a request-driven architecture to send data from the IoT component to the Internet [17,18]. This approach could be better for gas leak detection and monitoring, as it can lead to data loss and delays. A more reliable and efficient approach is to use an event-driven architecture, such as the Message Queueing Telemetry Transport (MQTT) protocol. This architecture ensures that data is received in real-time, minimizing the risk of data loss [19].

With these gaps observed and presented, the study is focused on developing a comprehensive alternative system for gas leakage detection and monitoring that addresses gaps in existing research and presents it as features that offer safety mechanisms to users.

**Objectives of the Study**

The project's primary objective is to take LPG users away from the potentially harmful effects of a gas leak by providing an alternative gas leakage monitoring and detection system through a dedicated mobile application with extended remote capabilities using the Internet of Things.

This study specifically aims to perform the following functionalities to achieve the general objective:

1. Implement gas concentrations, humidity, temperature, and flame data using IoT sensors;
2. Send real-time updates to a cloud-based platform from the IoT device using a WiFi module;
3. Implement an algorithm that prevents false alarms through data smoothing using Exponential Moving Average
4. Display real-time data received from a cloud server in a mobile application through charts using React Native;
5. Warn users before the gas leak threshold is exceeded and alert during a suspected gas leak through an actuator and a device notification.

## 2. METHODOLOGY

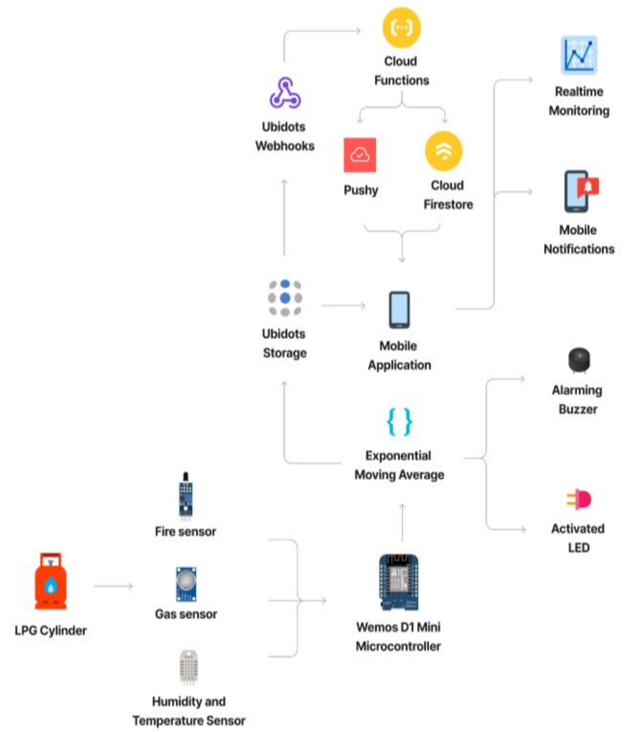


**Figure 1:** Conceptual Framework

Initially, data input is automatically acquired from a potential leakage from a nearby LPG cylinder, as shown in Figure 1. Specifically, gas, flame, humidity, and temperature data are gathered through sensors connected to a microcontroller.

The data gathered using multiple sensors is processed by the microcontroller. Specifically, the gas sensor data from the collected sensor data is applied with the Exponential Moving Average (EMA) formula. Applying EMA ensures the data is free of noise due to volatility. This mechanism applies as an initial filter for the data to prevent false alarms from occurring.

There will be certain cases where there will be fluctuations in concentrations or changes in the sensor readings that would cause data to exceed the threshold without a filtering mechanism. In this case, using EMA would be efficient. After applying EMA to the collected data, information will be sent to Ubidots cloud storage through the MQTT protocol.

With Ubidots' service, webhooks can be triggered to perform events under certain conditions. In this case, webhooks are set up, performing HTTP POST requests to the endpoints exposed through Google Cloud functions. These HTTP requests create notifications in Cloud Firestore and send push notifications using Pushy.

While Ubidots is the storage for IoT data, Google's Cloud Firestore is added to store user data, including notification history, settings, and authentication. Each sensor data will be displayed through charts or gauge meters for monitoring purposes. A threshold for the gas sensor will be set for gas leakage detection. When a possible gas leakage is detected, the user will be alarmed remotely by a push or in-app notification using the mobile application. At the same time, the IoT component will actuate the buzzer and an LED.

### 2.1 Hardware Implementation

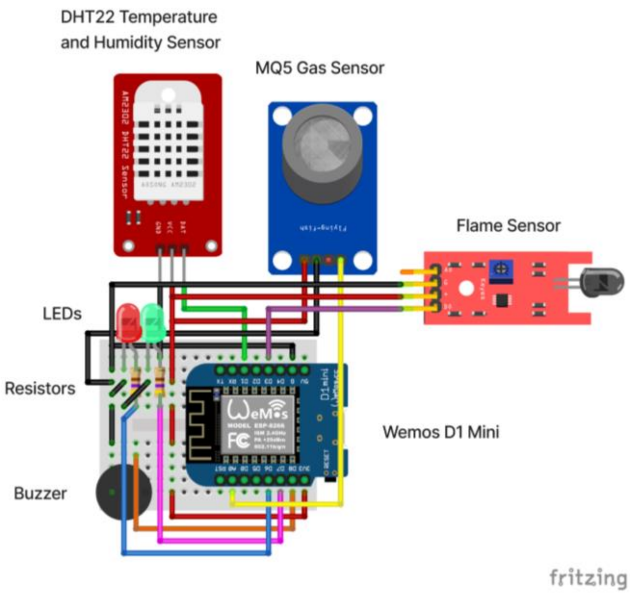


**Figure 2:** Schematic diagram of the IoT component

The Schematic diagram of the IoT component shows the study has three (3) major sensors for gathering data. As presented in Figure 2, the IoT component will have two (2) actuator types: LEDs and a buzzer. A built-in ESP8266 WiFi module is used to enable internet connectivity. All of the presented components will be connected and powered by a Wemos D1 Mini microcontroller. Sensors such as the DHT22 Humidity and Temperature sensor and flame sensor send data through the digital inputs of the microcontroller. Meanwhile, the MQ5 gas sensor is connected to the analog input.

### 2.2 Software Implementation

After the IoT component gathers data from the sensors, a moving average method is applied to the raw data acquired from the gas sensor. The study applies the exponential moving average (EMA) formula to the raw gas data before processing it in the cloud. Unlike other moving averages like Simple Moving Average (SMA) and Cumulative Moving Average (CMA), which entirely discards old data, EMA gives weight to recent data and reacts more quickly to changes.

The EMA formula is expressed as:

$$EMA_c = (Value_c * k) + (EMA_p * (1 - k)$$
$$k = \frac{2}{n+1}$$

where:

$p = previous;$
$k = smoothing\ factor\ (weighted\ multiplier)$
$n = window\ size;$

```
float getEmaNumber (float prev, float curr) {
  float k = 0.33;
  return (curr * k) + (prev * (1 - k));
}
...
float prev = 0;
float curr = getEmaNumber(prev, gasValue);
prev = curr;
```

**Figure 3**: Arduino Code snippet in C for applying EMA

Currently, the study sets the window size represented by n to equal 5. With the n set to 5, the smoothing factor represented by k is equal to 0.33, as presented in Figure 3. The function *getEmaNumber* takes in the previous EMA value and the current value that will be applied with EMA. After getting the current EMA value, the variable prev is set to the current one represented by the variable curr.

After the gas data is applied with EMA, it is sent to the cloud storage using Ubidots Arduino library. The mobile application receives and displays this information. The primary software component of the study was developed using React Native and Typescript. React Native is a JavaScript library based on ReactJS that is made for building mobile applications.

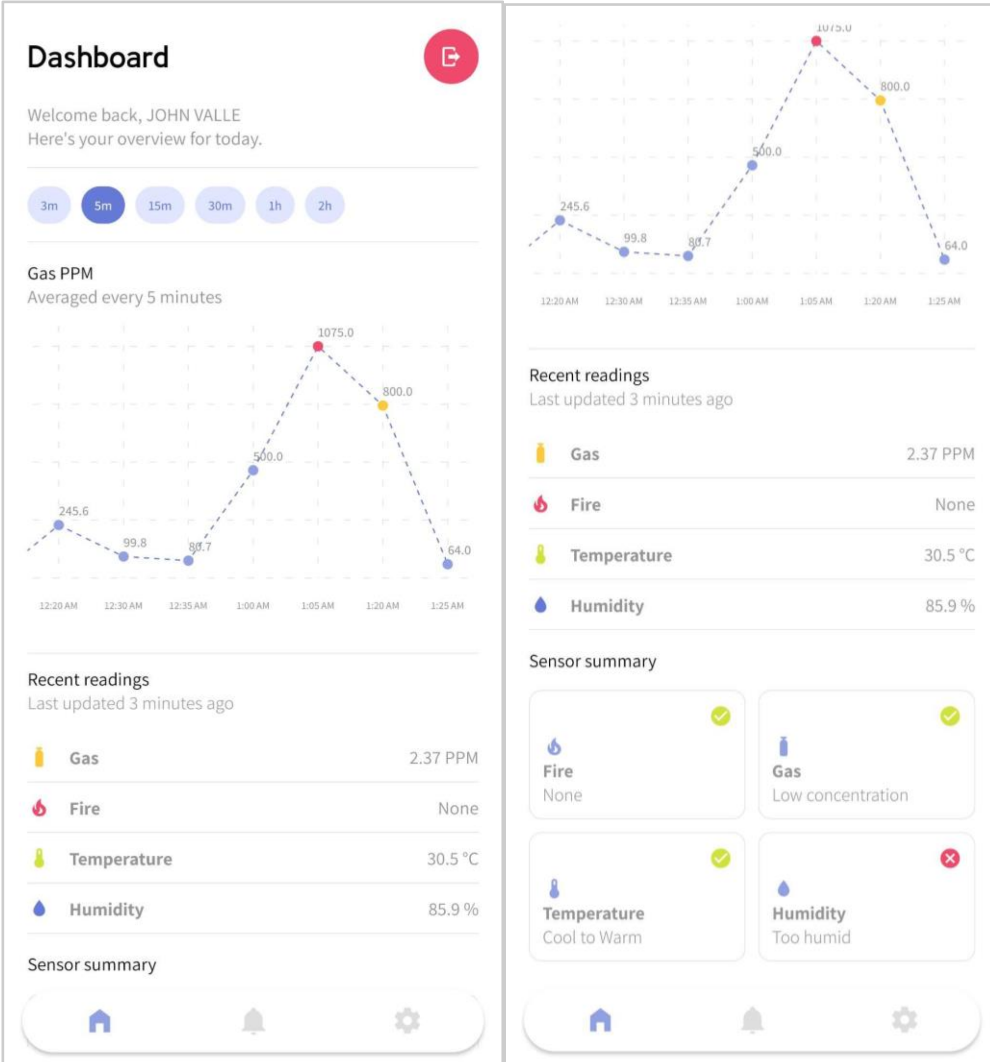


**Figure 4:** Dashboard page of the application

Users will see three (3) sections on the dashboard, as presented in Figure 4, which are the chart, recent readings, and sensor summary sections. For the first section, gas data is visually interpreted in a line chart. This is where the movements of the sensor data through time are visible.

A user has the capability to view the chart in different time intervals, starting from three (3) minutes up to two (2) hours. Data displayed in the chart is not real-time, and gas data is averaged depending on the selected time interval.

Below the overview section is the "Recent readings" section. In this section, the numerical data for each sensor is displayed. For a better experience, a user can see gas and fire changes in real-time. Other sensor values, such as temperature and humidity, are updated every three (3) seconds. Aside from the fire sensor interpreted in boolean logic, all other sensors will be displayed in numbers with their unit of measurement.

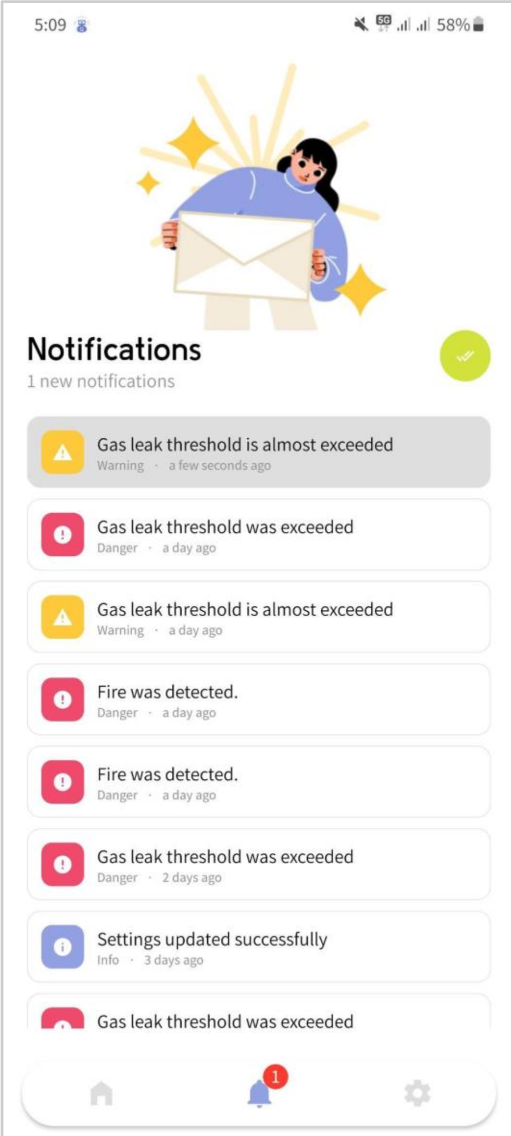


**Figure 5**: Notifications page of the application

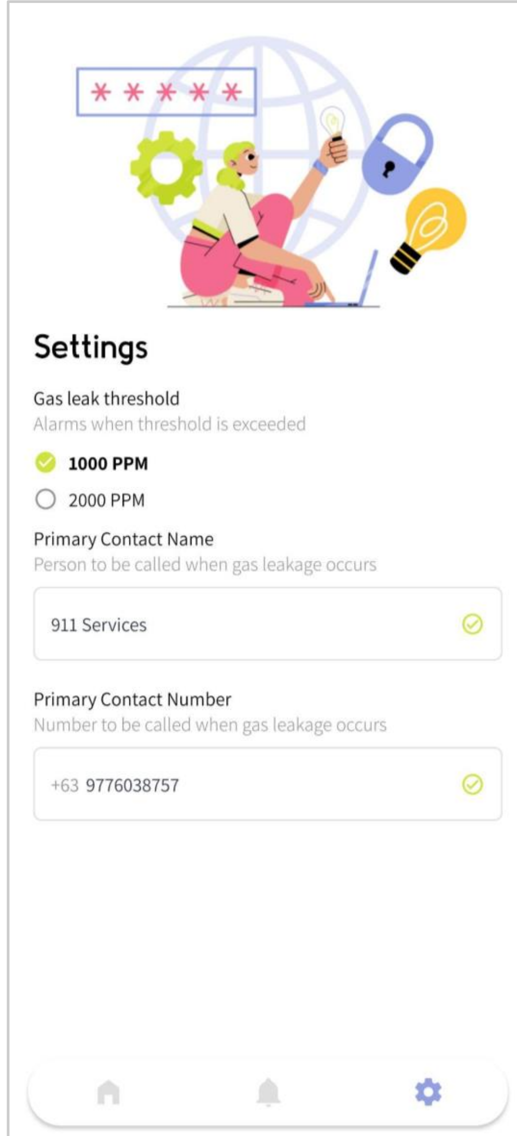


**Figure 6**: Settings page of the application

Figure 5 shows the notifications page where users will see all historical notifications the app has received. A user can receive three types of notifications: an alert, a warning, or an info notification. Alerts are received when a gas leak is detected, while warnings are received when sensor values are near the threshold or are at moderate levels. On the other hand, info notifications are received when there are changes in the settings or updates in the application. A badge is added to the notification page icon on the bottom tab when new notifications arrive.

In Figure 6, users have the capability to modify the threshold for detecting the gas leak. By default, the app would have a threshold of 1000 ppm. When a user wants to be alarmed earlier or later, the threshold could be adjusted from 1000 to 2000 ppm. The user could also set the primary contact's information in the application. By default, the primary contact's number is set as 911, but the user could change it to anyone who could be called when a gas leak occurs. This contact information is linked to the device's phone app and is used as an auto-fill.

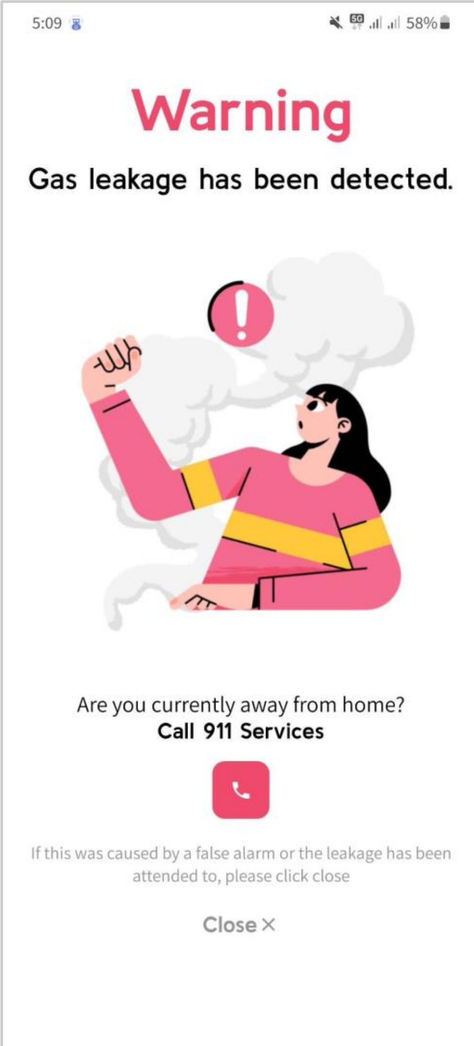


**Figure 7**: Alarm page of the mobile application

Meanwhile, the alarm page in Figure 7 serves as the in-app notification that will show when a gas leak occurs while the app is in use. This alert could be closed by pressing the close button if the user is near the component and the gas leak is suppressed. However, when the user is remotely monitoring the app, and it is not immediately possible for the user to close the source of the gas leak, the primary contact added in the settings could be reached through the call button to address the gas leak.

The system will alarm the user only if certain conditions, even if a presence of gas is detected. To define the conditions, the following symbols are defined as:

$P = Gas \geq threshold$
$Q = Fire\ detected$
$R = Temperature \geq threshold$
$S = Humidity \geq threshold$

With the expression $(p \vee (q \vee ((q \wedge r) \vee (q \wedge s))))$ the user will be alarmed by gas leak locally and remotely except for the following conditions:

1. Humidity is greater than or equal to the threshold
2. Temperature is greater than or equal to the threshold
3. Both humidity and temperature are greater than or equal to the threshold
4. All sensors are less than the threshold

Additionally, propane, the primary component of LPG, has a safe area within 0-20% of its Lower Explosive Limit (LEL). This range implies that at the LEL of 2.1% or 20,100 PPM, concentrations between 0 and 4,020 PPM are safe. However, according to AFC International (n.d.), if PPM values reach the LEL, a risk of an explosion will exist as concentration values exceed the safe area range. This level is up to the Upper Explosive Limit (UEL) of 9.5%, or at 90,500 PPM of propane concentration in the air, which has an extremely high risk of causing an explosion.

Although the employed MQ5 sensor can only detect up to 10,000 PPM or 1% of propane, this amount of concentration should already be informed as alarming as this is beyond the safe area, and it could detect the set limit for being IDLH at 2,000 PPM as per the LEL Charts (2020). This results in the study setting the default threshold for gas at 2,000 PPM. For flexibility improvement, the research allows the user to increase the threshold sensitivity by setting the threshold limit as sensitive as 1000 PPM and receiving warnings at 750 PPM.

## 3. RESULTS AND DISCUSSION

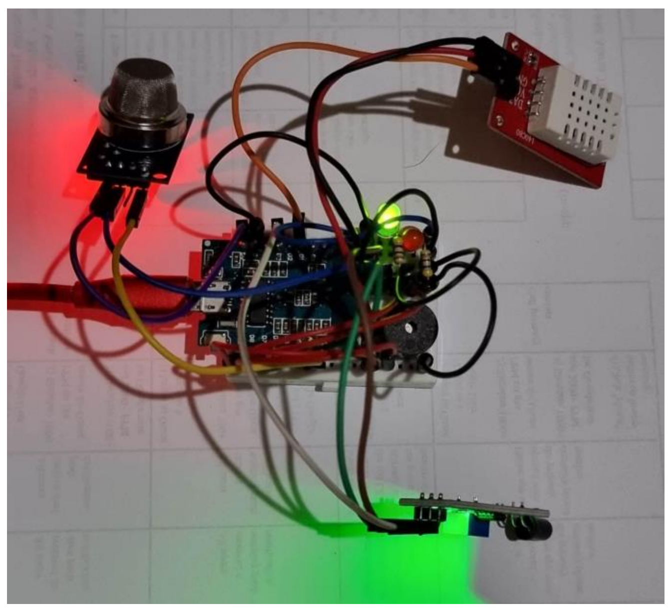

**Figure 8**: Default IoT component state

Initially, as presented in Figure 8, a green LED is activated while the red LED is turned off. This state indicates that the IoT component is connected to the Internet and has established communication with the cloud server.

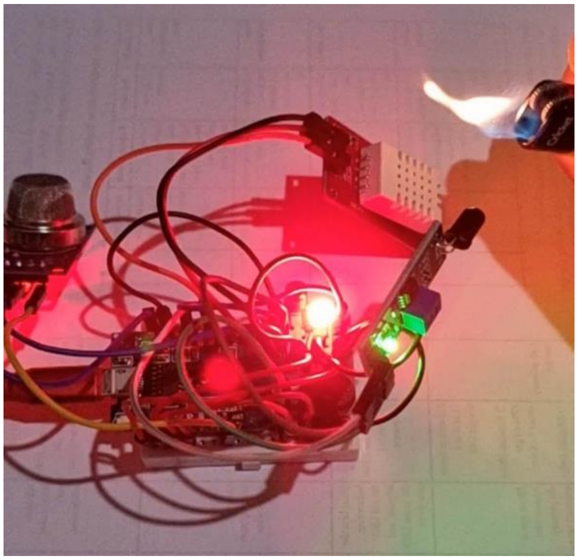

**Figure 9**: Triggered actuators

The cigarette lighter simulated flame detection, aside from gas leak detection. As presented in Figure 9, a red LED is shown to be activated, while a buzzer emits a high-pitched sound. The presented figure shows that the hardware component detects a presence of a flame.

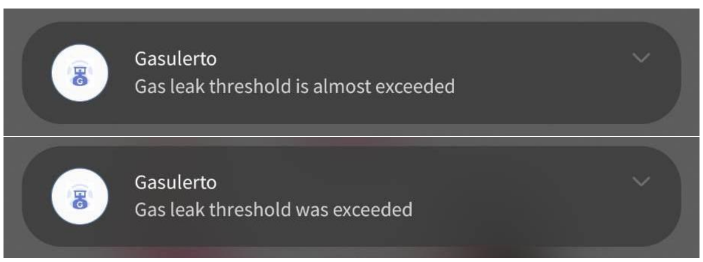


**Figure 10**: Push notifications received in the mobile application

Aside from hardware notifications, push notifications indicating a possible leak were also received in the mobile application, as presented in Figure 10. These notifications were received when the state of the mobile application was quit or running in the background. On top of push notifications, when the mobile application was being used in the event of a gas leak or fire, the alarm page was shown to the user (Fig. 7) to allow communication to a primary contact or emergency services.

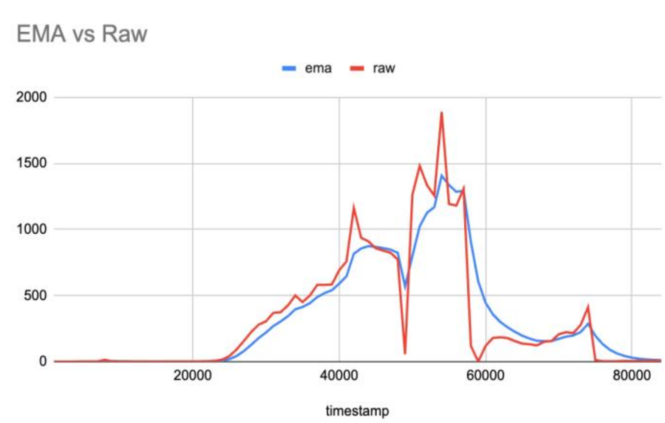


**Figure 11**: Comparison of EMA-applied gas data vs raw gas data

Moreover, EMA-applied gas data can be seen as smooth in trend when gradually reaching ideal thresholds or dropping from a higher value as seen in Figure 11. When applied to various thresholds, false alarms occurred as expected, as presented in Table 1 below.

**Table 1:** EMA testing results on different thresholds

| Threshold (PPM) | Samples | False alarms | No alarm | Error % |
|---|---|---|---|---|
| 500 | 22 | 3 | 4 | 31.82% |
| 750 | 16 | 1 | 1 | 12.50% |
| 1000 | 9 | 2 | 0 | 22.22% |
| 1250 | 8 | 4 | 2 | 75.00% |

With a threshold of 500 PPM, 3 out of 22 samples were tagged as false alarms. Moreover, it was found that raw data also did not alarm in some cases. In its practical implication, false alarms could be bothering or disruptive when there is too much occurrence. However, not triggering alarms could potentially be harmful.

The same case also was exhibited when the threshold was set at 1250 PPM. Out of 88 samples that exceeded the threshold, four samples were false alarms, and two were not. This means that 75% of the acquired data were unreliable. The thresholds 750 to 1000 PPM ranked the lowest in terms of error percentage at 12.50% and 22.22%, respectively.

Although the two thresholds had an occurrence of false alarms, only 1 sample did not cause an alarm. With the results presented, it is understood that without EMA, unusable data could range from 12.50% to 75% of the total sample. It is ideal that data is smoothened to ensure data is stable and usable.

## 4. CONCLUSION

From the testing and deployment that were done, the research concludes that the study was able to develop an alternative IoT device capable of local and remote gas leakage monitoring and detection through a dedicated mobile application. The study also concludes that aside from reducing a minimum of false alarms, applying EMA on raw gas data also prevents the possibilities false negatives or alarms that are not triggered. In detail, the study concludes:

1. the system was able to implement gas concentrations, humidity, temperature, and flame data using IoT sensors;
2. real-time updates were received in cloud-based platform from the IoT device using a WiFi module;
3. exponential moving average was effective at preventing false alarms when detecting gas leaks;

4. real-time data received from a cloud server in a mobile application was displayed through charts using React Native;
5. users were notified before and during a suspected gas leak through an actuator and a device notification.

## 5. RECOMMENDATIONS

From the performed testing and limitations of the study, the researcher recommends the following to further improve the study:

1. integrate additional flame sensors or use a 5-channel flame sensor for a wider flame detection range;
2. create an ios version of the application to have wider audience reach;
3. add additional IoT components to have a better overall range detection and enable wireless sensor network communication;
4. allow device tagging per user to allow devices to be connected and available to certain users only.

## REFERENCES


[1] Liu, E., Yue, S.Y., and Lee, J. A Study On LPG As A Fuel For Vehicles, 1997.

[2] Fraiwan, L., Lweesy, K., Bani-Salma, A., & Mani, N. (2011). A wireless home safety gas leakage detection system. 2011 1st Middle East Conference on Biomedical Engineering. doi:10.1109/mecbme.2011.5752053

[3] Muthuvinayagam, M., Meganathan, S., Janakiraman, S., & Dineshraja, S. (2014). Industrial gas monitoring with safety closure. International Journal Of Engineering Research & Technology (IJERT), 3(3), 1035-1040.

[4] Kodali, R. K., Greeshma, R. N. V., Nimmanapalli, K. P., & Borra, Y. K. Y. (2018). IOT Based Industrial Plant Safety Gas Leakage Detection System. 2018 4th International Conference on Computing Communication and Automation (ICCCA). doi:10.1109/ccaa.2018.8777463

[5] Chraim, F., Bugra Erol, Y., & Pister, K. (2016). Wireless Gas Leak Detection and Localization. IEEE Transactions on Industrial Informatics, 12(2), 768–779. doi:10.1109/tii.2015.2397879

[6] Rukhiran, M., & Netinant, P. (2020). IoT Architecture based on Information Flow Diagram for Vermiculture Smart Farming. *TEM Journal*, *9*(4), 1330–1337. https://doi.org/ 10.18421/TEM94-03

[7] Ilieva, G., & Yankova, T. (2020). IOT in distance learning during the COVID-19 pandemic. *TEM Journal*, 1669–1674. https://doi.org/10.18421/tem94-45

[8] Fuada, S., Yusuf Fathany, M., Adiono, T., & Afifah, K. (2021). Controlling mini exhaust fan through Android-based smartphone for IOT-based Smart Home System. *TEM Journal*, 1301–1306. https://doi.org/10.18421/tem103-37

[9] Saraubon, K., & Limthanmaphon, B. (2021). IOT telemonitoring system for covid-19 quarantine. *TEM Journal*, 105–112. https://doi.org/10.18421/tem101-13

[10] Asraf Hairuddin, M., Fikri Nazri, M., & Farid Saaid, M. (2022). IOT-based detection and early warning system for acid leaking in underground pipeline. *TEM Journal*, 1674–1679. https://doi.org/10.18421/tem114-31

[11] Unnikrishnan, S., Razil, M., Benny, J., Varghese, S., & Hari, C. V. (2017). LPG monitoring and leakage detection system. 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). doi:10.1109/wispnet.2017.83001

[12] Varma, A., Prabhakar S, & Jayavel, K. (2017). Gas Leakage Detection and Smart Alerting and prediction using IoT. 2017 2nd International Conference on Computing and Communications Technologies (ICCCT). doi:10.1109/iccct2.2017.7972304

[13] Mohd. Hussin, Z., Sulaiman, M. S., & Arasu, P. T. (2017). Highly Sensitive Portable Liquid Petroleum Gas Leakage Detector. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(2-8), 109–112. Retrieved from https://jtec.utem.edu.my/jtec/article/view/2638

[14] Ruqsar, H., Chandana, R., Nandini, R., & Surekha, T. (2014). Internet of Things (IOT) based real time Gas leakage Monitoring and Controlling. In Proceedings of the 2nd International Conference on Current Trends in Engineering and Management ICCTEM. Mysore, Karnataka, India.

[15] Pakhala, R. (2021). Smart LPG Gas Leakage Detector. SSRN Electronic Journal. doi: 10.2139/ssrn.3915520

[16] Jijusasikumar, S., Kaviya, K., Logida, R., Chinmaya, S., & Sangeetha, K. (2021). Gas leakage monitoring system using IOT. International Research Journal on Advanced Science Hub, 3(Special Issue ICARD 3S), 108–111. https://doi.org/10.47392/irjash.2021.075

[17] Ralevski, M., & Stojkoska, B. R. (2019). IoT based system for detection of gas leakage and house fire in smart kitchen environments. 2019 27th Telecommunications Forum (TELFOR). doi:10.1109/telfor48224.2019.8

[18] Zinnuraain, S. M., Hasan, M., Hakque, M. A., & Arefin, M. M. N. (2019). Smart Gas Leakage Detection with Monitoring and Automatic Safety System. 2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET). doi:10.1109/wispnet45539.2019.

[19] Donta, P.K. et al. (2022) 'Survey on recent advances in IOT Application Layer Protocols and machine learning scope for Research Directions', Digital Communications and Networks, 8(5), pp. 727–744. doi:10.1016/j.dcan.2021.10.004.