PROGRESS IN ENGINEERING APPLICATION AND TECHNOLOGY
e-ISSN: 2773-5303

PEAT

Vol. 7 No. 1 (2026) 380-388
https://publisher.uthm.edu.my/periodicals/index.php/peat

Development of a Hybrid Edge-Cloud Integrated Multi-Sensors Monitoring System Dashboard

Ivan Eu Sik Wei[1], Azlina Bahari[1*], Brian Law Jia Ching[1], Aimi Syamimi Ab Ghafar[1]

[1] *Department of Electrical Engineering Technology, Faculty of Engineering Technology Universiti Tun Hussein Onn Malaysia, 84600, Pagoh, Johor, MALAYSIA*

*Corresponding Author: lina@uthm.edu.my
DOI: https://doi.org/10.30880/peat.2026.07.01.044

Article Info

Received: 12 January 2026
Accepted: 20 February 2026
Available online: 25 May 2026

Keywords

Hybrid edge-cloud, monitoring dashboard, IoT, real-time data monitoring, System Usability Scale (SUS)

Abstract

This article describes the development of a hybrid edge-cloud integrated multi-sensor monitoring dashboard designed to address the limitations of traditional environmental monitoring through the utilization of IoT-enabled real-time data collection and visualization through an interactive dashboard. The system integrates both edge-based sensor nodes with cloud communication protocol and local data processing, creating a scalable monitoring framework for various environmental applications. The system utilizes a PT1000 temperature sensor, along with pH and TDS sensors from DFRobot to demonstrate the feasibility of data transmission from the sensor node all the way to the dashboard. The system is equipped with a Telegram alert as the main way to detect the connectivity of the sensor nodes. Due to the usage of a dashboard, a System Usability Scale (SUS) was utilized and a score of 56.25 was obtained, which indicates the need for further improvement on the dashboard until it reaches an industry standard.

1.0 Introduction

Environmental issues in recent years have been exacerbated in the advent of global warming. Earth is home to many natural disasters, which are weather or climate related as tropical cyclones, floods, tornadoes, draughts whose effects are incredibly damaging to all living beings. The unnatural warming of Earth's atmosphere causes issues where even the western mountains in North America region to warm up, leading to decrease in snowpack, increase in winter flooding and reduction in summer flows. Crops that utilized high water resources will also notably face major challenges. This highlights the great changes occurring in the world in terms of weather and its impact on human activities [1]. Thus, this has now sparked the need for environmental monitoring, driven by the growing awareness of ecological needs. To fight against the damage of environmental issues, one must first gather the necessary information to in turn create an effective countermeasure. Initial environmental monitoring relies heavily on manual sampling and laboratory analysis, which now can be seen as time-consuming and labour intensive. While can be accurate, it does not exactly provide real-time insight into most dynamic environments [2].

The emergence of Internet of Things (IoT) has now revolutionized the techniques to monitor the surrounding environment. It has now enabled the possibility of continuous, real-time data acquisition from

distributed sensor networks. The ability to integrate sensors, microcontrollers, wireless communication modules and cloud platforms to create a supporting pillar for a real-time monitoring network [3]. This provides the ability for developers and users to detect anomalies in the environment and ensure compliance with the environmental standards.

A critical development within IoT-based monitoring is its ability to integrate multiple sensors seamlessly. Utilizing multi-sensor system networks allows the provision of environmental conditions with simultaneous tracking of several interconnected variables. The common variables or parameters may include temperature, humidity, pH, pressure, different gaseous concentration, air quality, total dissolved solids (TDS) and many more depending on the requirements.

The integration of cloud computing and advanced data visualization tools such as dashboards further increases the usefulness of IoT monitoring systems. For example, cloud platforms such as The Things Network (TTN), Node-RED and MySQL, which are used in this project can help facilitate and handle issues of data ingestion, storage and management while integrated into visualization dashboard such as Streamlit offers a user-friendly interface for real-time data display aiding analysis. This is one of the examples that showcases the key components of modern smart monitoring frameworks.

IoT-based systems can integrate multiple sensors to monitor various environmental parameters simultaneously. When combined with edge applications, cloud platforms and data visualization dashboards, these systems allow efficient data storage, analysis, and real-time monitoring, forming the foundation of modern environmental monitoring solutions.

2.0 Methodology

In this section explains the overall process of developing the Streamlit-based web dashboard. Which constitutes the primary interface layer of the interactive monitoring system. The dashboard was designed with user-friendliness in mind, while fully providing useful and easy to understand graphs to aid in visualizing the fluctuations in the wastewater quality parameters.

The data transmission will be demonstrated utilizing Heltec WiFi LoRa 32(V3.1) Microcontroller and a few sensors such as PT1000 temperature sensor, DFRobot TDS and pH sensor. The reason this microcontroller was used is due to its ability to connect to a gateway via radio frequencies, specifically the ISM band, which in this project will be 915-928 MHz range. This project makes it so the sensor node will not be dependent on WiFi due to direct RF link, where the LoRa transceiver broadcasts data packets directly over air. With its low frequency radio, it also consumes very little power, allowing it to operate with minimal power consumption and allowing remote usage.

This project focuses solely on receiving and processing data obtained from sensor node in multiple checkpoints. From Figure 1, the LoRa end devices send data to the gateway and to the TTN live applications for initial collection. Utilizing MQTT protocols, one can retrieve the raw data from TTN via Node-RED, where the data will be processed. The processed data will then be stored in MySQL database for long-term historical record-keeping purposes. Next, the data stored in the database will then be displayed using Streamlit onto a web-based interactive dashboard. Figure 2 shows the overall project block diagram, starting from LoRa End Devices, moving up to TTN application, then into the dashboard. Machine Learning is part of a collaboration with the Development of Flood Risk Prediction using Water Level and Rainfall Volume using Multiple Regression Analysis. The Figure 3 illustrates the system flowchart. The flow chart explains how this project starts and progresses, such as starting the necessary software and hardware, how and where the data processing occurs, where the dashboard receives and how it displays the information stored.

Figure 4 illustrates the initial sketch for the dashboard. The initial idea is to have a multiple tab for multiple sensor node, making it easier to navigate between different sensor parameters.

The overall system also includes an alert system which will inform users of any disconnected sensor node. This alert utilizes Node-RED and Telegram. Telegram alert is a very simple system to utilize, using a Telegram Bot to send notification to the Telegram group, allowing the members to see chat messages. Using Node-RED, a node will be used to detect any incoming messages, and a timer mechanism was set to detect it. Meaning if there is no data transfer within the allocated time, the system will flag the sensor as disconnected, then proceed to send an automated message to the Telegram group. Figure 5 illustrates the example of node-RED connection to achieve this alert system, and there can be other methods as well and periodical checking should be done to ensure the system is functioning.

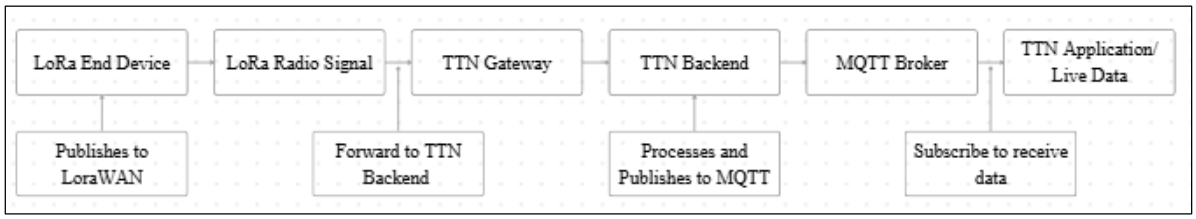

Fig. 1 Simplified system design flow

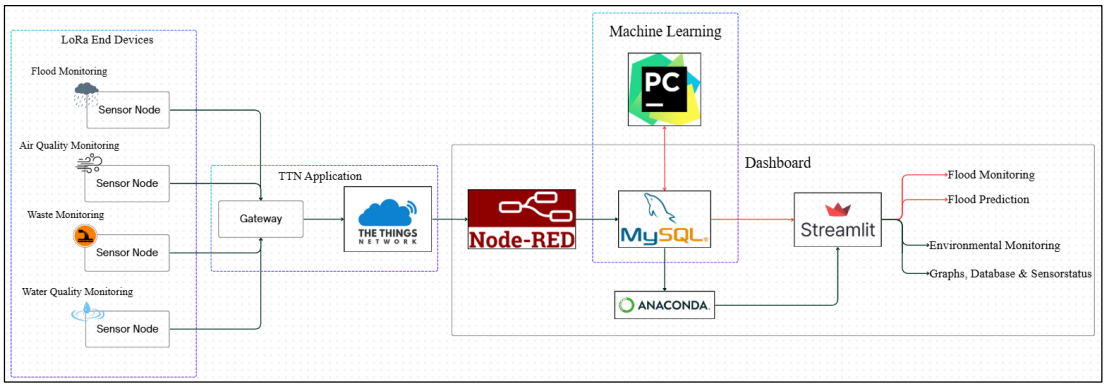

Fig. 2 Project Block Diagram

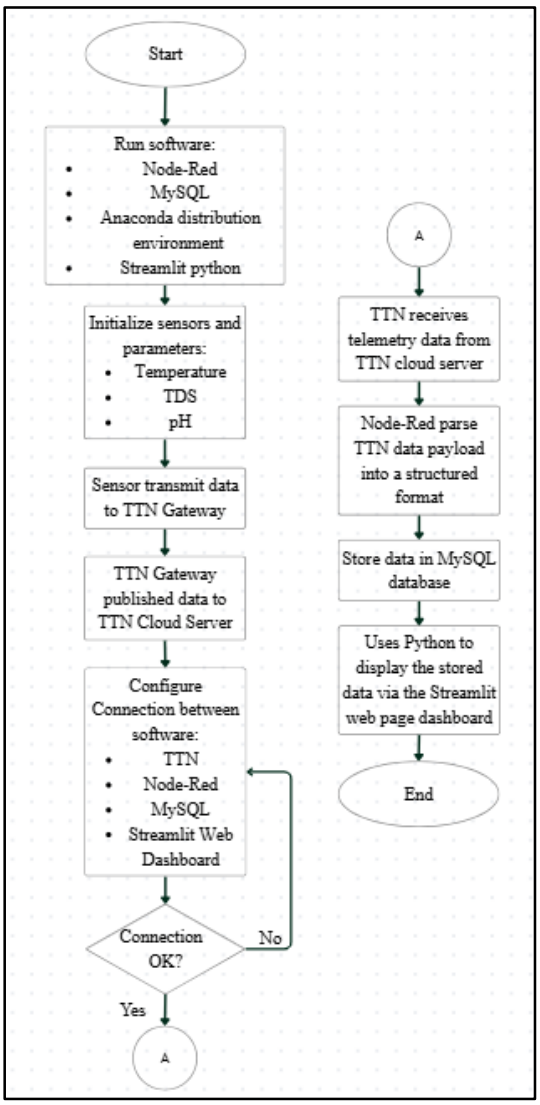

Fig. 3 Detailed system flowchart

Progress in Engineering Application and Technology Vol. 7 No. 1 (2026) p. 380-388

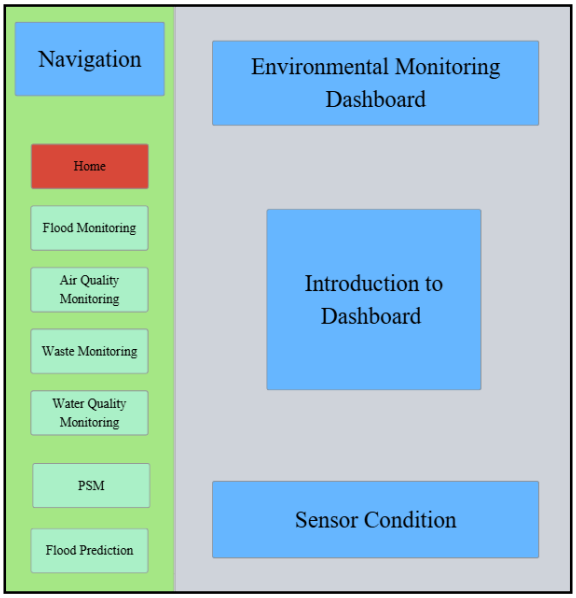

Fig. 4 Initial sketch of dashboard

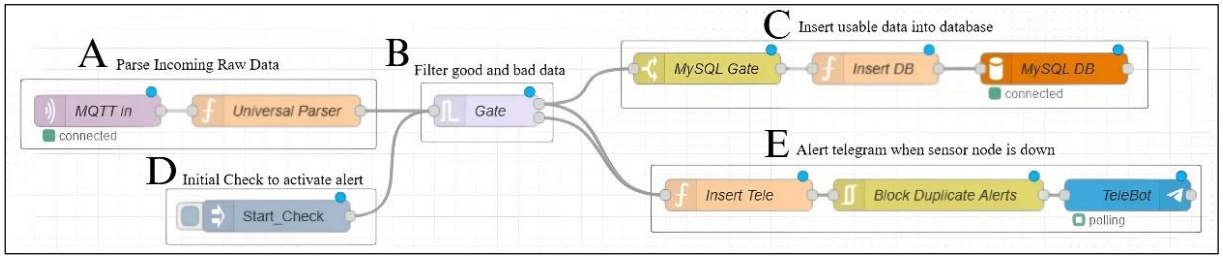

Fig. 5 Example of a Node-RED system flow for data extraction from TTN with alert node flow

Lastly, due to the presence of a dashboard, one should utilize the System Usability Scale (SUS). This is an instrument to quantify the usability of software and hardware products. Developed by John Brooke in 1984 to provide a tool for measuring the usability of various systems. As seen the usage of SUS to scale for digital health apps (DHAs) in one of the recent studies by researchers, proving their continued reliability into our current modern era [4]. The scale itself consists of a 10-item list utilizing Likert-scale, where 5 indicates the users strongly agree while a 1 likely indicates the users or respondents strongly disagree with the item's statement. To help understand the impact and interpretation of the scale, one can see it as getting a average SUS score of 68 or more would quantify as an average score, however recent studies indicate that due to current saturated market, an average score of 80 would quantify as average instead [5]. This will provide a clear guideline on what our dashboard score should strive for. To calculate the score, one must aggregate the individual scores from the ten items. As the items consist of alternating positive and negative statements, the raw score can be calculate with the following formula:

$$X_n = R_n - 1 \tag{1}$$

$$X_n = 5 - R_n \tag{2}$$

$$Final\ SUS\ Score = (\sum_{n=1}^{10} X_n) \times 2.5 \tag{3}$$

For Odd-Numbered Items (1,3,5,7,9), as these are positive statements, the score is calculated by subtracting 1 from the user's response (R) as uses the equation (1). As for Even-Numbered Items (2,4,6,8,10), as they are negative statements, the score instead is calculated by subtracting the user's response (R) from 5 and uses equation (2). Lastly the final SUS Score is the sum of X for all items will then be multiplied with 2.5 to normalize the final score of range 0-100. Utilizing these formulas, the final SUS score can be calculated.

3.0 Result and Discussion

First, as the project revolves around creating a dashboard for multiple sensors parameters, a sensor node was developed to test the connectivity and data transferability from the sensor node itself until the dashboard. Figure 6 is the developed hardware, utilizing a junction box as the casing, and containing the Heltec WiFi LoRa 32(V3.1) microcontroller, PT1000 temperature sensor, pH and Total Dissolved Solids sensor from DFRobot.

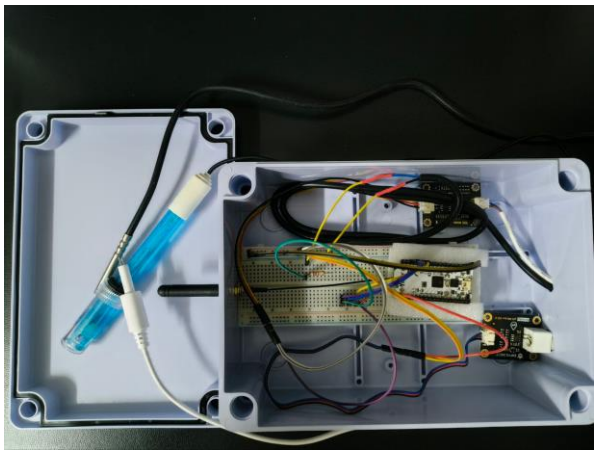

Fig. 6 Developed hardware

Utilizing this sensor node, the data transmission was tested, starting from Arduino IDE applications. Connecting the sensor node to a power supply, the sensor was activated and the sensor parameters were recorded on the Arduino IDE serial monitor. Note that the sensor was calibrated with thermometers, and buffer solutions. The data obtained in Figure 7 is from a soft tap water that is slightly warmed.

```
14:50:54.006 -> [SEND] Sending data...
14:50:54.221 -> Temperature: 40.54°C, pH: 6.58, TDS: 244.39 ppm
14:50:54.221 -> confirmed uplink sending ...
14:50:54.254 -> TX on freq 923400000 Hz DR 5 power 13
14:50:54.254 -> [CYCLE] Scheduling next transmission...
```

Fig. 7 Sensor readings in Arduino IDE serial monitor

Next, a gateway was set up to allow the microcontroller to transmit the payload to the TTN live application. As seen on Figure 6, the sensor data was successfully transmitted and viewable on the TTN live application. This will then allow the utilization of MQTT protocol to subscribe to the TTN live data via Node-RED. Figure 7 thus shows the plausibility of retrieving payload from TTN. After that comes the storage system of the project, MySQL. Since the data had been retrieved by Node-RED, the next step would be to store that data in MySQL for historical keeping. Figure 8 shows that the data had been successfully stored in MySQL.

Fig. 8 TTN Live application shows successful data transmission from sensor node

Progress in Engineering Application and Technology Vol. 7 No. 1 (2026) p. 380-388

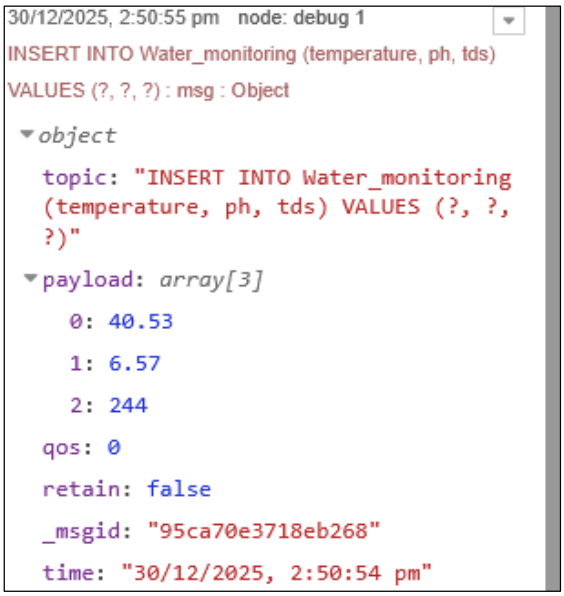

Fig. 9 Node-RED debug window shows successful data retrieval from TTN utilizing MQTT protocol

id	temperature	ph	tds	times
16	28.68	7.66	547	2025-12-29 21:54:53
15	29.8	7.61	552	2025-12-29 21:52:23
14	29.64	7.62	552	2025-12-29 21:51:14
13	28.53	7.62	551	2025-12-29 21:50:05
12	28.99	7.62	552	2025-12-29 21:49:37
11	28.6	7.62	552	2025-12-29 21:49:19
10	28.71	7.61	551	2025-12-29 21:48:44
9	28.73	7.61	552	2025-12-29 21:48:19
8	28.86	7.6	552	2025-12-29 21:47:52
7	29.05	7.59	552	2025-12-29 21:47:26

Fig. 10 MySQL database successfully storing the sensor data

Lastly, the dashboard must then be successfully displaying the data stored in MySQL. Utilizing Streamlit, the dashboard was created in Python language and Figure 9 shows that the data from PT1000, pH and TDS sensor can indeed be viewed on the dashboard. The dashboard is equipped with graphs, interactable slider and pull-down tabs. Having the parameters displayed using a graph aid in recognizing trends in the water parameters, allowing one to gauge and detect abnormalities. This will then help to prompt more immediate actions to uncover the reason for any unusual spikes in the parameters monitored.

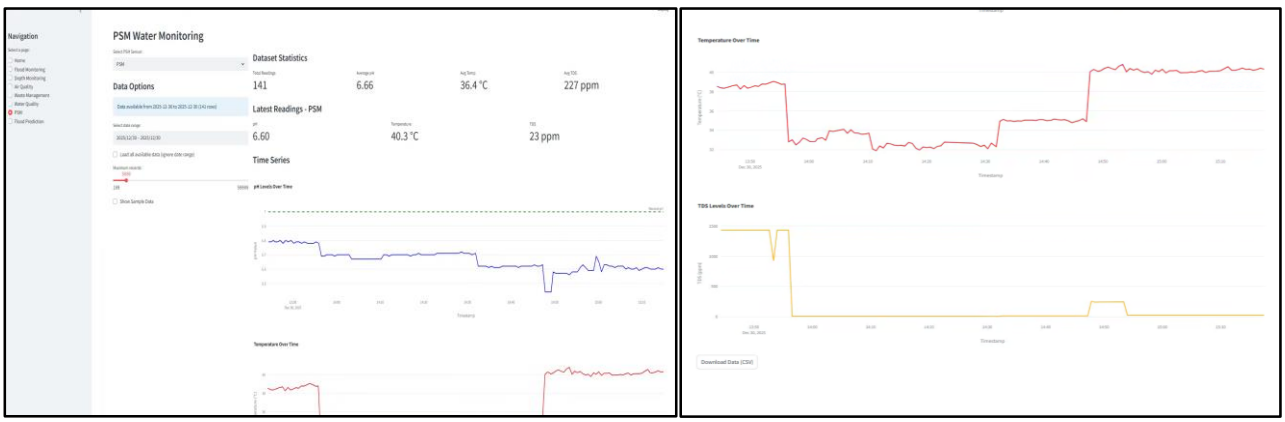

Fig. 11 Dashboard showcase (a) Data displayed successfully; (b) More of the sensor parameters from sensor node

Lastly, a simple Telegram alert system was also implemented to detect sensor abnormalities when placed in remote locations. An allocated time of 60 seconds will be given to each sensor's node, where if they do not transmit any data in that allocated time, they will be flag as disconnected by Node-RED. Thus, via Node-RED, a output will be created and a pre-generated message will be sent to the Telegram Group via a Telegram Bot. This means anyone in the group will be notified, thus allowing prompt action to fix the sensor.

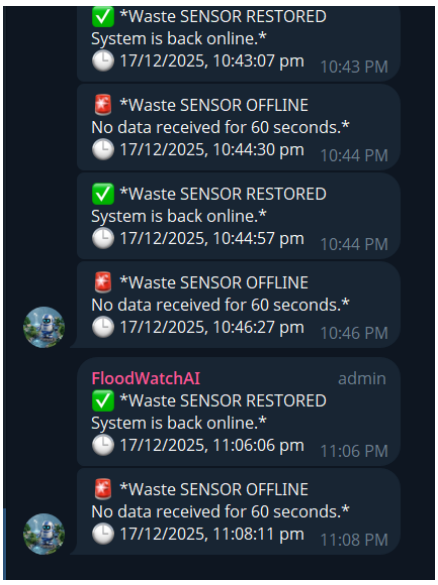

Fig. 12 Telegram Alert System

To evaluate the user experience and interface effectiveness, a System Usability Scale (SUS) survey was conducted with 100 respondents. The survey was structured into four distinct sections to ensure comprehensive feedback:

- Section A: Provided respondents with direct access to the dashboard, ensuring hands-on interaction before evaluation.
- Section B: Collected demographic data, including gender, occupation, age, and tech-savviness, to identify specific usage trends.
- Section C: Contained the standard 10 SUS items using a 1-5 Likert scale, alternating between positive (odd numbered) and negative (even numbered) statements.
- Section D: Gathered qualitative feedback and personal opinions regarding potential system improvements.

The SUS results reveal discrepancy between functionality and user perception or experience. Based on Figure 13, while positive statements exceed the 2.5 average mark, the respondents are also generally in high agreement with the negative statements.

The calculated SUS score was 56.25 as seen in Figure 14, which in industry standards will fall below the average mean of 68. Furthermore, when measured against a modern benchmark, it is far from the average mean of 80 which generally will be considered as acceptable. This score indicates that the dashboard system still falls below the required standard for modern web applications.

The system seems to be successful as it achieved its technical objectives, but the lower rating can be contributed to the aesthetic of the overall dashboard. Qualitative feedback from Section D highlights specific need for increase in icon variety, refined colour palettes and overall better visual representations. This can be seen from Figure 15, a snippet of the few responses chosen from the survey. The survey overall suggests that the dashboard system is functional and user friendly in terms of data presentation, but the visual representations is what drags its score down, when compared to other dashboards it can be quite functionality-heavy and not very graphic, as would be the main gripe of a typical everyday user. Thus, showcasing that to create a dashboard, one should also heavily consider the overall aesthetic of the dashboard to first grab the attention of the everyday user, leading to more frequent visits and usage of the dashboard.

Progress in Engineering Application and Technology Vol. 7 No. 1 (2026) p. 380-388

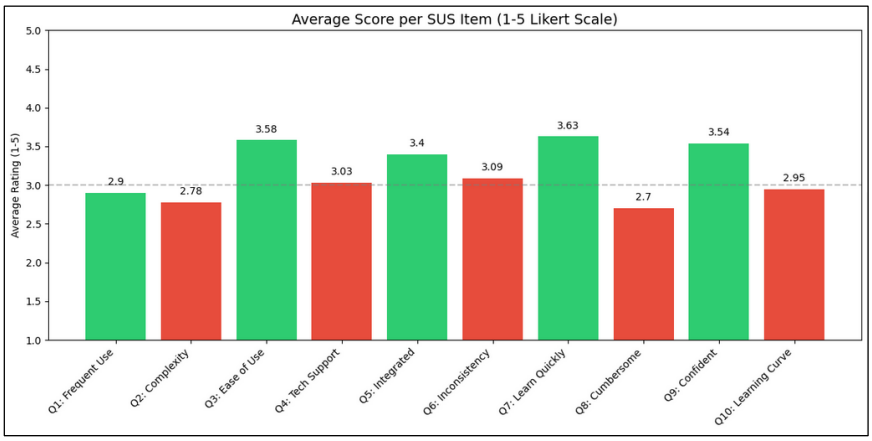

Fig. 13 Average score per SUS item in Likert Scale

```python
import pandas as pd

# 1. Load the data
file_path = '/content/drive/MyDrive/PSM/SUS.csv'
df = pd.read_csv(file_path)

# 2. Select columns G to P (indices 6 to 15) and first 100 rows
# This ensures we are only using the 10 SUS questions
sus_data = df.iloc[:100, 6:16]

def calculate_sus(row):
    """
    Standard SUS Calculation:
    - Odd items (1, 3, 5, 7, 9): score = response - 1
    - Even items (2, 4, 6, 8, 10): score = 5 - response
    - Final Score = Sum * 2.5
    """
    score = 0
    for i in range(10):
        val = row.iloc[i]
        if (i + 1) % 2 != 0:  # Question 1, 3, 5, 7, 9
            score += (val - 1)
        else:                 # Question 2, 4, 6, 8, 10
            score += (5 - val)
    return score * 2.5

# 3. Apply calculation and save results
df['SUS Score'] = sus_data.apply(calculate_sus, axis=1)

# Summary of results
average_score = df['SUS Score'].mean()
print(f"Average SUS Score: {average_score:.2f}")

# Optional: Save results to a new file
df[['Timestamp', 'SUS Score']].to_csv('SUS_Scores_Results.csv', index=False)
```

```
Average SUS Score: 56.25
```

Fig. 14 Average SUS score obtained

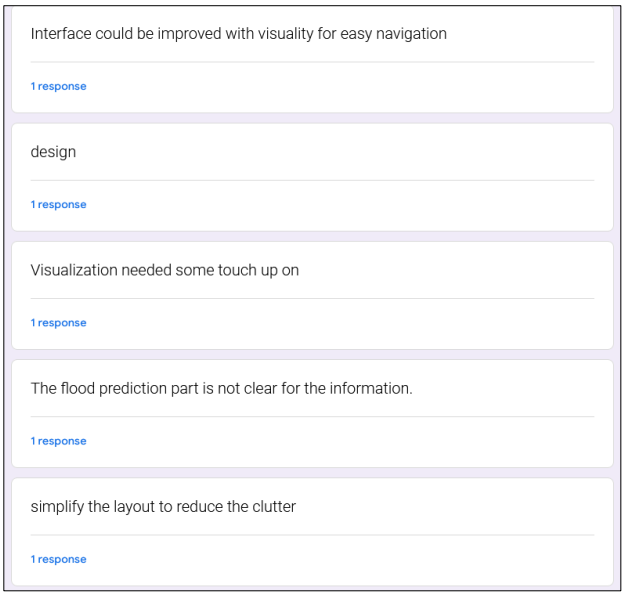

Fig. 15 Some of the responses in Section D for dashboard improvement

4.0 Conclusion

The aim of the project is to create a dashboard that can seamlessly integrated into both edge and cloud applications and integrated multiple sensors into the overall design. The integration of multiple sensor node into the system and displaying the necessary parameters onto the dashboard was also a success. The overall system is very usable, and it can be said that the project has been successful. The project showcases the functionality and feasibility of utilizing edge computing to handle localized data processing and storage whilst utilizing cloud infrastructure for remote accessibility. Users can access the dashboard at any time if they are connected to the Internet and the developers can continue to maintain system and due to processing data in edge devices, it allows for lower latency, allowing quick update to the dashboard. However, the usability grade of the overall dashboard seems to fall below the average industry standard, being at 56.25, which is lower than the average SUS score of 68 or the even higher modern standards of 80 score. From the google form survey, the respondents agree that while the dashboard itself is easy to use and does not necessarily require assistance when navigating it, it loses out a lot in terms of its ability to capture the respondent attention in terms of visual presentations. Requiring more icons, widgets, colors and overall better visual standards will be the first step in improving the dashboard SUS score. This means that at the current moment, the dashboard is more usability and functionality focused dashboard, which is a good start, but a lot of improvement can be made to further improve the dashboard.

Acknowledgement

The author would like to thank the Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia for their support.

References

[1] J. T. Overpeck and B. Udall, "Climate change and the aridification of North America," Proceedings of the National Academy of Sciences, vol. 117, no. 22, pp. 11856-11858, 2020.

[2] P. Grenni, V. Ancona and A. B. Caracciolo, "Ecological effects of antibiotics on natural ecosystems: A review," *Microchemical Journal,* vol. 136, pp. 25-39, 2018.

[3] A. A. Cook, G. Misrili and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet of Things Journal,* vol. 7, no. 7, pp. 6481--6494, 2019.

[4] M. Hyzy, R. Bond, M. Mulvenna, L. Bai, A. Dix, S. Leigh and S. Hunt, "System usability scale benchmarking for digital health apps: meta-analysis," *JMIR mHealth and uHealth,* vol. 10, no. 8, p. 37290, 2022.

[5] J. Sauro and J. R. Lewis, "Quantifying the user experience: Practical statistics for user research," Cambridge, MA, Morgan Kaufman, 2024.