IJMIC Vol. 3, Issue 1, 90 - 108

Research Article

Smart IoT-Based Water Tank Monitoring Using ESP32 and Mobile Application

Douglas Anak Gurang[1], Gohar Rahman[1]*, Mehmood Ahmed[2]

[1]Faculty of Computing and Informatics, Universiti Malaysia Sabah, Sabah, Malaysia

[2]Department of Information Technology, The University of Haripur, Khyber-Pakhtunkhwa, Pakistan

*Corresponding author: Gohar Rahman, gohar_315@ums.edu.my

Received: [July 27, 2026]
Accepted: [September 1, 2026]
Published: [September 4, 2026]

Abstract – In rural, institutional or residential areas, water shortage and ineffective water tank monitoring continue to be major problems leading to unplanned water outages and ineffective water resource management. Conventional monitoring techniques involve manual inspections which are time-consuming, prone to human error and cannot offer real-time information. Therefore, the aim of this research is to create a Smart IoT-Based Water Tank Monitoring System that enables real-time monitoring of water level and TDS-based purity indicators via a mobile application. An ESP32 microcontroller, a waterproof ultrasonic sensor (JSN-SR04T) to measure water level, a Total Dissolved Solids (TDS) sensor to monitor water quality and a DHT11 sensor to measure ambient temperature and humidity are all integrated into the proposed system. Users can remotely access real-time information and receive low water level alarms by using Firebase Real-Time Database as a cloud platform to synchronize sensor data with a mobile application created with Android Studio. A prototype model is used to create a system that facilitates design, implementation, testing and iterative improvements throughout the development process. Meanwhile, functional and real-world tests were conducted to evaluate the sensor accuracy, data transmission reliability, application responsiveness, and system stability under various operating situations. According to the experimental findings, the proposed system effectively tracks water levels and TDS-purity parameters in real time, ensures reliable communication between the hardware and the cloud database, and promptly notifies users when water levels exceed pre-defined criteria. In addition to improving monitoring effectiveness and reducing manual intervention, the created system provides a feasible and affordable alternative for smart water resource management. To accommodate more smart water management applications, further research can improve the system by adding predictive analytics, historical data visualization, and other water quality criteria.

Keywords: Internet of Things (IoT), Water Tank Monitoring, ESP32, Firebase Realtime Database, Mobile Application

1. INTRODUCTION

Due to fast progress in the sphere of Internet of Things (IoT), monitoring systems have been upgraded to more advanced technologies allowing for collecting data in real time and communicating and controlling remotely. By the combination of sensors, microcontrollers, wireless communication, and cloud computing, IoT allows constant monitoring and making decisions automatically in different spheres, such as smart home, agriculture, health care, environmental monitoring, and management of infrastructures [1], [2], [16]. In spite of the variety of areas IoT can be used in, water resource management is among those that became highly studied because of the need for the effective monitoring systems ensuring water consumption sustainability.

Water is one of the most vital resources needed by domestic and industrial sectors. Residential houses, university hostels, rural areas, and commercial buildings use water tanks to provide for the constant flow of water. At the same time, many water tank systems still use manual inspection to measure the amount of water left in the tank. Such a system is not only ineffective but time-consuming and vulnerable to human errors since users do not always notice that the level of water is low enough before the tank empties [3], [17].

Recent advances in IoT have encouraged researchers to develop smart water monitoring systems capable of providing real-time information through cloud platforms and mobile applications. Several studies have demonstrated that IoT-based monitoring systems improve operational efficiency, reduce manual intervention, and enable users to remotely access water level information using smartphones [4], [5]. Furthermore, cloud-based platforms such as Firebase allow sensor data to be synchronized instantly with mobile applications, improving accessibility and supporting timely decision-making [6].

Although numerous IoT-based water monitoring systems have been reported, many existing solutions primarily focus on water level detection while offering limited integration with additional environmental parameters, user-friendly mobile interfaces, and real-time notification features [7]. Some systems also lack comprehensive performance evaluation under different operating conditions, including sensor accuracy, communication reliability, and application responsiveness [8]. These limitations indicate the need for a more integrated and practical monitoring solution that combines accurate sensing, reliable cloud communication, and an intuitive mobile application.

Consequently, this research proposes an IoT-based Smart Water Tank Monitoring System by implementing an ESP32 microcontroller, JSN-SR04T waterproof ultrasonic sensor, Total Dissolved Solids (TDS) sensor, and DHT11 sensor in conjunction with Firebase Realtime Database and Android-based mobile application. The ultrasonic sensor is used for continuous measurement of the water level, while the TDS and DHT11 sensors evaluate TDS-purity parameters and the environment respectively. All gathered information is transmitted via Wi-Fi network to Firebase, allowing users to observe the water condition remotely through mobile application and receive alerts whenever the water level gets to the required limit.

The obtained data is sent through the Wi-Fi connection to Firebase where users can monitor the water condition in real time using a mobile app and also get alerts when the water level reaches a required point. The main benefit of the work done here is the complete integration of all real-time ultrasonic water-level detection, TDS water condition monitoring, and DHT11 environmental detection in one microcontroller with a Firebase cloud and a push-notification Android mobile app. Unlike other works that use multi-unit processing or have no active mobile alert systems, this integrated prototype is cost-efficient, and plug-and-play tested under household conditions.

This system was developed based on the Prototyping Model to enable iterative designing, implementation, testing, and refinement of the system. Functional testing as well as experimental evaluation was carried out to test accuracy of sensors, synchronization of the cloud, responsiveness of the mobile application, and reliability of the entire system. The developed prototype shows that the system under consideration is not only economical but also feasible and scalable IoT-based solution for smart water tank monitoring contributing to more effective water resource management and Sustainable Development Goal 6 (Clean Water and Sanitation).

2. LITERATURE REVIEW

2.1 IoT-Based Water Tank Monitoring Systems

Water tank monitoring by means of Internet of Things (IoT) technology has been widely discussed because of its capabilities to offer real-time data collection, remote accessibility, and automated decision making. Traditional methods of water tank monitoring are usually associated with manual checking, which is inefficient and prone to mistakes. This issue prompted many scientists to suggest IoT-based monitoring systems, which involved microcontrollers, sensors, wireless technologies, and cloud computing in order to increase the efficiency of the process.

For example, Sharma et al. suggested developing an intelligent water management system based on an Arduino device and NodeMCU ESP8266 microcontroller used in conjunction with ultrasonic and soil moisture sensors for water level monitoring and irrigation purposes [9]. It is worth mentioning that in their study, the authors suggested implementing an automatic pump control process based on predetermined threshold values; also, people could check the system status via the Internet. It was proved that automation helped reduce water waste and optimize water resource usage. Nevertheless, the system was aimed at irrigation only, without water quality monitoring and mobile app development.

In a similar approach, Livinti et al. also suggested a water level control and monitoring system based on Arduino Uno and NodeMCU ESP8266 development boards [10]. In this design, Arduino Uno was responsible for sensor interfacing and pumping operations, while NodeMCU was used for the communication process with Thing Speak cloud service. In this way, the authors were able to achieve an effective implementation of automatic pump operation and monitoring of the water levels. However, two different controllers made the hardware implementation more complex and expensive, while the monitoring process was limited only to water levels.

The current trends in the development of the IoT technology are characterized by an increasing number of ESP32 microcontrollers utilization due to their integrated Wi-Fi function, better performance, and simplified hardware structure.

2.2 IoT-Based Water Quality Monitoring Systems

In addition to the measurement of water levels, some studies have used IoT technology to assess the water quality through constant measurement of the physical and chemical properties of water. These technologies are able to detect contamination in water at an earlier stage while allowing users to monitor the water remotely.

Mahendiran et al. created an IoT water quality monitoring system by employing the ESP32 microcontroller which was equipped with pH sensor, turbidity sensor, TDS sensor, and temperature sensor [11]. Data from the sensors were sent to an online portal, allowing the users to monitor the condition of the water quality. It was shown that IoT technology is able to minimize manual sampling greatly and enhance the monitoring process. The IoT system proposed by Mahendiran et al. did not cover any water level monitoring or alerting system.

Jafri et al. also proposed a real-time river water quality monitoring system based on IoT which used a combination of sensors along with the ESP32 microcontroller to monitor pH level, temperature, turbidity, and TDS in the Selangor River [12]. The data gathered by the system were sent via Wi-Fi and then stored in a cloud platform where the results could be visualized and analyzed. However, while the system proved to work reliably in terms of environmental monitoring, it was mainly meant for river monitoring and did not have a mobile-based notification system for domestic water storage.

IoT technology has found numerous applications in the field of flood early warning systems too. For example, Adan and Alias proposed an ESP32-based early warning system that employed ultrasonic sensors for measuring increasing water levels and transmitting the data via the Blynk platform [13].

2.3 Research Gap

Despite the availability of many water monitoring systems based on the Internet of Things, there are still some limitations. Previous research pay more attention either on the development of water level monitoring [9], [10] or water quality monitoring [11], [12] but do not integrate these two features in one system. Also, many available systems use more than one microcontroller, which increases the difficulty of its implementation and its cost [10], [18]. Moreover, some systems offer cloud-based visualization without a dedicated mobile application providing real-time notifications [11], [12].

One more common limitation is that there is a lack of evaluations of the system performance in practical conditions. Prototype systems evaluate only the basic functioning of the system without any verification of the accuracy of sensors, communication effectiveness, application responsiveness, and operational stability [13], [19],[20].

In order to solve this issue, the research offers development of Smart IoT-Based Water Tank Monitoring System, which incorporates ESP32 microcontroller, JSN-SR04T ultrasonic sensor, TDS sensor, and DHT11 sensor with Firebase Realtime Database and Android mobile application. The difference from previous research is that the offered system combines all of the above-mentioned functions within one IoT system. In addition, the functional and experimental tests were conducted for evaluation of its performance.

3. METHODOLOGY

This study adopts a structured system development methodology for the design and implementation of a Smart IoT-Based Water Tank Monitoring System. The system is developed using the Prototyping Model, which is widely used in IoT and embedded system development due to its iterative nature and ability to refine system requirements through continuous testing and feedback [1], [14].

3.1 Prototyping Model

The Prototyping Model is an iterative software and hardware development approach in which an initial system prototype is developed, tested, evaluated, and progressively improved until the final system meets the desired requirements [2], [15]. This model is particularly suitable for IoT-based applications where hardware-software integration must be validated continuously during development.

In this project, the Prototyping Model is selected because the system integrates multiple components, including an ESP32 microcontroller, ultrasonic sensor, TDS sensor, DHT11 sensor, cloud database, and a mobile application. These components require early-stage integration testing to ensure reliable real-time performance and data synchronization [3].

The use of prototyping also reduces development risks by identifying errors in sensor readings, communication delays, and application synchronization at an early stage. Furthermore, it improves system usability by allowing iterative refinement of the mobile interface based on system behavior and performance evaluation [4].

Overall, the Prototyping Model ensures that the system evolves through continuous refinement, resulting in a stable, efficient, and real-time water monitoring solution. The Prototyping Model also provides an effective development framework for IoT-based monitoring systems due to its iterative improvement process and flexibility in integrating hardware and software components. Through continuous testing and refinement, the proposed system achieves reliable real-time water level monitoring with efficient cloud-based data synchronization.

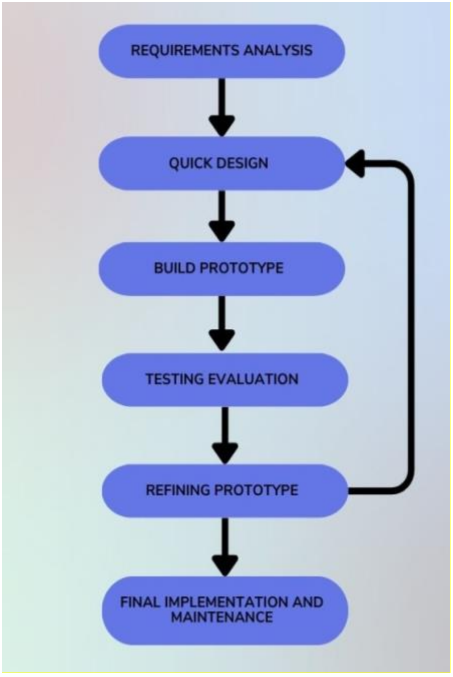

Figure 1: Prototyping Development Model

Figure 1 The development lifecycle of the Smart IoT-Based Water Tank Monitoring System is guided by the Prototyping Model. Unlike traditional, rigid frameworks such as the Waterfall model, this iterative approach provides the flexibility required for Internet of Things (IoT) projects, where hardware components, cloud telemetry, and mobile software must interface simultaneously. Once the data loops achieved absolute stability with zero transmission faults, the entire hardware and software framework was finalized and prepared for long-term domestic operation. Each phase contributes to the continuous improvement of both hardware and software components until the proposed system achieves the desired functionality and performance [1]. The iterative nature of the model enables early detection of hardware integration issues, sensor inaccuracies, and communication errors, thereby reducing development risks and improving overall system reliability.

3.2 Prototype Development Phases

1) Requirement Analysis

The practical boundaries of the project were established first. This involved selecting the core hardware components, focusing on the ESP32 microcontroller as the central processing unit, the waterproof JSN-SR04T ultrasonic sensor for depth measurement, and the TDS sensor to evaluate water clarity. The requirement analysis phase focuses on identifying system functional requirements for real-time water monitoring. The main requirement is accurate water level detection using an ultrasonic sensor, combined with stable wireless communication via ESP32 for real-time data transmission to a cloud database [5].

2) Quick Design

System logic was mapped out by organizing data flow paths, structuring the JSON database nodes inside the Firebase Realtime Database, and drafting the user interface wireframes for the mobile application. In this phase, the system architecture is designed, including the data flow between sensors, microcontroller, Firebase database, and mobile application. The mobile interface layout is also designed to ensure clear visualization of water level status and system alerts.

3) Build Prototype

The hardware configurations were physically assembled on a breadboard using an experimental, scaled water container jug to simulate a real-world tank. Concurrently, core firmware scripts were compiled via the Arduino IDE, and the mobile application interface was built in Android Studio. A working prototype is developed using a small-scale water tank model. The system integrates:

a) Water level monitoring module (ultrasonic sensor)

b) Data processing module (ESP32)

c) IoT communication module (Wi-Fi transmission)

d) Mobile application module (real-time display)

e) Cloud synchronization module (Firebase integration)

4) Testing Evaluation

The active prototype was subjected to targeted field evaluations. These tests included physical tank filling and draining cycles to verify depth accuracy, sensor submersions across distinct water samples (tap, mineral, and saltwater), and intentional Wi-Fi disconnections to evaluate cloud database reconnection loops. The prototype is tested based on sensor accuracy and data transmission performance. The ultrasonic sensor readings are validated against actual water levels, while system latency is measured to ensure real-time updates between hardware and mobile application [6].

5) Refining Prototype

This phase served as a systematic optimization gateway. When testing highlighted performance anomalies-such as ultrasonic signal noise caused by water surface ripples or database synchronization delays-the lifecycle utilized the feedback loop to return to the Quick Design stage to adjust the underlying software logic. Improvements are carried out based on test results to ensure that the system is accurate, stable, and synchronized. This process is repeated until the system becomes reliable.

6) Final Implementation and Maintenance

Once the data loops achieved absolute stability with zero transmission faults, the entire hardware and software framework was finalized and prepared for long-term domestic operation. Improvements are carried out based on test results to ensure that the system is accurate, stable, and synchronized. This process is repeated until the system becomes reliable.

Table 1. Prototype Development Phases

Phase	Description
Requirement Analysis	Select ESP32, JSN-SR04T, TDS, and DHT11 sensors based on monitoring needs.
Quick Design	Map hardware pins, Firebase JSON paths, and mobile UI wireframes.
Build Prototype	Assemble components on breadboard and program via Arduino IDE and Android Studio.
Testing & Evaluation	Verify sensor accuracy, fluid level shifts, and Wi-Fi reconnection handling.
Refining Prototype	Use the feedback loop to adjust design settings if any errors are found.
Final Implementation and Maintenance	Finalize the stable hardware-to-cloud setup for regular domestic deployment.

The following Table 1 depicts the five stages associated with the Prototyping Model approach. An iterative design process enables the proposed system to be continually tested and modified for ensuring proper connectivity between ESP32, sensors, Firebase Realtime Database, and the mobile application.

3.3 Tools and Technologies

Embedded systems and IoT technology are implemented in order to enable real-time monitoring of the system. In particular, ESP32 microcontroller acts as the main processing unit and the sensors used include the JSN-SR04T ultrasonic sensor, TDS sensor and DHT11 sensor.

In developing the software for this project, the tools used include the Arduino IDE for firmware development, Android Studio for the mobile application and Firebase Realtime Database for cloud-based data storage and synchronization. Other software tools used in this project include Fritzing, Draw.io, and Canva.

Table 2. Hardware and Software Used

Category	Components
Hardware	ESP32 DEVKIT V1, JSN-SR04T Ultrasonic Sensor, TDS Sensor, DHT11 Sensor
Software	Arduino IDE, Android Studio, Firebase Realtime Database, Fritzing, Draw.io

Table 2 shows the hardware and software components used in this research. The ESP32 will act as the main microcontroller to process sensor readings, whereas Firebase will be used as the cloud data synchronizer. For the development of the mobile app, Android Studio will be used, whereas Arduino IDE will program the ESP32 firmware.

3.4 System Integration Approach

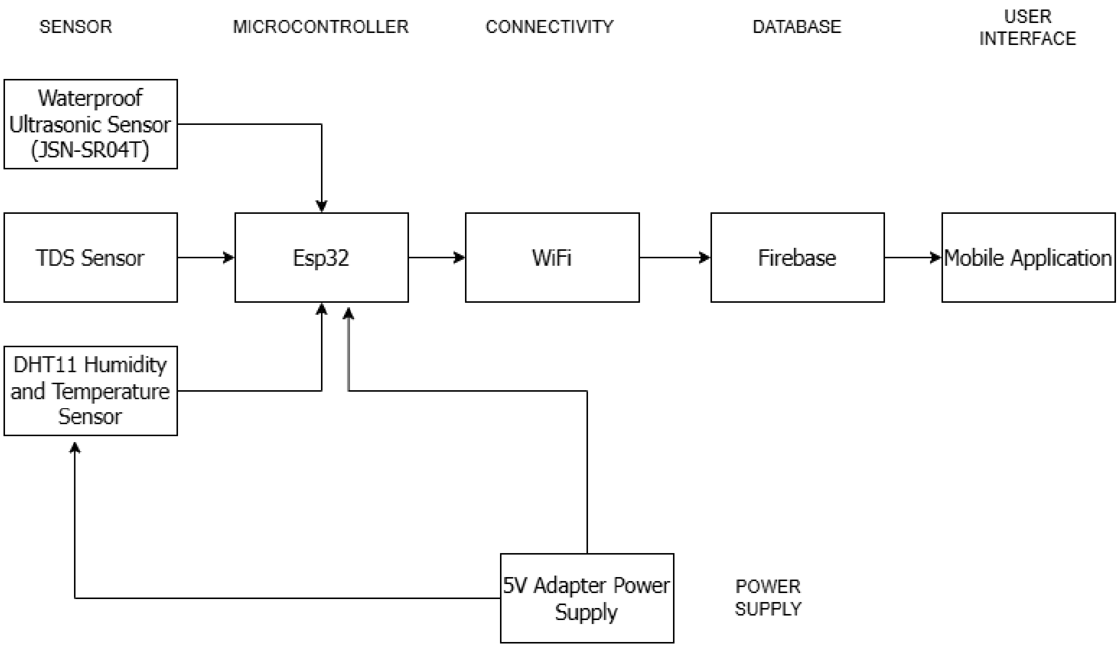

Figure 2: Block Diagram of Overall System Architecture

Figure 2 is used to describe the whole architecture of the proposed Smart IoT-Based Water Tank Monitoring System. Data that are generated from the sensors like the JSN-SR04T ultrasonic sensor, TDS sensor, and DHT11 sensor undergo processing by the ESP32 microcontroller and are finally uploaded to the Firebase Realtime Database using Wi-Fi. The synchronized data are presented on the Android mobile application. The architecture of the cloud-based system enhances constant synchronization of data between the physical system and the mobile application.

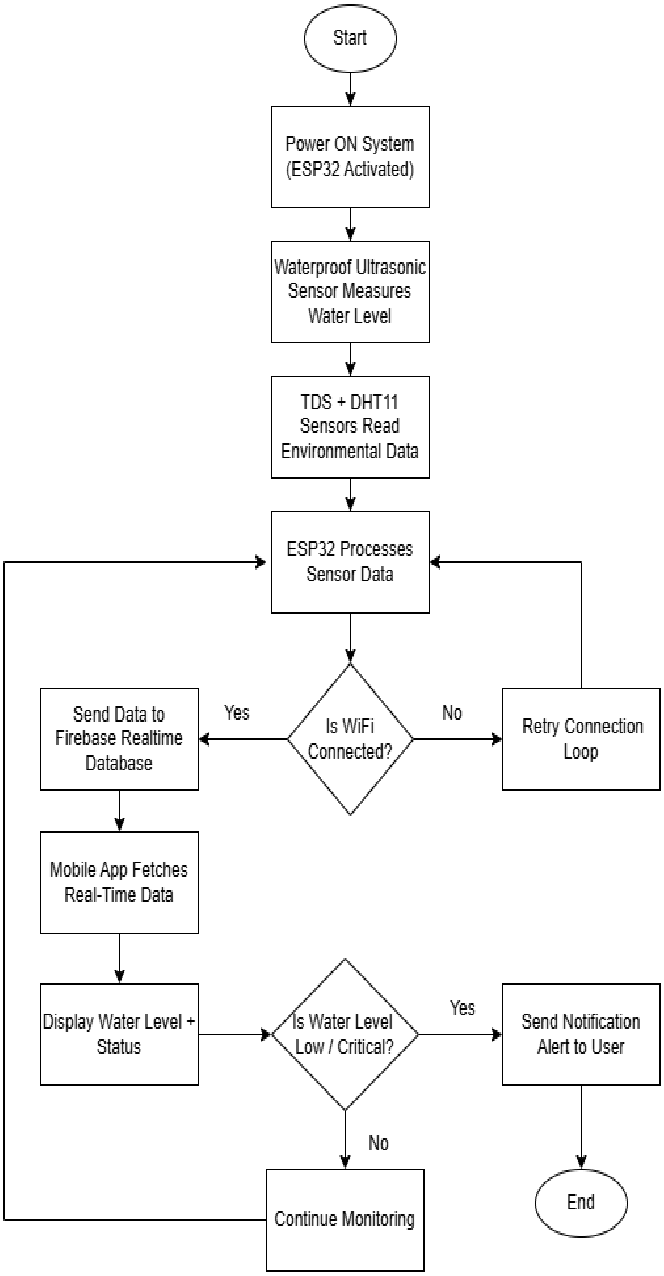

Figure 3: System Functional Workflow and Data Logic Loop

The system implements IoT architecture which involves collecting data from sensors, processing it with ESP32 microcontroller, transferring via Wi-Fi connection, storing data in Firebase cloud database, and displaying the data on mobile app.

Water level is measured by ultrasonic sensors whereas the quality and environment parameters of the water are measured by TDS and DHT11 sensors respectively. All the processed data is synchronized in real-time through Firebase.

4. SYSTEM DESIGN AND IMPLEMENTATION

4.1 Prototype Implementation

The proposed design of the Smart IoT-Based Water Tank Monitoring System is depicted in Figure 4. In this case, the design includes ESP32 DEVKIT V1 with an ultrasonic sensor, TDS sensor, and DHT11 sensor in order to conduct the monitoring of water level, water quality, temperature, and humidity in real time. The sensor data is processed by ESP32 and then sent to the Firebase Realtime Database via Wi-Fi connection. The implementation of the design was done by using a small-scale model of the water tank, which simulates real-life conditions for testing purposes.

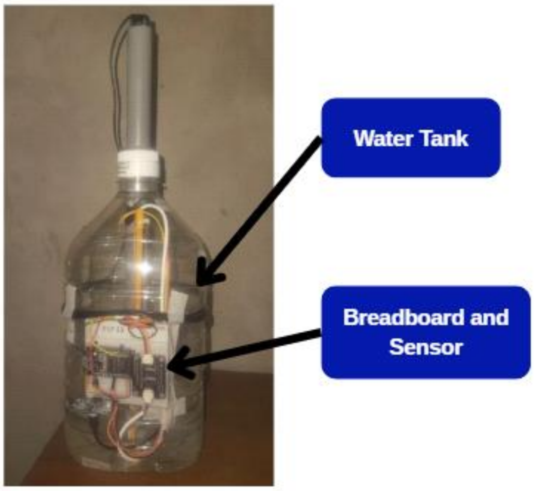

Figure 4: Developed Smart IoT-Based Water Tank Monitoring System Prototype

4.2 System Implementation

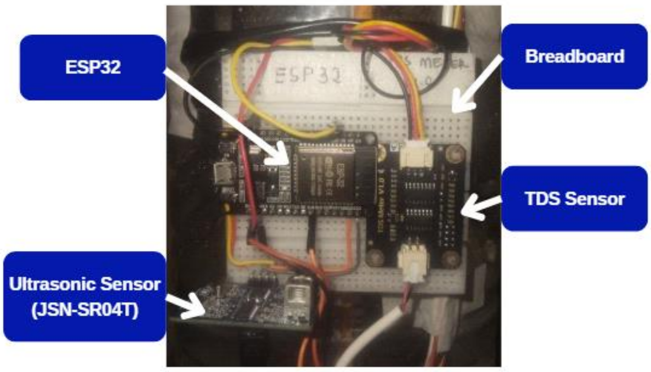

Figure 5: Hardware Assembly and Circuit Integration

Figure 5 shows the hardware implementation of the proposed system. The sensors were connected to the ESP32 through jumper wires and breadboard in accordance with the designed circuit diagram. The waterproof ultrasonic sensor was placed above the tank for measuring the level of water, whereas the TDS sensor and the DHT11 sensor were placed in accordance with their respective functionalities.

The hardware setup facilitated the steady communication between the sensors and the ESP32 to continuously acquire the monitoring data.

4.3 Mobile Application

Figure 6: Mobile Application Interfaces

Dashboard offers real-time visualizations of water level, water quality, temperature, and humidity data retrieved from the Firebase Realtime Database. On the other hand, analytics and sensor trends pages offer users the option of analyzing sensor performance history, while the monitoring history maintains all monitoring history records. There is also a TDS parameters baseline guide included in the application.

4.4. Experimental Results

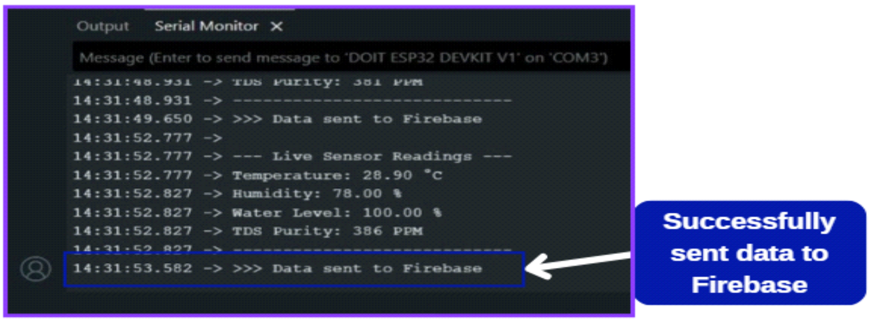

Figure 7: Real-Time Sensor Output in Arduino IDE Serial Monitor

Figure 7 illustrates real-time sensor readings shown on the Arduino IDE Serial Monitor during testing of the system. The use of the Serial Monitor helped developers debug and verify whether the ESP32 could receive data from the JSN-SR04T ultrasonic sensor, TDS sensor, and DHT11 sensor before sending the data to the Firebase Realtime Database.

The sensor readings shown include the water level, TDS value, temperature, humidity, and system connectivity, including the connection to Wi-Fi and Firebase databases. The monitoring of the sensor readings through the Serial Monitor helps developers verify sensor readings, data processing, and data transmission.

The fact that the continuous sensor readings were successfully displayed shows that the ESP32 was able to receive and send data for monitoring. Thus, the hardware and software components were working as expected before connecting to the cloud and the mobile application.

Table 3. Initial Functional and Core Component Checkout Results

Test Item	Expected Result	Status
Water Level Detection	Accurate reading	✓
TDS Measurement	Real-time reading	✓
Temperature Measurement	Accurate reading	✓
Humidity Measurement	Accurate reading	✓
Firebase Synchronization	Successful	✓
Mobile Application Update	Real-time	✓
Notification Alert	Triggered below threshold	✓

Table 3 shows the functionality test carried out on the developed system. All functionalities worked properly during the testing procedure. The sensors' values were correctly retrieved and sent to the Firebase database, while the mobile app was able to present the monitoring data in real-time mode.

4.5. System Performance

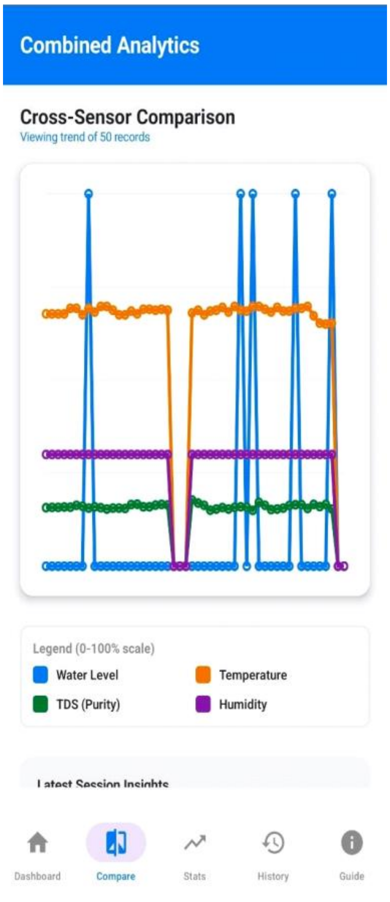

Figure 8: Combined Analytics of Real-Time Sensor Readings

The analytics graph illustrated in Figure 8 below is a graph created by the application incorporating all the data points gathered from different sensors into a single graph. The analytics graph includes the reading of the water level, TDS level, temperature, and humidity. The users can be able to observe how the different variables of the environment and physics change in the graph.

The integration of the graphs allows the user to see the trends, abnormalities, and sudden changes in the behavior of the sensors when comparing to seeing each variable separately. The changes in the water level can be easily compared with the water quality and the environment to find if there is any correlation while the system is running.

Generally, the combined analytics graph shows the capacity of the proposed system in gathering, synchronizing, and visualizing different data points.

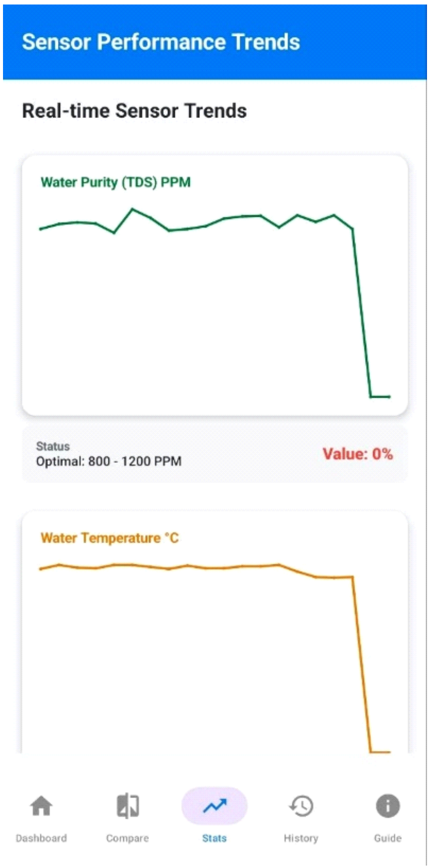

Figure 9: Sensor Performance Trends

The performance trends of each individual sensor are shown in Figure 9. Unlike the overall analytics graph presented above, this interface enables the user to analyze the individual sensor parameters independently in terms of water level, TDS, temperature, or humidity over the specified period.

With the help of the historical trend presentation, users are able to analyze the behavior of sensors and compare the obtained results with the previously set thresholds for sensors. Consistent sensor readings mean that the system is working properly, while any unusual readings can help users detect problems with sensors.

Moreover, the presented individual trend analysis offers useful historical data for the purpose of preventive maintenance and system evaluation. The possibility to use historical performance data increases the reliability of the IoT-based monitoring system proposed.

4.6. Firebase Database Implementation

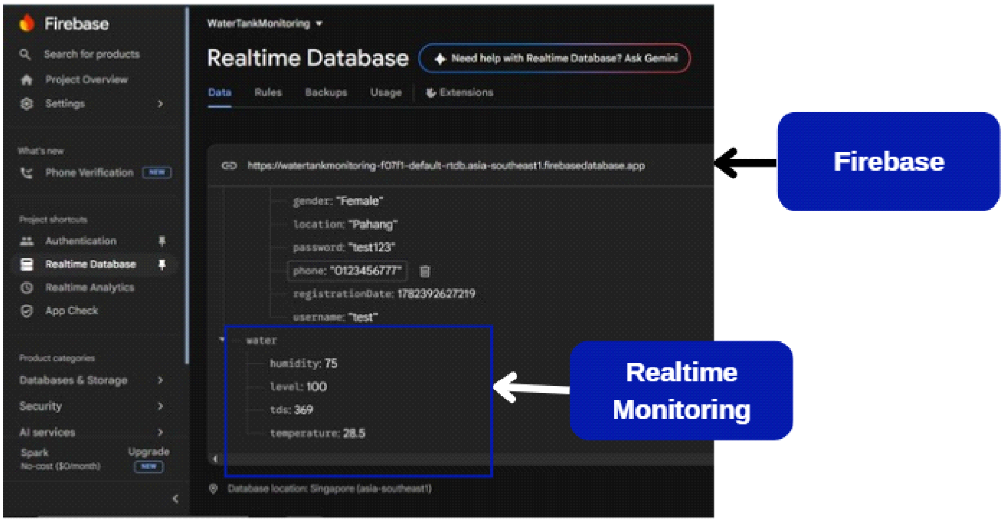

Figure 10: Firebase Realtime Database Structure

Figure 10 illustrates the Firebase Real-Time Database that would be used in the suggested solution. This database contains information like water level, TDS reading, temperature, humidity, and notification information. Firebase will aid in ensuring the synchronicity of ESP32 microcontroller with the mobile application in real-time.

5. TESTING AND RESULTS

5.1 Functional Testing

Table 4 highlights the functional tests carried out on the proposed Smart IoT-Based Water Tank Monitoring System. All modules of the system worked well during the testing process. The ESP32 effectively collected data from the sensors and forwarded them to the Firebase Realtime Database, and the mobile application showed real-time updates of the monitoring system data. In addition, the notifications functionality worked well when the level of water hit the set threshold.

Table 4: Functional Test Results

Test Item	Expected Result	Status
Water Level Detection	Sensor measures water level accurately	√
TDS Parameter Detection	TDS value displayed correctly	√
Temperature Monitoring	Temperature displayed correctly	√
Humidity Monitoring	Humidity displayed correctly	√
Firebase Synchronization	Real-time data synchronization	√
Mobile Application	Sensor readings displayed correctly	√
Notification Alert	Alert triggered below threshold	√

5.2 Real-World Testing Scenarios

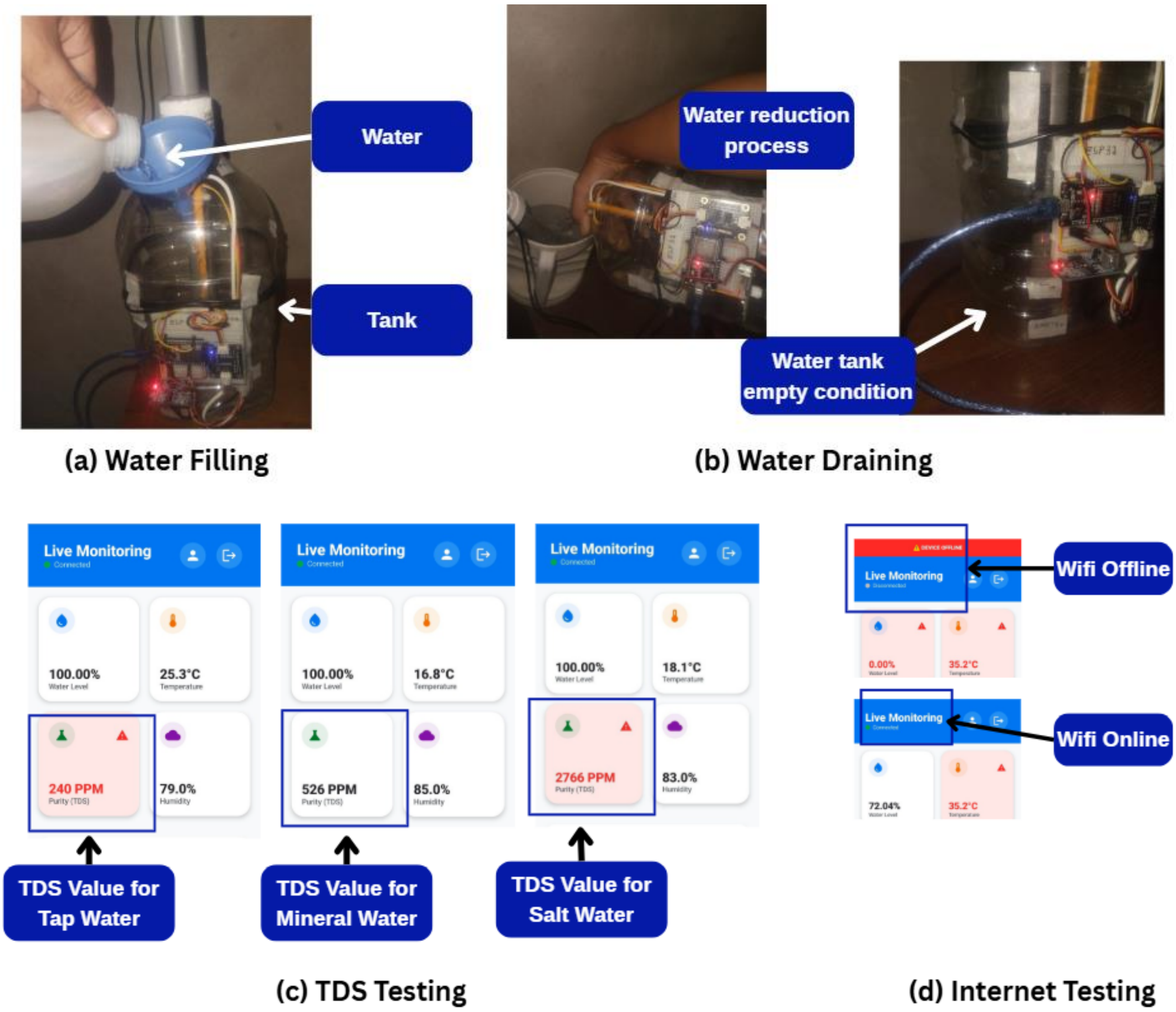

Figure 11: Real-World Testing Scenarios

Several testing cases shown in Figure 11 have been implemented to examine the efficiency of the proposed system. Water filling and water draining tests have been executed to test the reliability of the system in determining the correct water level at varying tank conditions. The TDS test was carried out to test the ability of the system in detecting varying levels of water quality by using different water samples. Internet connectivity test was done to test data synchronization in case of network interruption and restoration. The experiment results indicated the reliability of the system performance throughout all test cases.

5.3 Performance Evaluation

The testing of the performance of the proposed system involved functional tests and practical testing scenarios. The areas of concern for the testing were water level measurement, water quality monitoring, data synchronization, internet connection, app updates, and continuous monitoring performance. During the whole testing period, the system generated stable data from the sensors as well as successfully communicated between ESP32 microcontroller, Firebase Realtime Database, and the Android application.

The experimental data proved the accuracy of the ultrasonic sensor in measuring changes in the water level, while TDS and DHT11 sensors were measuring the TDS purity indexes and environmental parameters. Moreover, the notifications were activated in case the measured water level was lower than the required one. The system performed its automatic recovery upon internet connection restoration.

5.4 Comparison with Project Objectives

Table 5: Comparison between project objectives and achievements

Project Objective	Achievement
Develop an IoT-based water tank monitoring system	Achieved
Monitor water level in real time	Achieved
Monitor TDS purity using TDS sensor	Achieved
Display monitoring data through mobile application	Achieved
Synchronize data using Firebase	Achieved
Generate notification alerts	Achieved

A comparison between the set project goals and the actual achievements is given in Table 5. It can be seen from the table that all the goals that had been predetermined for the development had been successfully met.

6. DISCUSSIONS

6.1 Discussions

From the test results, it is clear that the implementation of the Smart IoT-Based Water Tank Monitoring System met the expected goals and objectives. The use of ESP32 microcontroller, JSN-SR04T ultrasonic sensor, TDS sensor, DHT11 sensor, Firebase Realtime database, and the Android mobile app gave a good IoT platform to monitor the water tanks in real time.

The tests conducted showed that the ultrasonic sensor effectively sensed the change in water level, and that the TDS and DHT11 sensors effectively monitored the TDS purity parameters and environmental conditions respectively. The sensor readings were effectively synced to Firebase and displayed by the mobile app with low latency over local network tests.

The automatic notifications worked efficiently to inform the users about the situation when the water level was lower than the set threshold. In addition, the internet connection test showed that the ESP32 can restore its connection with the cloud services immediately after any network failures.

When compared with conventional techniques such as manual monitoring and traditional float switches, the described IoT system presents many advantages due to the availability of features such as real-time monitoring, remote access, cloud synchronization, and multiple sensor usage. This increases the convenience of users by minimizing the need to visit water tanks physically.

In summary, the described system proves that low-cost IoT technology can be an efficient solution for water tank monitoring applications.

6.2 Limitations and Future Improvements

However, despite the positive functioning of the system, several limitations have been found during the process of its development and implementation. The main limitation of the proposed system is its dependency on the stable connection of the Wi-Fi network, since any disconnection of the internet will lead to delays in data exchange between the ESP32, the Firebase, and the mobile application. Also, the work of the ultrasonic sensor can be influenced by the water flow in the tank, incorrect positioning of the sensor, and the properties of the tank.

The current solution can only be applied to one tank only. Therefore, its extension to several tanks and integration into larger systems for monitoring the whole smart buildings will need additional improvements of the hardware architecture and the software.

For instance, industrial water level sensors, like radar sensors and pressure sensors, can be implemented in order to improve the measurements of the large tanks. The additional sensors that can be used include pH sensors, turbidity sensors, and dissolved oxygen sensors. Moreover, the use of machine learning can help in predicting the water usage pattern and detecting abnormal cases.

Table 6: Comparison between the proposed system and existing water tank monitoring systems

Feature	Traditional System	Basic IoT System	Proposed System
Real-time Monitoring	✗	✓	✓
Remote Mobile Access	✗	Limited	✓
Cloud Database Integration	✗	Limited	✓
Water Level Monitoring	✓	✓	✓
Water Quality Monitoring	✗	✗	✓
Temperature & Humidity Monitoring	✗	Limited	✓
Notification Alert	✗	Limited	✓
Cost	Low	Medium	Low

In this comparative operation, "Limited" refers to systems which have a web portal update manually done without the use of an in-built mobile push notification system. Cost categories have been classified using the local retail prices for each component; "Low" implies an integration single microcontroller costing below RM90 while "Medium" involves multi-board developments and shields.

7. CONCLUSION AND FUTURE WORK

7.1 Conclusion

A Smart IoT-Based Water Tank Monitoring System has been successfully implemented to automate existing water level monitoring techniques. The implementation involves IoT technologies, water level sensors, microcontrollers, and the application program for the smartphone in order to implement remote real-time monitoring of water tanks.

As it follows from the experiments conducted, the developed system succeeded in achieving its major goals, which were to implement the process of continuous collecting of data, visualization of it, and making it accessible via a smartphone application. As can be seen, the cooperation between hardware and software components proved to be effective in using IoT technology in monitoring processes and saving time on manual inspections.

In other words, the system allows users to have a reliable control over the amount of water available. In other words, the proposed system represents a good, cost-effective, and scalable solution to the problem of managing water supply in an effective way. Overall, this work shows how IoT-based solutions could be applied in order to transform the existing household systems into intelligent applications.

7.2 Future Work

Although the proposed system design meets the requirements set by the project, there is still a range of improvements which can be used during the further development process of the system. For example, it is possible to install more advanced water level sensors such as radar level sensors, pressure sensors, or ultrasonic industrial water sensors. Secondly, the water level sensors can be implemented in order to automate the process of water filling when the required water level is reached.

Additionally, there is an option to implement new communication protocols such as MQTT protocol or LPWAN technology in order to increase the effectiveness and reliability of the data transfer even in case of poor Internet access. Finally, it is necessary to add some additional functions to the mobile application such as visualizing historical data, analyzing water consumption, and using Cloud and AI technologies for predicting water consumption.

Further research could involve conducting comprehensive user acceptance tests, which would include different users and test the usability, functionality, and general satisfaction from the system. Such data would be very useful for making necessary modifications to the system's design and user interface. The future of the system would also be about scaling the system into an IoT-based water monitoring system that could handle more tanks than just one.

ACKNOWLEDGEMENT

The authors would like to thank the Faculty of Computing and Informatics and Universiti Malaysia Sabah for providing the resources to complete this research.

REFERENCES

[1] L. Da Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.

[3] World Health Organization, Guidelines for Drinking-water Quality, 4th ed., Geneva, Switzerland: WHO, 2022.

[4] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A Literature Review," Journal of Computer and Communications, vol. 3, no. 5, pp. 164–173, 2015.

[5] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A Survey," IEEE Internet of Things Journal, vol. 3, no. 1, pp. 70–95, Feb. 2016.

[6] Firebase, "Firebase Realtime Database Documentation," Google, 2025.

[7] S. Khandelwal and S. Dubey, "Smart Water Monitoring System Using IoT," International Journal of Innovative Research in Technology, vol. 7, no. 4, pp. 254–259, 2020.

[8] M. Rahman, M. R. Islam, S. Hasan, and M. A. Hossain, "IoT-Based Smart Water Monitoring System for Sustainable Water Management," International Journal of Advanced Computer Science and Applications, vol. 14, no. 3, pp. 623–631, 2023.

[9] P. Sharma, R. Verma, and A. Singh, "Arduino-Based Smart Water Management System Using IoT," International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 8, pp. 1452–1458, 2020.

[10] C. Livinti, A. Cojocaru, and D. Ursache, "Water Level Control and Monitoring System in a Tank Made with Arduino Uno and NodeMCU ESP8266 Development Boards," Engineering Proceedings, vol. 58, no. 1, 2023.

[11] R. Mahendiran, P. Kumar, and S. Rajesh, "IoT-Based Water Quality Monitoring System Using ESP32," International Journal of Scientific Research in Engineering and Management, vol. 9, no. 2, 2025.

[12] N. A. Jafri, M. R. Abdullah, and N. A. Rahman, "IoT-Based Real-Time Monitoring of River Water Quality: A Case Study of the Selangor River," Malaysian Journal of Computing, vol. 10, no. 1, pp. 45–58, 2025.

[13] N. H. Adan and M. Alias, "Design and Development of an Early Warning System for Monitoring Rising Water Levels," Journal of Advanced Research in Applied Sciences and Engineering Technology, vol. 38, no. 1, pp. 180–193, 2025.

[14] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson, 2016.

[15] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York, NY, USA: McGraw-Hill, 2019.

[16] M. Alshamrani, A. Myneni, A. Chowdhury, and D. Huang, "A survey on IoT architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 120–145, 2021.

[17] A. Whitmore, A. Agarwal, and L. Da Xu, "The Internet of Things—A survey of topics and trends," *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, 2015.

[18] A. Bahga and V. Madisetti, *Internet of Things: A Hands-on Approach*. Hyderabad, India: VPT, 2014.

[19] M. A. Hannan, A. Hussain, P. J. Ker, et al., "IoT applications and sensors integration in smart systems: A review," *Sensors*, vol. 21, no. 3, 2021.

[20] Rahman, G. and A.R.B.M, Zaidi., 2026. "Network Traffic Classification using Machine Learning: A Review of Decision Tree, Random Forest, and SVM Approaches on CIC-IDS2017". *International Journal of Machine Intelligence and Computing*, 3(1), pp.1-27.