IoT-Enabled Edge–Cloud Battery Management System for Real-Time Monitoring

Siva Sankar R.[1], Gokila Nivetha E.[2]

Assistant Professor, Department of Computer Science and Engineering, Kangeyam Institute of Technology, Kangayam, India.

Email: [1]sivasankar.cse@kitech.edu.in, [2]gokilanivetha.cse@kitech.edu.in

Abstract

The growing usage of lithium-ion batteries in electric vehicles, renewable energy generation and consumption systems, and portable electronic devices increases the need for a smart battery monitoring and management solution. In this paper, an IoT-enabled Edge–Cloud Battery Management System (BMS) is proposed to provide real-time monitoring and predictive analysis by means of lightweight Decision Tree classifier. The suggested approach is based on a unification of battery sensing, edge computing, wireless communication, and cloud analytics processes. Voltage, current, and temperature of a battery are monitored continuously using a voltage divider circuit, ACS712 current sensor, and DS18B20 temperature sensor. Data processing, battery state classification, and activation of battery protection mechanisms under abnormal conditions are done by Raspberry Pi Pico (RP2040) device, and Wi-Fi connection via ESP8266 module is used to communicate with a cloud platform. Battery state classification is performed into such three states as Normal, Warning, and Critical depending on voltage, current, and temperature values and protection is provided when abnormal state of the battery is detected. The results of experimental evaluation of the system demonstrate average Decision Tree execution time of 12 ms, end-to-end monitoring latency of 163 ms, packet delivery ratio of 98.2%, and cloud upload success rate of 98.7%.

Keywords: Battery Management System (BMS), Internet of Things (IoT), Edge-Cloud, Raspberry Pi Pico (RP2040), ESP8266 Wi-Fi module, lightweight Decision Tree.

DOI: https://doi.org/10.36548/jeea.2026.3.001
Received: 25.04.2026, received in revised form: 30.05.2026, accepted: 18.06.2026, published: 30.06.2026

1. Introduction

With the expansion of applications for electric cars, energy storage systems based on renewables, portable devices, and industrial automation the use of lithium-ion batteries as an energy storage becomes more frequent. The advantages of lithium-ion batteries include high energy density, good longevity and fast charge and discharge. Nevertheless, the performance and safe functioning of the batteries is heavily dependent on the environmental conditions, such as overheating, over-charging and deep discharging. Hence, constant monitoring of the parameters of the lithium-ion battery and intelligent management become necessary.

Most of the conventional Battery Management Systems (BMS) concentrate on such parameters as temperature, voltage, current, state of charge of the battery and provide certain protections. Nevertheless, most of the currently implemented BMS are based on the centralized architecture that could introduce communication delays, and hence impede rapid decision making. Besides, with the growing popularity of the distributed battery-powered systems there is a necessity in more scalable monitoring solutions.

In this context, the evolution of edge computing and cloud technologies presents potential for building intelligent battery management systems. In addition to performing local processing of information from sensors and carrying out protection procedures, edge computing technology provides the possibility of storing and analyzing data using cloud platforms. This combination allows increasing the efficiency of monitoring systems as well as enhancing their reliability. Nevertheless, building effective decision-making algorithms for resource-limited edge devices is a challenging task.

To solve these problems, this work offers to design an Edge-Cloud Battery Management System which involves real-time battery monitoring, lightweight Decision Tree-based edge intelligence, wireless IoT communication, and cloud analytics. The presented system allows constantly tracking the values of the voltage, current, and temperature of batteries, fast battery state classification using edge intelligence, and battery data visualization on the cloud.

2. Related Works

The increasing number of electric cars, renewable energy storage system applications, and portable devices has resulted in the emergence of a need for intelligent Battery

Management Systems (BMS), which can ensure safety and effective use of the battery. The recent research has been aimed at introducing machine learning algorithms, IoT, and embedded computing systems into battery monitoring and predictive analytics. The application of lightweight machine learning algorithms for battery state-of-charge (SOC) estimation on limited hardware showed promising results for battery monitoring in terms of prediction accuracy without high computational requirements [1].

Machine learning algorithms have widely been used for battery life prediction and health assessment. Predictive analytics using IoT-enabled BMS framework has yielded promising results in RUL estimation of lithium-ion batteries based on continuous collection of sensor data [2]. Also, real-time monitoring systems using machine learning algorithms allowed improving battery state-of-charge prediction and battery performance evaluation [3].

Cloud computing has proven to be an efficient platform for large scale battery monitoring owing to its capabilities of centralized data storage, remote access, and sophisticated analytics [4]. Nonetheless, cloud computing architectures can suffer from latency issues, high communication costs, and dependence on network availability. End-edge-cloud architecture was proposed to mitigate the above problems, allowing distributed processing of data on sensors, edge nodes, and cloud infrastructure [5]. The use of such architecture decreases response time, makes better bandwidth usage possible, and increases scalability of the system [6].

Precise estimation of state of health of the battery and its remaining useful life is another important challenge that requires further investigation. Deep learning and graph-based models have shown good results in capturing degradation patterns and predicting performance of the battery [7]. Battery health evaluation methods which include voltage-based analysis and machine learning-based methods using data obtained during experiments and simulations have made further progress in increasing the reliability of battery predictions in different environments [8], [9]. Moreover, IoT-based low-cost platforms for battery monitoring with cloud services allow gathering battery data remotely [10].

Even with the advancement of such technologies, very little research has been carried out on the application of decision trees in the management of battery management systems using edge-cloud computing. Given the simplicity in computation of the decision tree approach and its fast implementation, it is highly recommended for use in edge environment. It is thus important to develop an edge cloud battery management system that makes use of decision trees as its analytics algorithm.

3. Proposed Methodology

The design of the BMS includes four functional layers including Battery and Sensing Layer, Edge Processing Layer, Communication Layer, and Cloud Analytics Layer as shown in Figure 1.

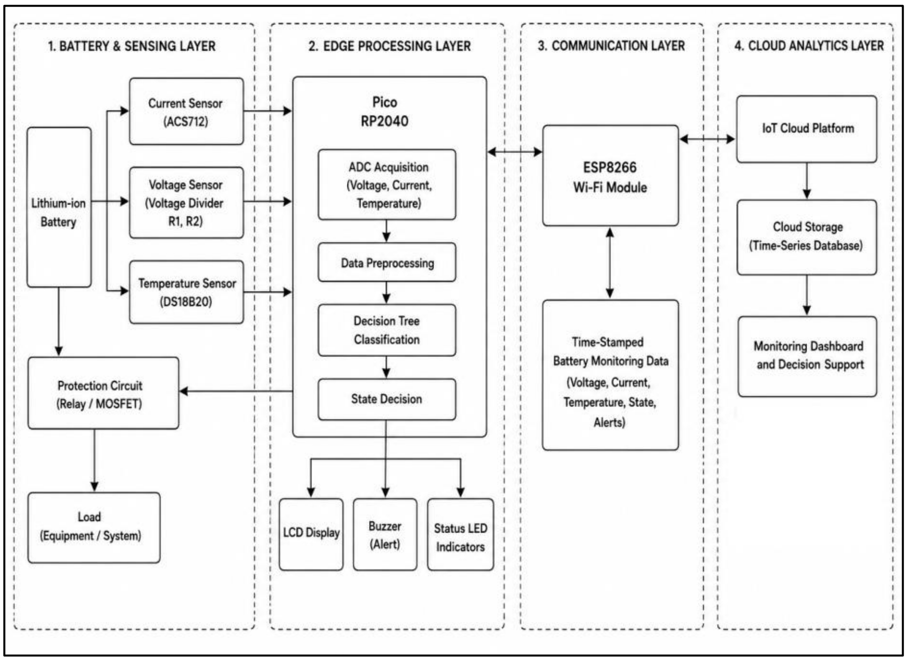


Figure 1. Proposed IoT-Enabled Edge–Cloud BMS Architecture

In this architecture, the combination of battery sensing, edge computing, wireless communication, and cloud monitoring is used to achieve low-latency battery monitoring and intelligent decision making. Battery and Sensing Layer is responsible for acquiring battery operating parameters by using ACS712 current sensor, voltage divider circuit, and DS18B20 temperature sensor. The current sensor, voltage divider circuit, and temperature sensor measures battery current, battery voltage, and battery temperature, respectively. The parameters are passed to the edge controller for further processing. The relay/MOSFET-based battery protection circuit is connected between battery and load.

The Edge Processing layer is constructed from the Raspberry Pi Pico (RP2040) microcontroller. The RP2040 will undertake analog to digital converter (ADC) sampling and pre-processing of the acquired signal by implementing a light-weight decision tree classifier for estimating battery state. In particular, the decision tree classifier receives battery voltage (V), current (I) and temperature (T) as the input feature set and classifies battery behavior into either Normal, Warning or Critical levels according to Table 1 below.

Table 1. Battery State Classification Thresholds

Parameter	Normal State	Warning State	Critical State
Battery Voltage	3.2–4.2 V	3.0–3.2 V or 4.2–4.3 V	<3.0 V or >4.3 V
Battery Current	≤2 A	2–3 A	>3 A
Battery Temperature	20–45°C	45–55°C	>55°C
Relay Status	ON	ON	OFF
Buzzer Status	OFF	OFF	ON

The classification procedure employed by the Decision Tree is illustrated in Figure 2. The classification process starts with the voltage determination since an under voltage state or an over voltage state directly influences the battery safety. The battery will be classified as Critical when the value of the voltage is less than 3.0V or greater than 4.3V. The battery will be classified as Warning when the voltage value is within the warning range (3.0-3.2V or 4.2-4.3V). If the voltage value is within the safe range, the classifier moves on to current and temperature analysis.

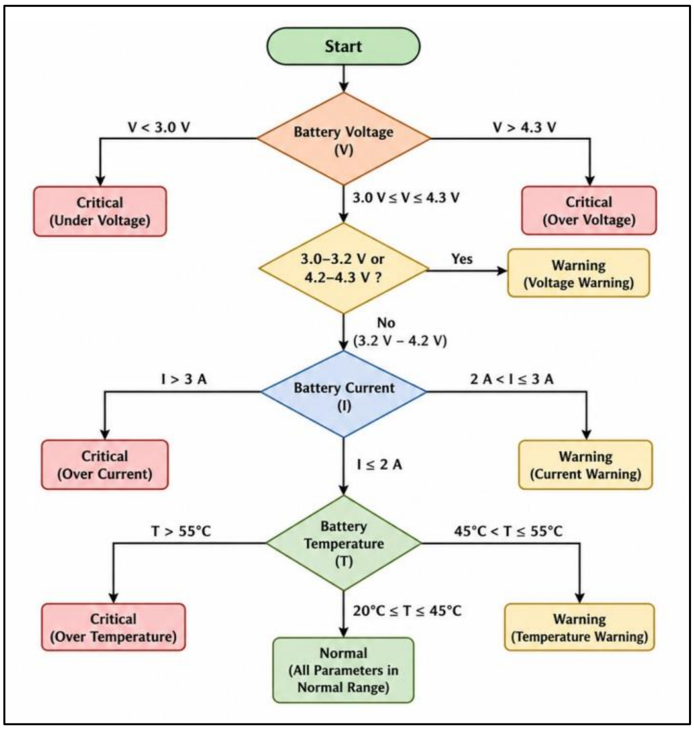


Figure 2. Battery State Classification Using Decision Tree

All current levels above 3 A are categorized as Critical, while all current levels ranging between 2 A and 3 A are designated as Warning. Likewise, all temperature levels that are above 55°C are defined as Critical, and all temperature ranges between 45°C and 55°C are considered Warning. Provided that all measured levels stay within normal operating ranges presented in Table 1, the battery is designated as Normal. As per the Decision Tree, when normal operating conditions prevail, the monitoring process goes on unimpeded. Whenever the parameters exceed the warning thresholds, visual alarms are set off. Under the conditions of critical operation, the protection circuit disconnects the battery from the load, and an audible alarm sounds to protect against any damage.

The Communication Layer employs the ESP8266 Wi-Fi module to create a connection between the edge device and the cloud platform. The module sends the battery data, including voltage, current, temperature, battery state, and alerts at regular intervals with a timestamp. The Cloud Analytics Layer allows centralized storage and visualization of the battery information. Incoming records are stored in a time-series database.

In order to measure the battery operating conditions, the below mentioned equations have been used.

The battery voltage is determined using the voltage divider relationship:

$$V_{bat} = V_{adc}\left(\frac{R_1+R_2}{R_2}\right) \tag{1}$$

where V_{bat} is the actual battery voltage, V_{adc} is the measured ADC voltage, and R_1 and R_2 are the divider resistances.

The battery current is calculated from the ACS712 sensor output as

$$I = \frac{V_{\text{out}} - V_{\text{offset}}}{S} \tag{2}$$

where V_{out} is the sensor output voltage, V_{offset} is the zero-current reference voltage, and S is the sensor sensitivity.

The RP2040 pre-processes the acquired data and uses a light-weight Decision Tree algorithm to classify the battery operational state.

The classification process is represented as

$$\text{State} = f(V_{\text{bat}}, I, T) \tag{3}$$

where V_{bat}, I, and T denote battery voltage, current, and temperature, respectively.

The instantaneous battery power is computed as

$$P = V_{bat} \times I \tag{4}$$

while the cumulative energy consumption is estimated using

$$E = \sum_{k=1}^{N} V_k I_k \Delta t \tag{5}$$

where E denotes the total energy consumption, V_k and I_k represent the voltage and current measurements at the kth sample, N is the number of measurements and Δt is the sampling interval.

Algorithm 1: Edge–Cloud Battery Monitoring Using Decision Tree

Input: Battery Voltage (V), Current (I), Temperature (T)
Output: Battery State (Normal, Warning, Critical), Alert Status, Protection Action

1. *Start*
2. *Initialize RP2040, Sensors, ESP8266, LCD, Buzzer, and Protection Circuit*
3. *Acquire battery parameters V, I, and T*
4. *Preprocess sensor readings*
5. *Apply Decision Tree classification*
6. *If (V, I, T within safe limits)*
   State ← Normal
   Display Normal Status
7. *Else if (V, I, T exceed warning thresholds)*
   State ← Warning
   Activate LED Indicator
   Display Warning Message
8. *Else*
   State ← Critical
   Activate Buzzer
   Disconnect Battery using Relay/MOSFET
   Display Critical Alert
9. *Generate timestamped monitoring record*
10. *Transmit V, I, T, State, and Alert Status to Cloud Server*
11. *Store data in cloud database*
12. *Update cloud dashboard and notifications*
13. *Repeat Steps 3–12 continuously*
14. *End*

4. Implementation

The proposed system was developed using the components specified in Table 2. The experimental setup is based on the following: Raspberry Pi Pico (RP2040), current sensor ACS712, voltage divider, temperature sensor DS18B20, ESP8266 Wi-Fi chip, LCD screen, relay protection scheme, DC cooling fan and lithium-ion battery. The RP2040 works as an edge controller performing sensors' data collection, processing, and decision tree calculation.

Table 2. Hardware Components Used in the Proposed System

Component	Function
Raspberry Pi Pico (RP2040)	Edge controller for data processing and decision tree execution
ACS712 Sensor	Measures battery charging/discharging current
Voltage Divider (R1, R2)	Measures battery voltage
DS18B20 Sensor	Monitors battery temperature
ESP8266 Wi-Fi Module	Enables cloud communication
16×2 LCD Display	Displays real-time battery parameters
Buzzer	Generates warning alerts
Relay/MOSFET Circuit	Disconnects battery during critical conditions
DC Cooling Fan	Dissipates heat to maintain safe battery operating temperature
Lithium-Ion Battery	Energy storage unit under monitoring
Power Supply Unit	Provides regulated power to all system components

The implemented prototype structure is presented in Figure 3 below. The edge device performs the real-time monitoring and battery safety testing through classification of the battery state to one of three possible states: Normal, Warning or Critical. According to the classification results, the proper actions, such as warning indicators activation, audio alarms triggering or battery disconnection through the protection circuit, are taken.

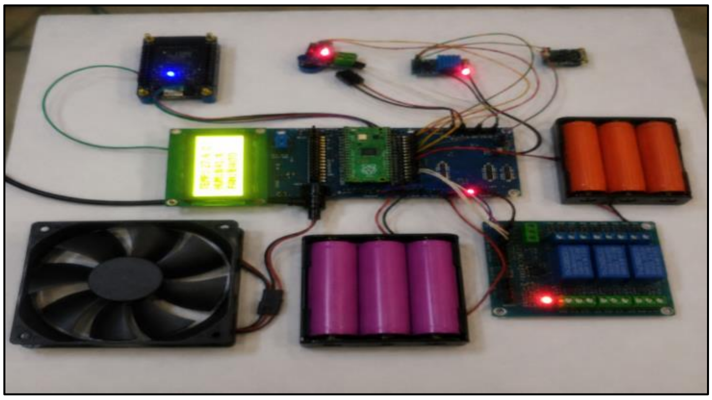


Figure 3. Experimental Prototype of the Proposed Battery Management System

In the remote monitoring process, the RP2040 microcontroller interacts with the ESP8266 Wi-Fi module, which sends the battery data along with timestamps to the cloud platform. Figure 4 shows the IoT cloud server dashboard that is used to store and visualize the parameters of the battery.

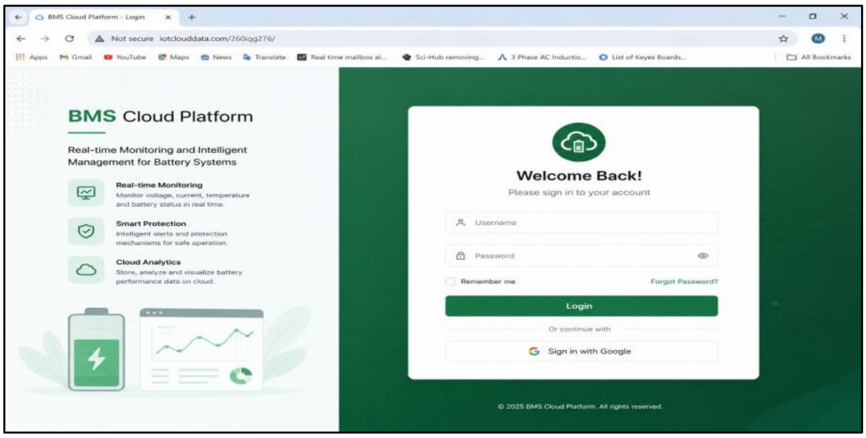
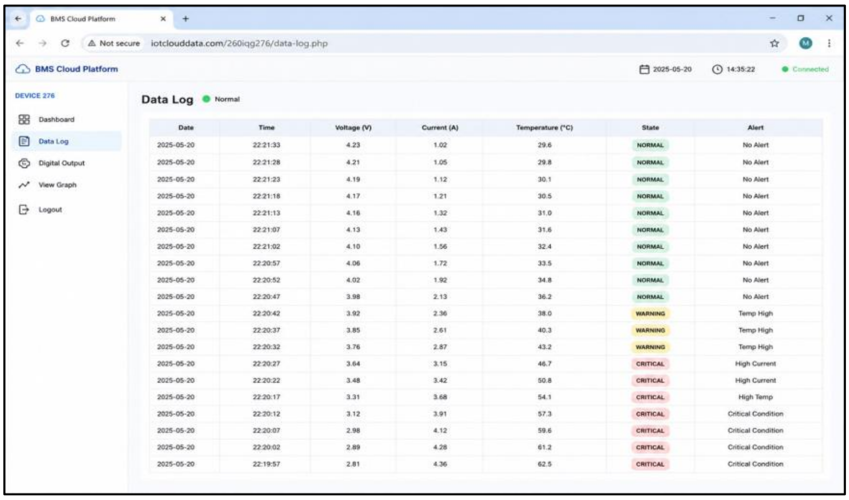


Figure 4. User Interface for Real-Time Battery Monitoring



Date	Time	Voltage (V)	Current (A)	Temperature (°C)	State	Alert
2025-05-20	22:21:33	4.23	1.02	29.6	NORMAL	No Alert
2025-05-20	22:21:28	4.21	1.05	29.8	NORMAL	No Alert
2025-05-20	22:21:23	4.19	1.12	30.1	NORMAL	No Alert
2025-05-20	22:21:18	4.17	1.21	30.5	NORMAL	No Alert
2025-05-20	22:21:13	4.16	1.32	31.0	NORMAL	No Alert
2025-05-20	22:21:07	4.13	1.43	31.6	NORMAL	No Alert
2025-05-20	22:21:02	4.10	1.56	32.4	NORMAL	No Alert
2025-05-20	22:20:57	4.06	1.72	33.5	NORMAL	No Alert
2025-05-20	22:20:52	4.02	1.92	34.8	NORMAL	No Alert
2025-05-20	22:20:47	3.98	2.13	36.2	NORMAL	No Alert
2025-05-20	22:20:42	3.92	2.36	38.0	WARNING	Temp High
2025-05-20	22:20:37	3.85	2.61	40.3	WARNING	Temp High
2025-05-20	22:20:32	3.76	2.87	43.2	WARNING	Temp High
2025-05-20	22:20:27	3.64	3.15	46.7	CRITICAL	High Current
2025-05-20	22:20:22	3.48	3.42	50.8	CRITICAL	High Current
2025-05-20	22:20:17	3.31	3.68	54.1	CRITICAL	High Temp
2025-05-20	22:20:12	3.12	3.91	57.3	CRITICAL	Critical Condition
2025-05-20	22:20:07	2.98	4.12	59.6	CRITICAL	Critical Condition
2025-05-20	22:20:02	2.89	4.28	61.2	CRITICAL	Critical Condition
2025-05-20	22:19:57	2.81	4.36	62.5	CRITICAL	Critical Condition

Figure 5. Historical Battery Data Logging on Cloud Platform

The timestamp-based battery data logging and cloud storage technique used in the suggested Edge-Cloud Battery Management System is shown in Figure 5. Battery voltage, battery current, battery temperature, battery status, and alert data are periodically sent from the edge system to the cloud server for storage as time-stamped data. The historical database will be useful in monitoring battery operation.

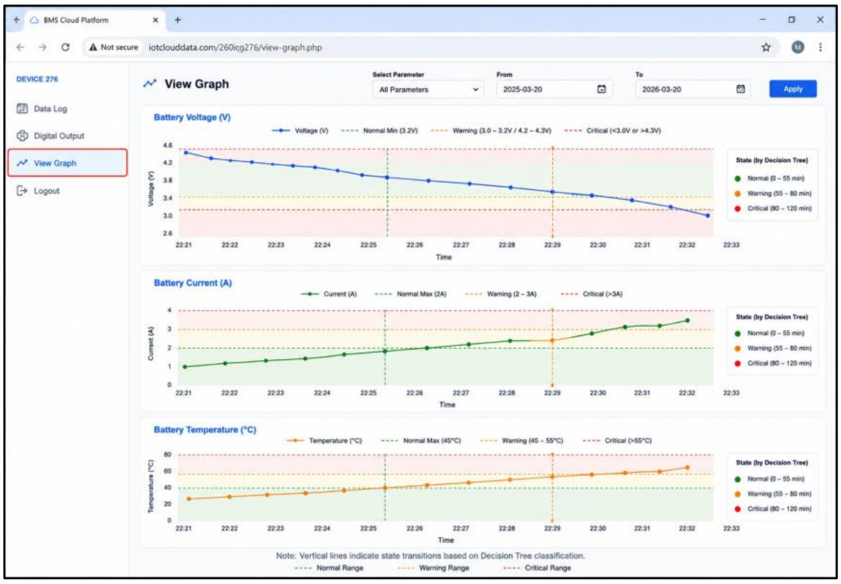


Figure 6. Battery Performance Trends and State Classification Dashboard

Figure 6 presents the Cloud based battery monitoring dashboard, which consists of voltage, current and temperature trends in addition to decision tree based battery state classification. The graphic interface provides the ability to visualize the Normal, Warning and Critical operational zones, providing an opportunity for monitoring the performance of the batteries.

5. Results and Discussion

The performance of the suggested Edge-Cloud Battery Management System was analyzed based on monitoring responsiveness, communication reliability, and classification efficiency.

The total latency time of monitoring is defined by the following expression:

$$L_{\text{total}} = L_{\text{sense}} + L_{\text{edge}} + L_{\text{trans}} + L_{\text{cloud}} \tag{6}$$

where L_{sense}, L_{edge}, L_{trans}, and L_{cloud} are sensing delay, edge computing delay, communication delay, and cloud computing delay, respectively.

Communication reliability was evaluated using the Packet Delivery Ratio (PDR):

$$\mathrm{PDR}(\%) = \frac{N_{\text{received}}}{N_{\text{sent}}} \times 100 \quad (7)$$

where N_{received} is the number of successfully received packets and N_{sent} is the total number of transmitted packets. The obtained PDR of 98.2% confirms reliable wireless communication between the edge node and cloud server.

Similarly, the cloud upload success rate was determined using

$$USR(\%) = \frac{D_{\text{success}}}{D_{\text{total}}} \times 100 \quad (8)$$

where D_{success} and D_{total} represent successful and total uploaded records, respectively. The achieved upload success rate of 98.7% indicates stable cloud synchronization.

The experimental tests proved stability of the process under continuous monitoring, along with the proper acquisition and delivery of the battery parameters to the cloud computing system.

Table 3. Performance Evaluation of the Proposed System

Metric	Value
Voltage Measurement Range	0–16 V
Current Measurement Range	±30 A
Temperature Measurement Range	-10°C to 85°C
Sensor Sampling Rate	1 sample/s
Decision Tree Execution Time	12 ms
Edge Processing Delay	18 ms
Data Transmission Delay	145 ms
End-to-End Monitoring Latency	163 ms
Cloud Data Upload Success Rate	98.70%
Packet Delivery Ratio	98.20%
Alert Detection Time	< 200 ms

Dashboard Refresh Interval	1 s
Continuous Monitoring Duration	24 h
System Availability	99.10%

From the data presented in Table 3, it is possible to conclude that the experiment has proven the effectiveness of using the Decision Tree classifier for the recognition of normal and abnormal working states of the battery along with the low execution time (12 ms). Together with the measurement of the time necessary to give an alert (less than 200 ms), it can be stated that the developed edge-based system provides fast detection of faults and battery protection.

6. Conclusion

This research has developed the Edge-Cloud Battery Management System, which unites real-time battery measurements, edge computing, communication technologies, and cloud analytics in one comprehensive system. The proposed system constantly collects information on battery voltage, current, and temperature and uses the lightweight Decision Tree classifier running on the Raspberry Pi Pico (RP2040) for performing battery condition classification. Classification at the edge level allows detecting faults and taking corresponding measures instantly while minimizing the use of cloud computing resources. The implementation of the ESP8266 communication module and cloud allows remotely monitoring batteries, storing collected data in the cloud, and performing battery analytics in the cloud. As a result of experimental evaluations, we have achieved high performance of the system with the Decision Tree execution time of 12 ms, end-to-end delay of 163 ms, packet delivery ratio of 98.2%, and cloud uploading efficiency of 98.7%. The proposed architecture can be considered effective in terms of low delay and reliable communications.

References

[1] Pau, Danilo Pietro, and Alberto Aniballi. "Tiny Machine Learning Battery State-of-Charge Estimation Hardware Accelerated." Applied Sciences 2024, vol. 14, no. 14: 6240.

[2] Krishna, Gopal, Rajesh Singh, Anita Gehlot, Ahmad Almogren, Ayman Altameem, Ateeq Ur Rehman, and Seada Hussen. "Advanced Battery Management System

Enhancement Using IoT and ML for Predicting Remaining Useful Life in Li-Ion Batteries." Scientific Reports 2024, vol. 14, no. 1: 30394.

[3] Gururagavendran, G., M. Deepak Kumar, and B. Blessed Sam. "Real-Time Monitoring System for Battery SOC Prediction and Metrics Analysis Using ML Techniques." 5th International Conference on Communication, Computing and Electronics Systems (ICCCES), IEEE, 2026, 541-546.

[4] Ismail, Mohanad, and Ryan Ahmed. "A Comprehensive Review of Cloud-Based Lithium-Ion Battery Management Systems for Electric Vehicle Applications." IEEE Access 2024, vol. 12: 116259-116273.

[5] Mulpuri, Sai Krishna, Bikash Sah, and Praveen Kumar. "An Intelligent Battery Management System (BMS) with End-Edge-Cloud Connectivity–A Perspective." Sustainable Energy & Fuels 2025, vol. 9, no. 5: 1142-1159.

[6] Tran, Manh-Kien, Satyam Panchal, Tran Dinh Khang, Kirti Panchal, Roydon Fraser, and Michael Fowler. "Concept Review of a Cloud-Based Smart Battery Management System for Lithium-Ion Batteries: Feasibility, Logistics, And Functionality." Batteries 2022, vol. 8, no. 2: 19.

[7] Wei, Yupeng, and Dazhong Wu. "State of Health and Remaining Useful Life Prediction of Lithium-Ion Batteries with Conditional Graph Convolutional Network." Expert Systems with Applications 2024, vol. 238: 122041.

[8] Huang, Rui, Jackson Fogelquist, and Xinfan Lin. "Robust Estimation of Battery State of Health Using Reference Voltage Trajectory." American Control Conference (ACC), IEEE, 2025, 4065-4070.

[9] Al-Rahamneh, Anas, Irene Izco, Adrian Serrano-Hernandez, and Javier Faulin. "Machine Learning-Based State-of-Health Estimation of Battery Management Systems Using Experimental and Simulation Data." Mathematics 2025, vol. 13, no. 14: 2247.

[10] Aydın, Mustafa, and İsmail Gürbüz. "IoT Based a Low Cost Battery Monitoring System Using ESP8266 and Arduino IoT Cloud Platform." International Journal of Automotive Engineering and Technologies 2024, vol.13, no. 4: 170-179.