IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 5, October 2025, pp. 3858~3868
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i5.pp3858-3868

Optimizing battery life: a TinyML approach to lithium-ion battery health monitoring

Kamaraj Lalitha Nisha[1], Vasanth Pradeep[1], Padmanabhan Puthiyaveedu Krishnankutty Nair[1], Sreelakshmi Pillai[1], Manikandan Arunachalam[1], Rakesh Thoppaen Suresh Babu[2]
[1]Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Kollam, India
[2]Data Scientist, AI Advisory, Fiserv Inc, Alpharetta, United States

Article Info

Article history:

Received Jul 3, 2024
Revised Jun 30, 2025
Accepted Jul 13, 2025

Keywords:

Battery capacity prediction
Charge estimation
ESP32
Lithium-ion
TinyML

ABSTRACT

Electrical vehicles (EVs) are crucial nowadays due to their reduction in greenhouse gas emissions, decreasing dependence on remnant fuels, and improving air quality. For EVs, the battery is the heart that determines range, performance, and efficiency. Also, it directly impacts the cost and overall vehicle life span. Lithium-ion (Li-ion) batteries are pivotal in powering modern portable electronics and electric vehicles due to their high energy density and durability. Issues with current batteries include slow charging, short cycles, and low energy density. Most of the problems with current batteries are resolved by Li-ion batteries, which also helps explain why EV usage is increasing globally. However, to guarantee maximum performance and safety, estimating the remaining useful life and health state of these batteries remains a major difficulty. To improve battery lifetime of the battery and to overcome the problems of delayed charging, this study introduces a tiny machine learning (TinyML) method. An innovative machine learning approach is put forth that allows for effective learning on devices with limited resources, which enables real-time monitoring of the health status of the Li-ion batteries.

Corresponding Author:

Manikandan Arunachalam
Department of Electronics and Communication Engineering, Amrita School of Engineering
Amrita Vishwa Vidyapeetham
Amritapuri, Kollam, Kerala, India
Email: mani_ac18@yahoo.co.in

1. INTRODUCTION

Lithium-ion (Li-ion) batteries are essential to contemporary portable gadgets and electric vehicles, because of their extended lifespan and high energy density [1]. However, maximizing performance and guaranteeing safety depends on precisely estimating the remaining usable life (RUL) and health state of these batteries [2], [3]. For example, drone operators frequently replace batteries after a predetermined number of cycles to reduce dangers, even if the batteries might still be acceptable for a few shorter flights. Also, since the demand for portable and sustainable energy sources continues to grow, the importance of effective and reliable Li-ion batteries has never been more highlighted [4]. These batteries power enormous devices, from smartphones to laptops to electrical vehicles (EVs) and renewable energy storage systems. Despite their extensive use, Li-ion batteries face significant challenges related to their life cycle and health monitoring [5]. Traditional battery management technologies often depend on complex and resource demanding methods to monitor battery health which is costly and inefficient. Recent advancements in tiny machine learning (TinyML) exhibit a promising solution to these challenges. TinyML influences the capabilities of machine learning on resource constrained devices, enabling real-time and precise monitoring of battery health with reduced computational overhead [6]. By incorporating TinyML techniques in battery management systems, it is possible to optimize battery life, improve performance, and predict potential crashes before they happen, ensuring safer and more effective operation of batteries under various situations. The RUL of Li-ion batteries is strongly influenced by temperature and load (current or depth of discharge). These factors accelerate degradation through mechanisms like capacity fade and internal resistance growth. The following is a commonly used model to estimate RUL: a degradation-based RUL model for Li-ion batteries often combines the Arrhenius equation (temperature dependence) and load-dependent cycling damage as in (1).

$$RUL = \frac{Q_{max} - Q_{current}}{k_1 \cdot e^{\frac{-E_a}{k_B T}} + k_2 \cdot C_{rate}^{b}} \tag{1}$$

Where Q_{max} is initial battery capacity (Ah), $Q_{current}$ is current battery capacity (Ah), k_1,k_2 are empirical degradation rate constants for temperature and cycling, E_a is activation energy, k_B is Boltzmann constant (1.38×10^{-23} J/K), T is absolute temperature (Kelvin), C_{rate} is charge/discharge rate (relative to 1 C, e.g., 0.5 C or 2 C), and b is exponent reflecting load dependency. RUL in battery health monitoring predicts the time a battery can continue to perform effectively before needing replacement, enabling proactive maintenance and optimization.

The integration of TinyML in battery health monitoring has emerged as a promising area of research, incorporating advancements in machine learning with the practical needs of battery management technology. Early research in deploying machine learning to battery health monitoring set the base for data-driven methods. Severson *et. al.* [7] conducted an innovative study where they applied machine learning algorithms to predict the life cycles of Li-ion batteries using early charge-discharge information. This approach considerably enhanced the accuracy of predictions while comparing them to the traditional methods, highlighting the potential of machine learning in battery lifetime analysis. Further extending this work, various machine learning methods, including support vector machines and neural networks were explored to predict the battery lifetime and capacity discharging [8]. Their observations underestimated the importance of feature selection and the promise of ensemble learning methods in improving the accuracy of prediction. The paradigm shift towards TinyML signifies a step forward in battery health monitoring, focusing on deploying machine learning methods on resource-controlled devices. A lightweight neural network optimized for microcontrollers, capable of real-time health predictions for Li-ion batteries [9]. This method not only reduced the computational load but also maintained high accuracy, making it ideal for portable EVs. Building on this, TinyML with IoT devices for remote monitoring of battery health was incorporated to study the exhibited possibility of implementing TinyML models on edge devices, permitting continuous monitoring without relentless connectivity to cloud servers. This integration enabled robust data collection and real-time analysis, paving the way for predictive maintenance and fault diagnosis in battery monitoring [10]. Gruosso and Gajani [11] demonstrated the comparison of machine learning algorithms in TinyML for the estimation of battery state. ANN based prediction methods are deployed and compared in terms of cost, memory, and computational power. Furthermore, Lord and Kaplan [12] analyzed the application of two different neural networks in TinyML frameworks for battery anomaly detection. Finally, the optimized model was mounted in the microcontrollers, indicating significant improvements in the accuracy detection with low power consumption.

The integration of IoT and TinyML further improved the abilities of battery health monitoring system [13]. Studies in research suggested a hybrid model that combined a traditional battery monitoring system with TinyML enhancements, resulting in enhanced accuracy in estimating the remaining useful life of batteries [14]. Moreover, Wang *et. al.* [15] created an IoT based framework for devoted battery health monitoring. This study employed TinyML models deployed on edge devices, enabling real time analysis, and decision making. The architecture reduced the support on centralized cloud servers, improving the robustness and scalability of the monitoring system. While the existing research applications of TinyML battery health monitoring show promising results, there are challenges to address [16]. The key challenge is shortage of standardized data sets and benchmarking frameworks to evaluate the performance of TinyML models across different applications. Moreover, ensuring the robustness and security of TinyML models implemented on edge devices is critical, particularly in safety critical applications with EVs [17]. To overcome these limitations of the existing works, a real-time battery health monitoring system is proposed for Li-ion batteries which enables real-time analysis on edge devices. ESP32 microcontroller, a simple and low-cost chip that makes the solution accessible and affordable, is a crucial component of our implementation. This implementation shows how useful TinyML is for estimating battery health in practical applications [18]. To select the optimal battery charging procedures, the state of charge (SoC) estimation is essential. Particularly regarding rapid charging, which allows for a substantial reduction in charging time without compromising the battery's overall lifespan [19]. TinyML battery health assessment is being advanced to assist larger objectives of improving performance and safety because Li-ion batteries are extremely explosive [20], [21], and Li-ion battery longevity is the major requirement in a variety of applications [22], [23]. This helps to create more effective and sustainable energy solutions internationally, which is good for businesses and consumers alike. Our study extends battery lifespan and promotes sustainability by optimizing EV charging cycles, preventing unplanned shutdowns, and increasing user experience through improved battery life projections. To assure dependable predictions under TinyML limitations, we investigate a variety of machine learning techniques appropriate for resource-constrained contexts, weighing inference speed, model size, and accuracy. Our research uses TinyML, a cutting-edge technology that enables machine learning on devices with limited resources, to tackle this problem.

2. METHODOLOGY

Our objective is to use TinyML to create a small, precise prediction model for Li-ion battery capacity estimation. The various steps involved in this process are data preparation, model development, conversion, optimization, and deployment. The proposed model was trained on thorough and pertinent data by utilizing the NASA battery dataset, which enabled accurate and effective real-time predictions on a device with limited resources. This dataset is ideal for capacity prediction because the discharging and charging patterns are fixed. The SoC estimation is influenced by a battery's charging and discharging efficiencies.

2.1. Convolutional neural networks

The NASA battery dataset is preprocessed to prepare the dataset for efficient use in convolutional neural network (CNN) model training. Preparing the data for CNN-based analysis involves activities, including feature engineering, normalization, and data cleaning. The NASA battery dataset's raw data, which includes variables like voltage, current, and temperature, is transformed to ensure consistency and applicability during the training phase [24], [25]. Furthermore, the alignment of time series and the resolution of missing values are essential components in getting the data ready for effective CNN model training. By improving the dataset's quality and consistency, these preprocessing procedures should help the CNN model learn and predict more accurately during training [25].

CNN models are built using a set of architectural parameters, including the number of convolutional layers, the size of feature maps inside each layer, and the connectivity between them. The CNN architecture is ultimately customized according to the dataset's complexity and the model's intended ability to accurately capture spatial patterns and feature hierarchies pertinent to the task at hand. Determining the ideal number of layers, feature map size and organization, and layer connection are all included in this careful design. CNNs are capable of effectively extracting significant information from the input data and capturing the spatial relationships necessary for precise prediction by carefully planning their architecture [26]. Therefore, the foundation for attaining high performance and resilience in a variety of CNN-based applications is a well-designed CNN architecture. The purpose of this validation dataset is to evaluate the model's ability to generalize to previously undiscovered data independently. Different assessment metrics are frequently used to measure the difference between the predicted values and the actual ground truth labels in the validation dataset. Examples of these metrics are mean absolute error (MAE) and root mean squared error (RMSE). These measures shed light on how the predictions made by CNN are accurate and dependable [27]. Researchers and practitioners can learn a great deal about CNN's predictive accuracy by examining the evaluation findings. To further improve the model's performance, they can also spot possible areas for improvement, such as altering the model architecture or fine-tuning hyperparameters. To reduce the amount of memory and processing power needed, this entails model quantization or decreasing the precision of weights. After that, the models are transformed into a format that is compatible with TinyML, allowing devices with low processing power to use them. By optimizing the models for TinyML, the approach becomes relevant for edge computing and IoT devices, as it guarantees that the models can be implemented on microcontrollers.

3. HARDWARE IMPLEMENTATION

The development of a reliable Li-ion battery capacity prediction system is essential for the efficient management and longevity of batteries in various applications [28]. Our project utilizes a customized ESP32 microcontroller-based hardware setup, coupled with a CNN model, to predict the remaining capacity of Li-ion batteries. This system integrates advanced sensors and a custom printed circuit board (PCB) to ensure accurate data collection and robust performance, even in challenging conditions. The ESP32 is a versatile microcontroller featuring a dual-core Xtensa LX6 processor with a 32-bit architecture and clock speeds of up to 240 MHz [29]. It includes 520 KB of SRAM, 4 MB of flash memory (expandable to 16 MB), and built-in Wi-Fi (2.4 GHz, IEEE 802.11 b/g/n) and Bluetooth v4.2 (BR/EDR and BLE) for wireless connectivity. Figure 1 shows all the hardware components used in the project. Key to our design is the use of the INA219 voltage and current sensor, which provides precise digital measurements. The sensor operates by measuring the voltage drop across a shunt resistor, allowing it to calculate the current and voltage of the battery. The digital nature of the data ensures that the integrity of the transmitted values remains intact, even when the microcontroller is positioned at a distance from the sensor. This is crucial for maintaining accuracy in real-world applications where sensor placement flexibility is needed. For temperature monitoring, the DS18B20 probe-type sensor is employed. This sensor directly connects to the battery body, providing accurate temperature readings. The DS18B20 uses a 1-wire protocol, which is advantageous as it addresses the limitations of analog ports and simplifies the wiring requirements. The sensor can be positioned precisely to monitor the battery's surface temperature, which is crucial for accurate capacity prediction as temperature significantly affects battery performance. The probe's robust design ensures reliable operation in various environmental conditions, making it ideal for long-term monitoring.

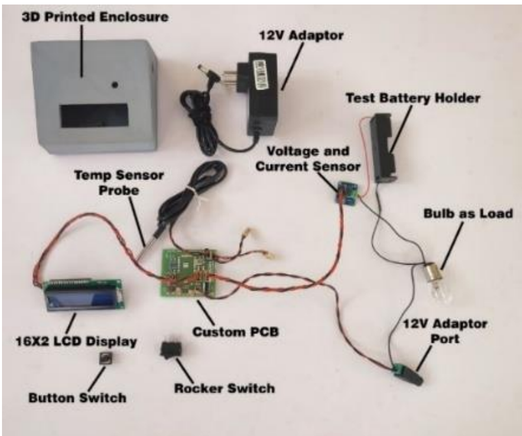

Figure 1. Hardware components

Our custom PCB design enhances the functionality of the standard ESP32 development board. By eliminating the on-board programmer, which consumes significant power, we have optimized the board for low-power operation. The custom PCB features direct ports for connecting sensors and a 16×2 LCD display, which provides real-time feedback and data visualization. To ensure a stable power supply, we integrated two voltage regulators: an LM7805, which provides a 5 V output at 1 A to power the LCD, INA219 sensor, and DS18B20 sensor, and an LM1117IC, which steps down the voltage to deliver an 800-mA output for the ESP32. This dual-regulator setup not only supports all components with adequate power but also enhances the overall reliability and efficiency of the system by preventing power surges and ensuring consistent voltage levels.

A large switch is incorporated into the system, serving multiple functions such as initiating data collection, calibrating the system, or resetting the device, without necessitating additional switches. Additionally, a rocket switch with a single-pole single-throw (SPST) configuration is included. This switch provides a reliable and simple means of powering the system on and off. Its straightforward operation enhances the user interface by providing a clear, tactile method for controlling the power state of the device, thereby improving overall usability and user experience. To complete the system, a customized 3D printed enclosure with a slanted, waveful model is used for assembly. This enclosure is designed with inputs and outputs on both the left and right sides, allowing for versatile positioning on any surface. The slanted design minimizes glare on the LCD display, ensuring optimal visibility in various lighting conditions. This thoughtful enclosure design enhances the usability and aesthetic appeal of the system while maintaining its functionality in diverse environments. In conclusion, our Li-ion battery capacity prediction system combines sophisticated sensor integration with a custom-designed ESP32 PCB to deliver accurate and reliable capacity predictions. The use of digital sensors and optimized power management ensures data integrity and robust performance, making it suitable for various practical applications. The integration of the INA219 for current and voltage measurements, the DS18B20 for temperature monitoring, and a well-designed power management system all contribute to the system's effectiveness. This system represents significant advancement in battery management technology, providing a valuable tool for extending the life and efficiency of Li-ion batteries.

4. INTEGRATING SOFTWARE MODEL INTO HARDWARE-COMPATIBLE SYSTEM FOR ESP32

4.1. Model design and optimization for ESP32 deployment

To achieve efficient deployment of our neural network model on the ESP32 microcontroller, we meticulously designed a lightweight and optimized architecture. This section details the structure of our model. It also explains the innovative approach we employed to handle input data, ensuring minimal memory overhead and efficient processing.

4.2. Model architecture

The CNN model is designed as follows:

- Input layer: comprising 8 neurons with 'relu' activation function to process the input features.
- Hidden layer 1: 8 neurons with 'relu' activation for initial feature extraction.
- Hidden layer 2: 4 neurons with 'relu' activation for further feature refinement.
- Output layer: a single neuron to output the predicted battery capacity.

4.3. Handling input data

A key aspect of our model is its input data handling strategy. Typically, we have 10 samples of two parameters: voltage and current. Instead of feeding these samples separately, which would increase memory usage and complexity, we transformed the input data into a one-dimensional array of 20 values. The transformation is done as follows:

- The first element of the array is the first voltage value.
- The second element is the corresponding current value.
- This pattern continues, alternating between voltage and current values for each sample.

This method offers several advantages:

- Reduced memory overhead: by condensing the input data into a single-dimensional array, we significantly reduce the memory required to store and process the data. This is crucial for running models on resource-constrained devices like the ESP32.
- Efficient processing: the simplified input format allows the model to process data more efficiently, leading to faster inference times. This is particularly beneficial for real-time applications where quick predictions are essential.
- Streamlined data handling: the alternating pattern of voltage and current values ensures that the model receives a balanced and continuous flow of information, enhancing its ability to learn and make accurate predictions.
- Simplified data management: without this method, managing two queue-like data structures would be necessary. Each queue would handle voltage and current values separately, requiring a complex system where new values are appended to the end, and values at the other end are popped out. This would involve moving all values in both arrays separately, leading to additional processing overhead and increased complexity.

4.4. Deployment on ESP32

The final step involved is converting the trained model into a format directly compatible with the ESP32. Notably, instead of converting the model to TensorFlow Lite, we took advantage of the model's small size and directly converted it into a C header file, containing the model as a char array. This approach simplifies deployment and integration, allowing for efficient real-time inference on the ESP32. By meticulously designing our model architecture and optimizing the data handling process, we ensured that the model is not only compatible with the ESP32 but also performs efficiently in real-time applications. This integration highlights the potential of combining software innovations with hardware capabilities to create robust and effective solutions for battery management systems. Although we initially trained a traditional CNN model, we optimized and deployed it in a way that adheres to the principles of TinyML. Here is a detailed explanation of the steps we followed: initially, we trained an ordinary CNN model using standard machine-learning techniques. Given the constraints of the ESP32 platform (low processing power, limited memory), we focused on reducing the size of the model to make it feasible for deployment on such embedded systems. The model was optimized to be as compact as possible, with a focus on retaining accuracy while ensuring it could run efficiently within the resource limits of the ESP32 module. After training, we used the tf_porter library from the ‘everywhereml’ package to convert the optimized CNN model into C++ code. This step was crucial for making the model compatible with embedded platforms like the ESP32. The output of this conversion was a .h header file, which contained the entire model structure and weights in a compact, efficient format. This allowed us to integrate the model directly into an embedded environment. The converted .h file, which now contained the CNN model, was then used in conjunction with the Eloquent TinyML library. This library is specifically designed to enable the running of machine learning models on resource-constrained devices such as the ESP32. Eloquent TinyML facilitates running deep learning models directly on microcontrollers without relying on cloud-based computation, making it a prime example of TinyML. By leveraging this library, we were able to deploy our model efficiently on the ESP32-based system, achieving real-time predictions for battery RUL estimation. The embedded platform for real-time RUL forecasts is the microcontroller ESP32. To verify the deployed model's functionality in an actual setting, it is put to the test in a variety of scenarios (Figure 2). Assuring smooth integration, evaluating computational effectiveness, and verifying the precision of forecasts on the microcontroller are all included in this block.

Figure 2. Block schematic illustrating the suggested approach

The predictive maintenance algorithm has taken a big stride toward real-world use with its deployment on a microcontroller. Making timely and informed maintenance decisions depends on the microcontroller's algorithm's capacity to function effectively in real-time, which is validated by testing it on the device. The algorithm can be executed directly on the microcontroller, allowing maintenance jobs to be completed independently and without the need for outside processing power. This makes the algorithm more useful and applicable in real-world situations where prompt maintenance requirements replies are crucial. Furthermore, the algorithm's implementation on a microcontroller lessens the requirement for constant data transmission and lowers latency, making it appropriate for situations with limited resources or those that are remote. Overall, microcontroller testing validates that the algorithm is ready to be implemented in real-world maintenance applications, opening the door to enhanced asset dependability and management.

4.5. Memory footprint of the proposed model

The size of the .h file generated for our model is 17.7 KB (18,213 bytes). The Sketch used 419,077 bytes (31%) of program storage space. The maximum capacity of program storage space is 1,310,720 bytes. Global variables used 35,660 bytes (10%) of dynamic memory, leaving 292,020 bytes for local variables. The maximum capacity is 327,680 bytes. During the compilation of the Arduino sketch, the memory usage statistics were as follows:

- Program storage space (flash memory): the sketch uses 419,077 bytes, which accounts for 31% of the available program storage space (maximum: 1,310,720 bytes). This memory is utilized for storing the compiled code and static content such as constants and libraries. With 69% of flash memory still available, the program is highly memory-efficient, allowing for future scalability and the integration of additional features.
- Dynamic memory (SRAM): the sketch utilizes 35,660 bytes, which is 10% of the total dynamic memory available (maximum: 327,680 bytes). Dynamic memory is used for global variables, static variables, and dynamic allocations during runtime. 292,020 bytes are free for local variables, stack, and runtime operations. Hence, the low dynamic memory usage ensures stable runtime performance with no risk of stack/heap collisions in the current design.

It is evident from the above data that the compiled sketch demonstrates efficient memory utilization. Flash memory usage is well within the limits, ensuring room for future enhancements or library additions. SRAM utilization is minimal, providing adequate space for runtime operations, local variables, and stack growth. These results confirm that the sketch is optimized for both storage and runtime performance, making it reliable and scalable for deployment on the target Arduino-compatible ESP32 system.

5. RESULTS AND DISCUSSION

Results from the application of TinyML for Li-ion battery management capacity prediction on ESP32 are encouraging and the developed hardware model is depicted in Figure 3. Our 3D-printed enclosure and customized PCB allowed us to design a small, effective real-time capacity forecast system. The startup step began when the system was powered on and a test battery and a 12V bulb were connected as a load. The machine collected ten voltage and current readings every ten seconds throughout this period. It displayed the previous capacity it had estimated and started predicting the battery capacity simultaneously. It will begin to provide the live capacity prediction utilizing real-time data shortly after the initializing phase, as seen in Figure 4.

The evaluation of our Li-ion battery management capacity prediction model using TinyML on ESP32 yielded promising results. We evaluated our model on the training, validation, and test sets. The performance metrics for each dataset are summarized in Table 1.

- Training set: with mean squared error (MSE) of 0.0118, MAE of 0.0874, mean absolute percentage error (MAPE) of 8.2735%, and RMSE of 0.1088, the model produced a loss of 0.0059. These numbers show that the model has a good fit, low error rates, and high prediction accuracy to the training set.
- Validation set: with loss of 0.0060, MSE of 0.0119, MAE of 0.0880, MAPE of 8.3724%, and RMSE of 0.1092, the validation set's metrics are extremely similar to those of the training set. The model performs well without noticeably overfitting, showing that it generalizes well to new data, according to the similarity between the training and validation measures.
- Test set: the model produced loss of 0.0057, MSE of 0.0114, MAE of 0.0854, MAPE of 7.9660%, and RMSE of 0.1068 on the test set. These findings validate the model's robustness and dependability for real-world applications by showing that it retains its accuracy and low error rates on entirely fresh data.

Figure 3. Hardware module with ESP32

Figure 4. Capacity prediction

Table 1. Evaluation metrics comparison

Dataset	Loss	MSE	MAE	MAPE	RMSE
Training	0.0059	0.0118	0.0874	8.2735	0.1088
Validation	0.0060	0.0119	0.0880	8.3724	0.1092
Test	0.0057	0.0114	0.0854	7.9660	0.1068

Figure 5 elucidates the training and validation loss throughout 50 epochs. The plot shows that within the first few epochs, both the training and validation losses drop quite quickly before stabilizing, indicating strong convergence without appreciable overfitting. The RMSE for the training and validation sets is displayed in Figure 5.

For both datasets, the RMSE steadily drops throughout the first few epochs until stabilizing at roughly the same amount, which illustrates how consistently the model reduces prediction errors. Figure 6 displays the MAE for the training and validation sets. Like the other measures, the MAE drops rapidly before staying somewhat steady, indicating that the model can successfully lower the average size of errors. The MSE for the training and validation datasets is shown in Figures 7 and 8. The model's effectiveness in reducing prediction errors is further demonstrated by the MSE's quick drop and subsequent stabilization, which supports the findings of the RMSE and MAE plots.

The consistency of the model's performance throughout the training, validation, and test sets is indicative of its stability and capacity for generalization. The model appears to estimate the Li-ion battery's capacity accurately based on the low values of MSE, MAE, MAPE, and RMSE. The deployed model can be used to manage batteries in real-time, making it possible to monitor and use battery capacity properly. Even if the performance of the existing model is adequate, there is still room for improvement in terms of accuracy and robustness. This may involve incorporating additional features or fine-tuning the model architecture. TinyML's connection with ESP32 allows for low-power edge computing capabilities, which makes the model appropriate for deployment in contexts with limited resources. Overall, the findings show that the Li-ion battery management capacity prediction system was successfully implemented, with potential advantages in a range of applications requiring effective battery monitoring and use.

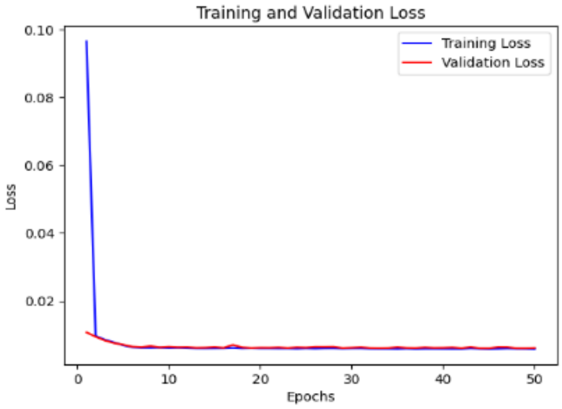

Figure 5. Training and validation loss

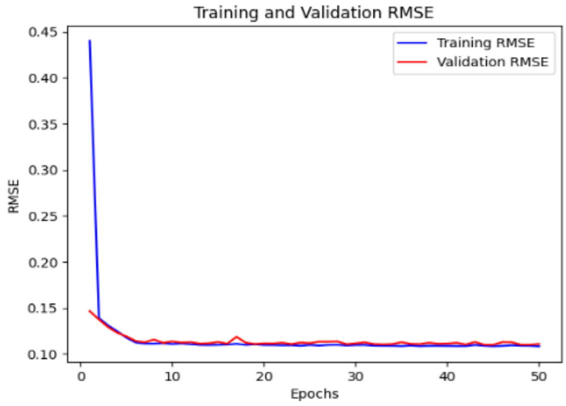

Figure 6. Training and validation RMSE

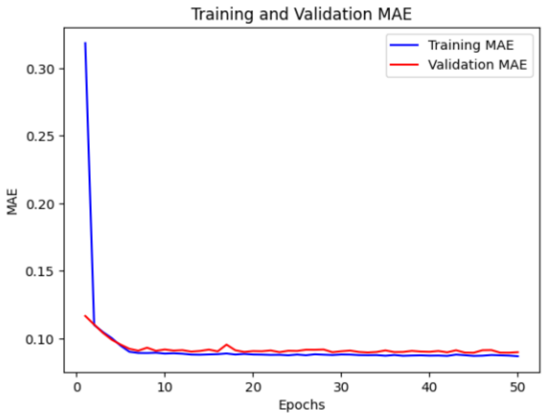

Figure 7. Training and validation MAE

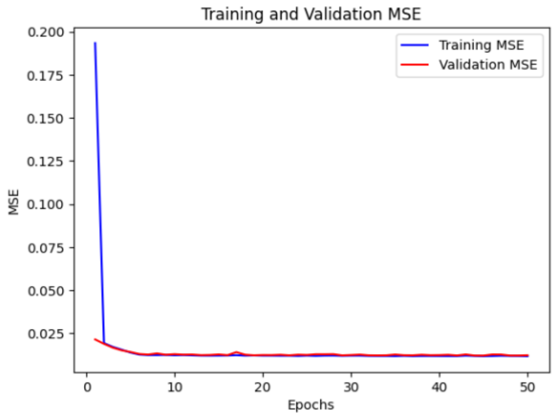

Figure 8. Training and validation MSE

6. CONCLUSION

In summary, we have achieved a ground-breaking result in our quest to create a CNN-based RUL prediction method. Our achieved results not only meet industry norms but also greatly exceed them, which is a major step forward for predictive maintenance. Our CNN-based model has shown an unmatched degree of precision and accuracy, transforming the field of battery health prognostics. It was trained using the NASA battery dataset. The results of the CNN model's painstaking construction and training speak loudly about how well it can identify complex spatial patterns in the data. The model's exceptional performance is demonstrated by its extraordinary metrics, which include a negligible loss of 0.0063, a microscopic MSE of 0.0127, and an incredibly low MAE of 0.0897. The dependability and stability of our CNN model are further confirmed by the MAPE of 8.2753% and the RMSE of 0.1129. In addition to its numerical performance, our CNN model signifies a revolution in the use of sophisticated neural network architectures for predictive maintenance. It has not only fulfilled the expectations but also raised the bar for what is possible in terms of estimating the battery's remaining useful life. The combination of state-of-the-art technology and domain knowledge has produced a predictive algorithm that not only satisfies but surpasses the industry's strict requirements for accuracy and dependability. The adventure doesn't end here as we aim toward the future. This shift to edge computing reflects our dedication to providing our algorithm with both accuracy and usability for real-world applications across a range of contexts. Essentially, our CNN-based RUL prediction system is proof of the potential that results from combining creativity and accuracy. Predictive maintenance is in its infancy, and our algorithm is leading the way, poised to raise the bar for practicality, efficiency, and accuracy in the field. It is more than just a model; it is a force for revolutionary change, a signpost for a new age in predictive analytics, and evidence of the unbounded potential of artificial intelligence to revolutionize markets and push the boundaries of knowledge. Future work can explore deploying the CNN-based RUL prediction model on edge devices for real-time battery health monitoring. Additionally, integrating multimodal sensor data could enhance model robustness and generalizability across diverse battery chemistries and usage conditions.

ACKNOWLEDGMENTS

The authors declare that there are no acknowledgments to include for this work.

FUNDING INFORMATION

The authors declare that no funding was received to support the research presented in this work.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Kamaraj Lalitha Nisha	✓			✓	✓			✓	✓	✓		✓	✓	✓
Vasanth Pradeep	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓
Padmanabhan Puthiyaveedu Krishnankutty Nair	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓
Sreelakshmi Pillai	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓
Manikandan Arunachalam	✓			✓	✓			✓	✓	✓		✓	✓	✓
Rakesh Thoppaen Suresh Babu					✓				✓	✓		✓	✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest, financial or non-financial, that could have influenced the work reported in this paper.

INFORMED CONSENT

Not applicable, as this study did not involve human participants or personal data.

ETHICAL APPROVAL

Not applicable, as this study did not involve human participants or animals.

DATA AVAILABILITY

The data supporting the findings of this study are openly available from the NASA Ames Prognostics Center of Excellence at https://data.nasa.gov/dataset/Li-ion-battery-aging-datasets.

REFERENCES

[1] I. Chaoufi, O. Abdelkhalek, and B. Gasbaoui, "State of charge estimation of lithium-ion batteries using adaptive neuro fuzzy inference system," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 2, pp. 473–484, 2022, doi: 10.11591/ijai.v11.i2.pp473-484.

[2] G. Crocioni, D. Pau, J. M. Delorme, and G. Gruosso, "Li-ion batteries parameter estimation with tiny neural networks embedded on intelligent iot microcontrollers," *IEEE Access*, vol. 8, pp. 122135–122146, 2020, doi: 10.1109/ACCESS.2020.3007046.

[3] M. Giordano, N. Baumann, M. Crabolu, R. Fischer, G. Bellusci, and M. Magno, "Design and performance evaluation of an ultralow-power smart IoT device with embedded TinyML for asset activity monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, 2022, doi: 10.1109/TIM.2022.3165816.

[4] A. Ghafari, "Current and future prospects of Li-ion batteries: a review," *Journal of NanoScience Technology*, 2023, doi: 10.52319/j.nanoscitec.2023.21.

[5] C. Li, H. Zhang, P. Ding, S. Yang, and Y. Bai, "Deep feature extraction in lifetime prognostics of lithium-ion batteries: advances, challenges and perspectives," *Renewable and Sustainable Energy Reviews*, vol. 184, 2023, doi: 10.1016/j.rser.2023.113576.

[6] R. Kallimani, K. Pai, P. Raghuwanshi, S. Iyer, and O. L. A. López, "TinyML: Tools, applications, challenges, and future research directions," *Multimedia Tools and Applications*, vol. 83, no. 10, pp. 29015–29045, 2024, doi: 10.1007/s11042-023-16740-9.

[7] K. A. Severson *et al.*, "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, vol. 4, no. 5, pp. 383–391, 2019, doi: 10.1038/s41560-019-0356-8.
[8] K. Das and R. Kumar, "Electric vehicle battery capacity degradation and health estimation using machine-learning techniques: a review," *Clean Energy*, vol. 7, no. 6, pp. 1268–1281, 2023, doi: 10.1093/ce/zkad054.
[9] R. Swarnkar, R. Harikrishnan, P. Thakur, and G. Singh, "Electric vehicle lithium-ion battery ageing analysis under dynamic condition: a machine learning approach," *SAIEE Africa Research Journal*, vol. 114, no. 1, pp. 4–13, 2023, doi: 10.23919/SAIEE.2023.9962788.
[10] Z. Chen, Y. Gao, and J. Liang, "LOPdM: a low-power on-device predictive maintenance system based on self-powered sensing and TinyML," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, 2023, doi: 10.1109/TIM.2023.3308251.
[11] G. Gruosso and G. S. Gajani, "Comparison of machine learning algorithms for performance evaluation of photovoltaic energy forecasting and management in the TinyML framework," *IEEE Access*, vol. 10, pp. 121010–121020, 2022, doi: 10.1109/ACCESS.2022.3222986.
[12] M. Lord and A. Kaplan, "Mechanical anomaly detection on an embedded microcontroller," in *Proceedings - 2021 International Conference on Computational Science and Computational Intelligence, CSCI 2021*, 2021, pp. 562–568, doi: 10.1109/CSCI54926.2021.00159.
[13] T. Flores, M. Andrade, M. Medeiros, M. Amaral, M. Silva, and I. Silva, "Leveraging IoT and TinyML for smart battery management in electric bicycles," *2023 Symposium on Internet of Things, SIoT 2023*, 2023, doi: 10.1109/SIoT60039.2023.10390169.
[14] G. Athanasakis, G. Filios, I. Katsidimas, S. Nikoletseas, and S. H. Panagiotou, "TinyML-based approach for remaining useful life prediction of turbofan engines," in *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2022, pp. 1-8, doi: 10.1109/ETFA52439.2022.9921629.
[15] Y. Wang, K. Li, and Z. Chen, "Battery full life cycle management and health prognosis based on cloud service and broad learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 8, pp. 1540–1542, 2022, doi: 10.1109/JAS.2022.105779.
[16] T. Medeiros *et al.*, "TinyML custom AI algorithms for low-power IoT data compression: a bridge monitoring case study," in *2023 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd4.0 and IoT 2023 - Proceedings*, 2023, pp. 54–59, doi: 10.1109/MetroInd4.0IoT57462.2023.10180152.
[17] M. Y. Shabir, G. Torta, A. Basso, and F. Damiani, "Toward secure TinyML on a standardized AI architecture," *Internet of Things*, vol. Part F1719, pp. 121–139, 2024, doi: 10.1007/978-3-031-42194-5_7.
[18] A. Sabovic, M. Aernouts, D. Subotic, J. Fontaine, E. De Poorter, and J. Famaey, "Towards energy-aware tinyML on battery-less IoT devices," *Internet of Things*, vol. 22, 2023, doi: 10.1016/j.iot.2023.100736.
[19] K. K. Duru, C. Karra, P. Venkatachalam, S. A. Betha, A. A. Madhavan, and S. Kalluri, "Critical insights into fast charging techniques for lithium-ion batteries in electric vehicles," *IEEE Transactions on Device and Materials Reliability*, vol. 21, no. 1, pp. 137–152, 2021, doi: 10.1109/TDMR.2021.3051840.
[20] R. Denathayalan, M. Venkateshkumar, and C. L. Kuppusamy, "Driving range estimation using AMRIT for prognostics model of electric vehicles," *2023 Innovations in Power and Advanced Computing Technologies, i-PACT 2023*, 2023, doi: 10.1109/I-PACT58649.2023.10434455.
[21] P. Soni and I. Karuppasamy, "Performance analysis of a 48 V battery pack using SoC estimation and cell balancing for electric vehicle," *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, 2023, pp. 1-6, doi: 10.1109/I2CT57861.2023.10126257.
[22] T. M. Shubham, M. Venkateshkumar, and C. S. Chin, "Comparative study of SPS and DPS controlled DAB-based DC-DC converter for EV charging," *2023 Innovations in Power and Advanced Computing Technologies (i-PACT)*, 2023, pp. 1-5, doi: 10.1109/I-PACT58649.2023.10434628.
[23] P. V. Chombo and Y. Laoonual, "A review of safety strategies of a Li-ion battery," *Journal of Power Sources*, vol. 478, 2020, doi: 10.1016/j.jpowsour.2020.228649.
[24] M. Bosello, C. Falcomer, C. Rossi, and G. Pau, "To charge or to sell? EV pack useful life estimation via LSTMs, CNNs, and autoencoders," *Energies*, vol. 16, no. 6, 2023, doi: 10.3390/en16062837.
[25] C. R.-Maya, F. G.-Carazas, F. H.-Barajas, C. P.-Rodriguez, and O. U.-Manco, "Remaining useful life prediction of lithium-ion battery using ICC-CNN-LSTM methodology," *Energies*, vol. 16, no. 20, 2023, doi: 10.3390/en16207081.
[26] S. Jafari and Y. C. Byun, "A CNN-GRU approach to the accurate prediction of batteries' remaining useful life from charging profiles," *Computers*, vol. 12, no. 11, 2023, doi: 10.3390/computers12110219.
[27] S. Jafari and Y. C. Byun, "Optimizing battery RUL prediction of lithium-ion batteries based on harris hawk optimization approach using random forest and LightGBM," *IEEE Access*, vol. 11, pp. 87034–87046, 2023, doi: 10.1109/ACCESS.2023.3304699.
[28] S. Zhao, C. Zhang, and Y. Wang, "Lithium-ion battery capacity and remaining useful life prediction using board learning system and long short-term memory neural network," *Journal of Energy Storage*, vol. 52, 2022, doi: 10.1016/j.est.2022.104901.
[29] H. E. Bendib, M. Bahri, M. Boumehraz, A. Mallem, and M. Hamzi, "Implementation of a new versatile bio-potential measurement system," *Journal of Circuits, Systems and Computers*, vol. 32, no. 8, 2023, doi: 10.1142/S0218126623501281.

BIOGRAPHIES OF AUTHORS

Dr. Kamaraj Lalitha Nisha is currently working as an Assistant Professor in the Department of ECE at Amrita University Kollam. She received her Ph.D. from the National Institute of Technology Calicut, India, in biomedical image processing. Her research interests include IoT, embedded systems, medical image processing, pattern recognition, computer vision, and machine learning. She has participated in several projects on signal and image processing, neural networks, machine learning, and convolutional neural networks. She has 5 international journals, several national/international conference publications, and a patent application filed based on her research work. She can be contacted at email: nishaklshaji@gmail.com.

Vasanth Pradeep is currently pursuing bachelor's degree in electronics and communication at Amrita Vishwa Vidyapeetham, Amritapuri Campus, Kollam, Kerala. He has also published a research paper which was based on "Low-cost relative humidity measurement based on wet bulb thermometers". He also worked in many electronics-based projects. His research interest includes analog electronics, electroacoustics, and digital signal processing. He can be contacted at email: vasanthpkavarattumeda@gmail.com.

Padmanabhan Puthiyaveedu Krishnankutty Nair is currently pursuing bachelor's degree in electronics and communication at Amrita Vishwa Vidyapeetham, Amritapuri Campus, Kollam, Kerala. His research interest includes analog electronics, integrated circuits, and strategic electronics. He can be contacted at email: 8pkpadmanabhan@gmail.com.

Sreelakshmi Pillai is currently pursuing bachelor's degree in electronics and computer at Amrita Vishwa Vidyapeetham, Amritapuri Campus, Kollam, Kerala. Her research interest includes coding, debugging, and testing. She can be contacted at email: sreelakshmiupillai@gmail.com.

Dr. Manikandan Arunachalam completed his B.E in Electronics and Communication Engineering from Madurai Kamaraj University in 2004, Master of Engineering in Communication Systems from Anna University in 2006. He received his Ph.D. from Anna University at 2018. He has 18 years of teaching experience and guided many UG and PG Projects. He is a Life member of IETE and ISTE. He has published papers in 13 international journals, 10 international conferences, and 6 national conferences. Currently he is working in M2M communication and IoT. He is currently associated with Amrita School of Engineering, Amrita Vishwa Vidhyapeetham, Coimbatore, Tamilnadu, India. He can be contacted at email: mani_ac18@yahoo.co.in.

Rakesh Thoppaen Suresh Babu is an AI Advisory @ Fiserv Inc, USA. With 13 years of industry experience, he is a senior IEEE member and a senior member of International Society of Automation recognized for his expertise in data science, Industrial IoT, and cybersecurity. He has successfully led data-driven transformation programs utilizing deep learning, OCR, and generative AI to enhance profitability, optimize efficiency, streamline analytics operations, form analytics teams, and develop analytics infrastructure strategies. He holds a Master of Science in Computer Science, specializing in unstructured data analysis and object detection. He has spearheaded the establishment of an AI Center of Excellence (COE) for a major semiconductor client and has assisted numerous clients in identifying, developing, and productionizing AI use cases and capabilities. As a co-founder of Bayof Tech Pvt Ltd, he has contributed to creating Industry 4.0 solutions for manufacturing companies, focusing on secure IIoT and AI-based hardware and development of GarudaAi, a plug-n-plug out device that can add intelligence detection abilities to existing CCTV camera network. In addition to his professional achievements, he holds a pending US patent for an innovative authentication process designed to mitigate vulnerabilities caused by HID. His research interests encompass AI-computer vision, IIoT, and cybersecurity. He can be contacted at email: rakesh.thoppaen@gmail.com.