Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Deploying machine learning for remaining useful life prediction of lithium-ion batteries on Coral Dev Board

Mariem Boujneh

boujnehmariem@yahoo.fr

National Engineering School of Tunis

Nesrine Majdoub

National Engineering School of Tunis

Taoufik Ladhari

National Engineering School of Tunis

Ali Serdar Atalay

Enterprise Estonia (Estonia)

Anis Sakly

National Engineering School of Tunis

Research Article

Keywords: Lithium-ion batteries, RUL prediction, ML algorithms, Coral Dev

Posted Date: September 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10618105/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Deploying machine learning for remaining useful life prediction of lithium-ion batteries on Coral Dev Board

Abstract

Lithium-ion batteries have garnered significant interest recently regarding their critical utility in several fields, covering storage for renewable energy and electric vehicles. The precise estimation of their remaining useful life (RUL) is critical to optimize their effectiveness and guarantee reliable use. This paper will present a data-based methods for predicting the RUL of lithium-ion batteries in EVs. Specifically, this study evaluates the performance of multiple machine learning (ML) techniques, such as Random Forest, XGBoost, and Temporal Convolutional Networks (TCN). These models are deployed on the Coral Dev Board to assess their practical applicability. A comprehensive comparative analysis is conducted using multiple evaluation metrics to identify the most effective strategy. The findings aim to support the development of predictive maintenance techniques in the battery industry by highlighting the most promising ML-based approach for battery health forecasting.

Keywords: Lithium-ion batteries, RUL prediction, ML algorithms, Coral Dev

1 Introduction

Battery technology, which offers clean and effective power for a range of applications, is setting the standard for sustainable energy solutions. Lithium-ion batteries in particular have gained popularity for powering a range of applications because of their benefits, which include high capacity,quick charging, and lower self-discharge rate [35], [23], [28]. To track and ensure the dependability of batteries after prolonged operation, several crucial indicators, such as the State of Charge (SOC), State of Energy (SOE), State of Temperature (SOT), State of Power (SOP), State of Health (SOH), and Remaining Useful Life (RUL), are used [12].
SOH and RUL are particularly significant as they provide clear insights into the battery's degradation in use [25]. These parameters are commonly defined in relation to a battery's capacity. RUL calculates the amount of time or cycles left before the battery's capacity deteriorates to predefined limit, whereas SOH reflects the battery's present state or performance level in relation to its initial capacity. These metrics are

essential for both manufactures and users to optimize battery utilization, plan maintenance activities, and develop effective replacement strategies. Accurate and timely RUL prediction is essential to improving battery longevity, operational reliability, and health-based maintenance planning [10].

Battery SOH and RUL estimation approaches are broadly divided into two principal groups: model-based and data-based techniques [19].

Model-based methods employ prior information and electrochemical reaction rules under restrictions. Common methods include electrochemical models [17], [15], which are grounded in battery aging theory, and equivalent circuit models (ECM) [3], [16], which simplify implementation by using fewer parameters. While electrochemical models are highly accurate, real-time RUL estimation is difficult due to their complexity. On the other hand, ECM are more useful but have lower estimating accuracy. Although model-based techniques have advanced significantly, none of them are able to adequately represent the dynamic and diverse responses of lithium-ion batteries within various working situations [26]. This restriction has prompted a move toward data-driven approaches, especially machine learning (ML) techniques, which do not require explicit physical equations and may adaptively predict complex deterioration patterns.

The emergence of data-based approaches to SOH forecasting has been made easier by the growth of big data technologies. Unlike the previously mentioned methods, these techniques don't involve a thorough comprehension of the system's physical and chemical workings. Rather, they examined the data using machine learning and statistical methods to detect deteriorating patterns [31].

Recent studies have employed various machine learning techniques for RUL prediction, including Support Vector Machines (SVM) [20], [9], Artificial Neural Networks (ANN) [29], Random Forest (RF) [27], XGBoost [14], Bayesian Neural Networks (BNN) [18] , and Long Short-Term Memory (LSTM) [11]. These methods offer diverse advantages, ranging from handling non-linear dependencies to capturing long-term temporal trends in battery degradation. A cross-stitch neural network architecture is proposed by Yao et al. [32] for predicting the SOC and SOH of battery technologies. By sharing information across jobs, the system increases estimation accuracy. Multi-scale data processing also helps the method better capture dynamic characteristics of battery state changes. Chang et al. [6] propose a hybrid technique that combines LSTM networks with Light Gradient Boosting Machines (LightGBM) to estimate the RUL and identify the best recycling locations for lithium-ion batteries. The proposed approach uses a sliding window technique to efficiently capture temporal degradation patterns and Ensemble Empirical Mode Decomposition (EEMD) for data preprocessing. In [20], the authors investigate data-driven approaches, including SVM, to predict the RUL and health diagnostics of lithium-ion batteries. The study shows how data-driven approaches can improve battery management systems by offering credible, up-to-date information on the longevity and health of batteries. To forecast the RUL of lithium-ion batteries, [34] suggested a hybrid deep learning framework combining LSTM and Gated Recurrent Unit (GRU). According to the findings, both LSTM and GRU models can effectively grasp the complex and nonlinear degradation of batteries, thereby producing accurate RUL predictions essential for battery management systems. [30]

introduced a novel residual longevity estimation approach for lithium-ion batteries that integrates data preprocessing techniques with previous knowledge-driven CNN-LSTM. This approach uses advanced data cleaning and feature extraction to improve input quality, while the hybrid technique effectively captures both temporal and spatial deterioration patterns. The suggested method's excellent accuracy and robustness are demonstrated by outcomes of experiments, allowing for improved battery management and maintenance scheduling. A random forest-based technique that uses limited deterioration history data to anticipate the RUL of lithium-ion batteries was proposed by [33]. The method makes use of random forests ensemble learning power to efficiently handle tiny datasets and capture intricate nonlinear degradation patterns, allowing for precise RUL estimation. A novel SOC-SOH estimation architecture comprising charging and SOH encoders and decoders was presented in [13]. The suggested framework achieved an MAE of 0.41 % for SOH estimate and 0.362 % for SOC estimation on the test set.

In addition to advancements in ML algorithms, practical deployment of these models is increasingly performed on edge computing devices. Among the most widely utilized platforms is the Raspberry Pi, which supports ML inference without requiring additional hardware [22]. Its lightweight design and compatibility with frameworks such as TensorFlow Lite make it a cost-effective option for on-device intelligence.

Recognizing the growing demand for efficient AI solutions, NVIDIA developed the Jetson series, specially designed for embedded applications [21]. It offers high-performance computing capabilities while maintaining low power consumption and enhanced energy efficiency.

Another notable single-board computer optimized for ML inference is the Google Coral Dev Board [7]. Equipped with a specialized Tensor Flow Unit (TPU), it enables fast and energy-efficient execution of ML models. Its architecture is designed to maximize efficiency, makes it possible to implement ML algorithms directly on edge devices.

In this study, three ML algorithms, RF, XGBoost, and TCN, will be compared in order to accurately estimate the RUL of Lithium-ion batteries. These models wil be implemented on the Coral Dev Board, taking advantage of its hardware acceleration to accomplish efficient inference, in order to evaluate their practicality. Performance will be evaluated using a variety of metrics, such as energy consumption, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

The remainder of the paper is structured with the following contents: The battery dataset and data preprocessing procedures are described in depth in Section 2. The methodology, covering the machine learning algorithms and their setups, is explained in Section 3. The experimental setting, assessment measures, and a thorough analysis on the experimental findings are presented in Section 4. The paper is finally concluded in Section 5, which also suggests future research areas.

2 Data analysis

To establish a strong foundation for our study, we first introduce the primary dataset that underpins the analysis. Understanding the characteristics of this battery dataset is essential before delving into the preprocessing techniques applied.

2.1 Battery dataset

This study uses real-world electric vehicle (EV) battery charging data, supplied by [8], which was extracted from 20 EVs over a period of around 29 months. These vehicles, all of the BAIC EU500 model, are equipped with CATL lithium-ion battery packs. The dataset repository contains several CSV files, each of which represents an individual vehicle and detailing its charging sessions. Each file includes key features that characterize battery charging behavior and degradation patterns, such as:

- *Record_time*: date and time of each measurement enabling the analysis of charging profiles over time.
- *State_of_charge Soc (%)*: the percentage of available battery capacity at a given point, which is used to monitor the progress of charging.
- *Pack_voltage (V)*: the total voltage measured across the battery pack.
- *Charge_current (A)*: the current supplied to the battery during charging process, influencing charge rate and efficiency.
- *Max and Min cell voltage (V)*: the highest and lowest voltage values observed in each of the battery pack's single cells.
- *Max and Min temperature (°C)*: the highest and lowest temperature readings recorded from battery temperature sensors.
- *Available_capacity (Ah)*: the estimated usable capacity of the battery.

Figure 1 displays the mean and median results of the battery pack capacity calculations for three vehicles.

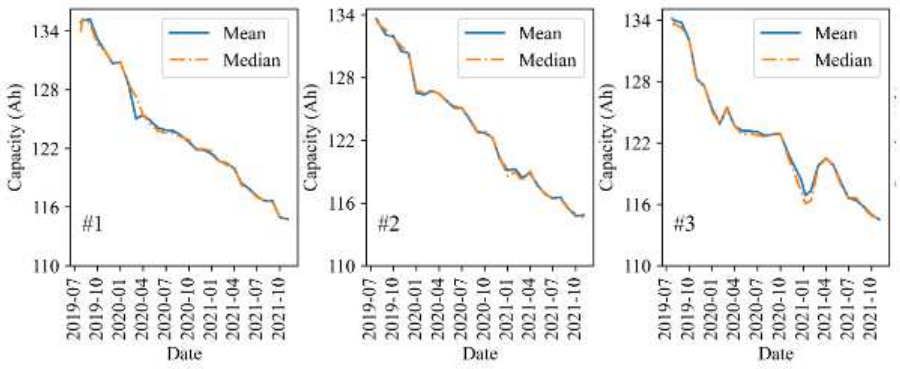

Fig. 1 Capacity curves of battery pack for vehicles 1, 2, and 3.

2.2 Data preprocessing

When working with real-world sensor data, such that gathered from electric vehicles, data preparation is essential to guaranteeing the dependability and accuracy of machine learning models. In this study, several preprocessing steps were applied to the raw charging data in order to generate consistent, clean, and informative inputs for Remaining Useful Life (RUL) prediction.

- *Data cleaning:* Raw datasets often contain missing entries, duplicated records, or anomalous values due to sensor failures or communication errors. To address these issues, we removed duplicates, filtered out outliers using domain-informed thresholds (e.g., voltage and current ranges), and imputed missing values via interpolation methods.
- *Feature engineering:* One of the key objectives in this study is to estimate the battery's RUL. Since this parameter is not directly available in the dataset, it is computed using the following relation [20]:

$$\text{RUL} = \frac{C_{\text{available}} - C_{\text{EOL}}}{C_{\text{init}} - C_{\text{EOL}}}$$

  where C_{init} is the nominal initial capacity (set to 1200 Ah), and C_{EOL} is the end-of-life threshold (set to $0.7 \times C_{\text{init}}$). The resulting RUL provides a normalized indication (between 0 and 1) of the remaining charge-discharge cycles before the battery becomes unusable.
- *Interpolation:* In cases where time-series gaps existed due to irregular logging or sensor dropouts, linear interpolation was used to estimate missing values. This method allowed us to maintain temporal continuity, which is especially crucial for models like TCN and CNN that rely on sequential input data.
- *Smoothing:* To reduce the effect of high-frequency noise present in the raw signals (e.g., voltage and current fluctuations), we applied the Savitzky-Golay filter. This smoothing technique fits successive windows of the data with low-degree polynomials, preserving the overall trends while eliminating outlier spikes that can mislead learning algorithms.
- *Normalization:* To ensure that features with large ranges do not dominate those with smaller scales, all continuous variables were normalized. Two strategies were tested: Min-Max normalization (scaling to [0,1]) and Z-score standardization (zero mean and unit variance). For the final model training, Min-Max scaling was selected due to its superior performance in preliminary experiments.
- *Sequence generation:* For deep learning models, particularly CNN and TCN, the data was segmented into overlapping sequences using sliding windows. Each input sequence consisted of a fixed number of time steps (window length), capturing the temporal evolution of attributes like SOC, temperature, voltage, and current. The goal output was the matching RUL at the end of each window.
- *Data splitting:* The final processed data was divided into subgroups for testing and training. Eighty percent of the sequences were utilized to train the models, while the remaining twenty percent were set aside for evaluation, according to the 80/20 ratio.

3 Methodology

This section outlines the methodological framework, describing the machine learning algorithms employed and their implementation details. Figure 3 shows the complete data flow diagram of the RUL estimation process.

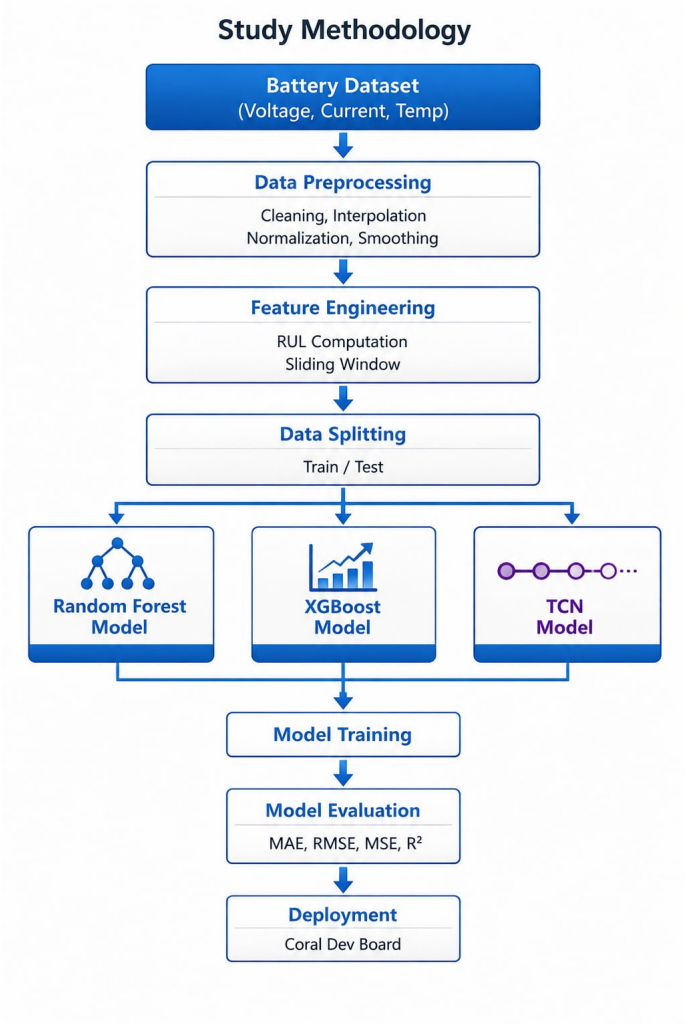

Fig. 2 Data flow diagram of RUL prediction process

3.1 Random Forest

The RF is an advanced machine learning technique that can estimate the RUL and SOH of lithium-ion batteries and is employed in regression and classification analysis. During training, the RF method, illustrated in Figure 3.1, constructs several decision trees and accumulates their forecasts by aggregating regression outputs. This results in superior precision, robustness, and adaptability than individual decision trees [24].
The RF method can be expressed mathematically as follow [4] :

$$\hat{y} = \frac{1}{K} \sum_{k=1}^{K} \hat{y}_k \tag{1}$$

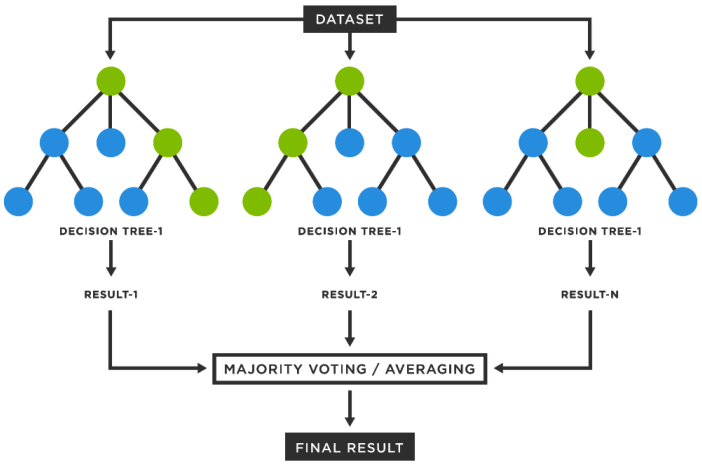

Fig. 3 Random Forest algorithm

where $\hat{y}$ is the aggregated forecast given by the RFand k denotes the number of decision trees. By merging the separate predictions $\hat{y}_k$ from each decision tree, the RF makes a prediction $\hat{y}$ for a given input data.

Key characteristics and advantages of RF can be summarized as follows:

* *Ensemble learning and robustness:* RF reduces over fitting and improves robustness against anomalies and noise that are typical in real-world data by combining outputs from many trees trained on random information and variants of features.
* *Non-linear capability:* Without requiring specific assumptions regarding the distribution of data, RF is able to efficiently describe intricate and non-linear correlations between battery characteristics and their deterioration patterns.
* *Interpretability:* The framework automatically generates feature importance metrics, allowing the model predictions to be explained and providing insights into the most important variables influencing battery health.

3.2 XGBOOST

XGBOOST (figure 3.2) is a ML algorithm that uses gradient boosting for its advanced, flexible, and highly efficient performance. Its superior predictive performance and robustness make it a popular tool for guided learning activities, like regression and classification. By building a sequence of decision trees one after the other, XGBOOST lowers an identified goal by training each new tree to rectify the faults of the preceding ensemble [2].
In mathematical terms, XGBoost integrates a differentiable loss function with an element of regularization which penalizes model complexity to optimize an objective

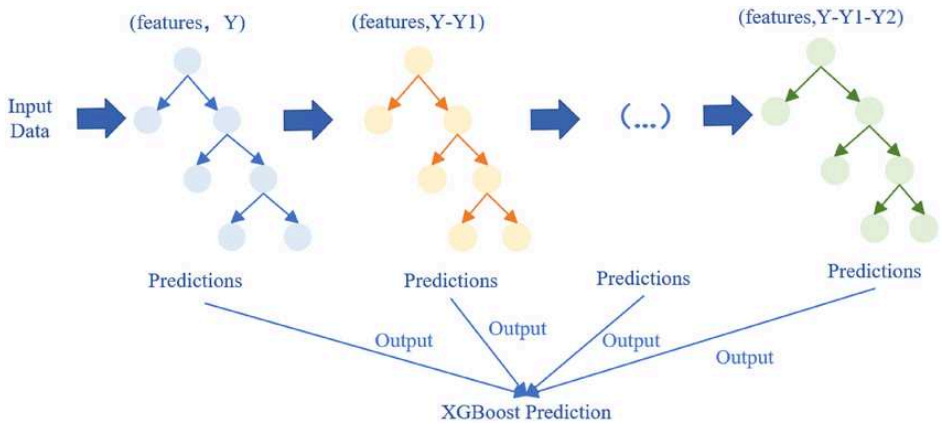

Fig. 4 XGBoost algorithm

function. The objective function is given by equation 2 [5]:

$$Obj = \sum_{i=1}^{N} L(y_i, \hat{y}_i^{(t)}) + \sum_{I=1}^{K} \Omega(f_i) \tag{2}$$

Where y_i stands for the real data for the i-th observer, $\hat{y}_i^{(t)}$ for the expected measure, $L(y_i, \hat{y}_i^{(t)})$ for the loss function associated with tree t, N for the quantity of observations in the dataset used to train, and K for the entire amount of distinct decision trees in the groups. The term $\Omega(f_i)$ refers to the term that uses regularization to control the level of complexity of the tree function.

Key characteristics of XGBoost include:

* *Regularization:* Integrates both L1 and L2 regularizing techniques to effectively manage model complexity and mitigate over-fitting.
* *Handling missing data:* Automatically learns optimal splits for missing values.
* *Parallel processing :* Utilizes parallel computation for accelerated model training by distributing the workload across multiple CPU cores.
* *Tree pruning:* Implements a depth-first tree pruning strategy that removes branches based on a complexity parameter, resulting in simpler trees that generalize better by avoiding over-complexity.

3.3 TCN

TCN is a specific type of deep neural networks model designed specifically to serve applications involving sequence modeling and time-series prediction.

The TCN architecture is characterized by two primary characteristics :

1/ *Causal Convolutions :* guarantee that the results at time t is a consequence of the information provided at step t and earlier, which preserves the causality and temporal order that are crucial for time-series modeling. This is a key characteristic that

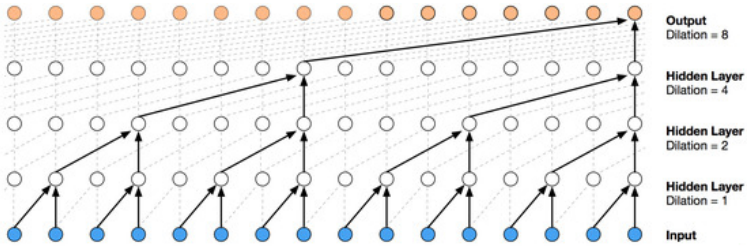

Fig. 5 TCN algorithm

sets TCNs apart from conventional convolutional netwroks, which have symmetric receptive fields and can break causality.

2/ *Dilated Convolutions :* are used in TCNs to effectively capture long-range temporal relationships by creating dilations (or gaps) between filter components. Without boosting the number of variables or the network's depth, this architecture allows the receptive field size to grow exponentially, allowing the algorithm to learn short-as well as long-term temporal patterns.

A dilated convolution operation F on a $1D$ series input x with a filter size k and a dilation factor d may be expressed mathematically as follows:

$$F(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i} \tag{3}$$

Key characteristics and advantages of TCN can be summarized as follows:

* Maintaining temporal order through strictly causal structure.
* Dilated convolutions provide an important receptive field that makes long-term dependency analysis possible.
* Training that is steady and effective thanks to residual connections.
* Performance is faster with parallelizable architecture than with recurrent models.

4 Results and discussion

The experimental methods and criteria used to assess machine learning algorithms are described in detail in this section. The experimental setting is first described, followed by the measures and findings that inform our comparative analysis.

4.1 Experimental setup: Coral Dev Board

For the deployement of the machine learning models, we used the Google Coral Dev Board, which is designed for edge AI applications. This boar integrates a dedicated Edge TPU, an ASIC specialized for accelerating deep learning inference, particularly TensorFlow Lite models [1] .
As a co-processor in real-world applications, the Edge TPU performs matrix computations frequently seen in neural networks remarkably effectively. This hardware acceleration lowers latency and energy overhead by enabling us to do inference locally

on the device rather than on cloud resources. The board runs Mendel Linux and has complete support for the PyCoral and TensorFlow Lite libraries to make programming easier. The main characteristics of the Coral Dev Board used in this work are summarized in Table 1 [7].

Table 1 Coral Dev Board specifications

Component	**Specification**
CPU	NXP i.MX 8M, Quad-core Cortex-A53 @ 1.5 GHz (ARMv8-A)
AI Accelerator	Google Edge TPU (up to 4 TOPS, ∼2 TOPS/W)
Memory	1 GB LPDDR4 @ 1600 MHz
Storage	8 GB eMMC, expandable via MicroSD
Operating System	Mendel Linux
Supported Frameworks	TensorFlow Lite, PyCoral

4.2 Evaluation indicators

In order to thoroughly evaluate the suggested machine learning models' predicted accuracy and dependability, we use a number of recognized assessment measures. With their ability to capture both average error magnitude and variance explanation, these indicators provide complementary viewpoints on model performance. These measures were chosen because they include:

- **Mean Absolute Error (MAE):** quantifies the mean size of prediction errors, regardless of their direction. It gives an easy-to-understand indicator of how well predictions match actual data. Its mathematical formulation is given in equation4 :

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

- **Mean Squared Error (MSE):** calculates the mean of the squared deviations between the actual and anticipated results.It is sensitive to variations since it penalizes greater errors more severely. Its mathematical formulation is given in equation 5:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5}$$

- **Root Mean Squared Error (RMSE):** is the square root of MSE, bringing the error metric back to the target variable's unit. It offers a more interpretable measure of prediction deviation. Its mathematical formulation is given in equation 6 :

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6}$$

- **Coefficient of Determination (R^2 Score):** evaluates the amount of variance in the variable of interest that can be predicted from the independent factors. It reflects the model?s explanatory power. Its mathematical formulation is given in equation 7 :

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{7}$$

Where :

- y_i is the actual measure;
- $haty_i$ is the anticipated value;
- n is the number of observations.

4.3 Experimental results

4.3.1 Model Parameters and Settings

To ensure fair and optimized comparison across models, each algorithm was configured with carefully selected hyperparameters. Table 2 summarizes the configuration used for Random Forest, XGBoost, and TCN during the RUL prediction experiments. These values reflect a balance between maximizing predictive accuracy and maintaining computational feasibility, especially considering the deployment constraints of the Coral Dev Board.

Table 2 Algorithm Parameters Used in Experiments

Algorithm	**Parameter**	**Value**
Random Forest (RF)	Number of trees (`n_estimators`)	100
	Maximum tree depth (`max_depth`)	10
	Minimum samples split (`min_samples_split`)	2
	Minimum samples leaf (`min_samples_leaf`)	1
XGBoost	Number of estimators (`n_estimators`)	100
	Maximum tree depth (`max_depth`)	5
	Learning rate (`learning_rate`)	0.3
	Subsample	0.6
TCN	Number of layers	4
	Filter size (kernel size)	3
	Dilation factors	1, 2, 4, 8
	Residual connections	Enabled
	Learning rate	0.001
	Batch size	32

4.3.2 Experimental results and discussion

RF, XGBOOST, and TCN were implemented and run on the Google Coral Dev Board to assess the efficacy and applicability of machine learning models for calculating the RUL of lithium-ion batteries. The deployment procedure involved transforming the

learned models into a compatible format with the TensorFlow Lite framework, allowing for hardware accelerated inference on the Coral Dev Board. This configuration enabled a thorough assessment of inference speed, energy consumption, and prediction accuracy, all of which are critical components for real-time battery surveillance applications.
The assessment measures mentioned before were used to assess the three models. The findings, which are presented in Table 3 below, show how the three methods trade off accuracy and computing efficiency.

Table 3 Experimental Evaluation Metrics for the ML Models

Model	**MSE**	**RMSE**	**MAE**	R^2 **Score**	**Energy consumption(%)**
RF	0.2000	0.4472	0.3500	0.500	89.1
XGBoost	0.1800	0.4243	0.1600	0.650	83.6
TCN	0.1432	0.3784	0.5394	-0.002	72.1

Applying RF, XGBOOST, and TCN to the RUL estimation of lithium-ion batteries on the Coral Dev Board reveals the trade-offs between accuracy, robustness, and computing efficiency.
Based on the results, XGBOOST had the lowest $MAE(0.16)$ and competitive $RMSE(0.42)$, which translated into a $R2score$ of 0.65, indicating the highest overall prediction accuracy. This shows that XGBOOST has a good computing footprint and is especially good at detecting non-linear deterioration patterns. Regularization and gradient boosting enable XGBOOST to generalize effectively across the dataset's varied charging and degradation situation. It also balanced precision and efficiency with a reasonable energy consumption of 83.6%.
On the other hand, RF showed a lesser explanatory power ($R2 = 0.50$) and greater error values ($MAE = 0.35, RMSE = 0.45$). Despite its reputation for interpretability and resilience, RF's ensemble of unpruned trees has a tendency to record local fluctuations without fully using long-term temporal connections. This restriction is especially noticeable in battery health forecasts, where the dynamics of sequential deterioration are crucial. Furthermore, RF appears to be less appropriate for resource-constrained edge devices given its greater energy consumption of 89.1%
For TCN ther performance was mixed. In terms of short-term prediction ability, it has the lowest $MSE(0.1432)$ and $RMSE(0.38)$. Instability in capturing general deterioration patterns is revealed by the surprisingly high $MAE(0.54)$ and nearly negative $R2score(-0.002)$. This disparity may be explained by the TCN's shallow design used to accommodate the Coral Dev Board's resource constraints, data sparsity, and its sensitivity to hyperparameter adjustment. Even though it uses less energy (72.1%), demonstrating its efficiency advantage, the inconsistent accuracy metrics imply that more optimization of the model configuration is necessary to fully realize its potential. These findings highlight the significance of maintaining a balance between computing efficiency and forecast accuracy from the standpoint of deployment. Although TCN has great potential for time-series modeling, its edge implementation necessitates more cautious adjustments, including hybrid techniques, deeper structures, or

improved dilation factors. The best option for real-time RUL prediction on Coral hardware, however, is XGBoost, which showed the most dependable performance within the specified limitations.

5 Conclusion

The present work looked at predicting the RUL of lithium-ion batteries used in electric vehicles using advanced machine learning techniques including RF, XGBoost, and TCN. By putting these models into practice on the Google Coral Dev Device, we demonstrated the feasibility and value of directly implementing predictive maintenance techniques on edge computing hardware.

Based on our thorough analysis, XGBoost is particularly advantageous for real-time battery health monitoring in resource-constrained contexts as it strikes the best possible balance among prediction precision and processing efficiency. In short-term forecasts, TCN performed well; nevertheless, their instability indicated that more model optimization and tuning was necessary to properly utilize their temporal modeling capabilities in edge contexts. Although it required more energy, RF provided strong baseline performance with benefits for interpretability.
Future research should investigate more sophisticated hybrid decomposition and feature extraction methods to improve TCNs' temporal modeling capabilities, especially while computing at the edge. Furthermore, examining deeper and more adaptable neural network topologies in conjunction with transfer learning may enhance generalization and resilience across various battery types and operation environments. Real-time, low-power battery health management may be enabled on the hardware side by further lowering latency and energy consumption through model quantization and pruning optimization of the Coral Dev Board's deployment pipeline.

Acknowledgements

This work was carried out within the framework of the TEAMING project, which aims to accelerate the global adoption of sustainable electric powertrains through innovative predictive maintenance solutions. For more information about the project, visit https://www.sustainablepowertrains.com/. The authors thank all project partners and the European Commission for their support through the Marie Sklodowska-Curie Actions Staff Exchange Programme.

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2015, November). TensorFlow: Large-scale machine learning on heterogeneous systems.

[2] Ali, Z. A., Abduljabbar, Z. H., Tahir, H. A., Sallow, A. B., & Almufti, S. M. (2023). eXtreme gradient boosting algorithm with machine learning: A review. Academic Journal of Nawroz University, 12(2), 320-334.

[3] Amir, S., Gulzar, M., Tarar, M. O., Naqvi, I. H., Zaffar, N. A., & Pecht, M. G. (2022). Dynamic equivalent circuit model to estimate state-of-health of lithium-ion batteries. IEEE Access, 10, 18279-18288.

[4] Biau, G., & Scornet, E. (2016). A random forest guided tour. Test, 25(2), 197-227.

[5] Bent jac, C., Cs rg?, A., & Mart nez-Mu oz, G. (2021). A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 54(3), 1937-1967.

[6] Chang, Z., Tang, H., Zhang, Z., Zhang, X., Li, L., & Yu, Y. (2024). Prediction of remaining useful life and recycling node of lithium-ion batteries based on a hybrid method of LSTM and LightGBM. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, 46(1), 1-13.

[7] Coral. (2022). Coral Dev Board. Available online: https://coral.ai/products/dev-board/ (accessed on 10 February 2025).

[8] Deng, Z., Xu, L., Liu, H., Hu, X., Duan, Z., & Xu, Y. (2023). Prognostics of battery capacity based on charging data and data-driven methods for on-road vehicles. Applied Energy, 339, 120954.

[9] Feng, X., Weng, C., He, X., Han, X., Lu, L., Ren, D., & Ouyang, M. (2019). Online state-of-health estimation for Li-ion battery using partial charging segment based on support vector machine. IEEE Transactions on Vehicular Technology, 68(9), 8583-8592.

[10] Ge, M. F., Liu, Y., Jiang, X., & Liu, J. (2021). A review on state of health estimations and remaining useful life prognostics of lithium-ion batteries. Measurement, 174, 109057.

[11] Hafizhahullah, H., Yuliani, A. R., Pardede, H., Ramdan, A., Zilvan, V., Krisnandi, D., & Kadar, J. (2022, November). A hybrid CNN-LSTM for battery remaining useful life prediction with charging profiles data. In Proceedings of the 2022 International Conference on Computer, Control, Informatics and Its Applications (pp. 106-110).

[12] Hu, X., Feng, F., Liu, K., Zhang, L., Xie, J., & Liu, B. (2019). State estimation for advanced battery management: Key challenges and future trends. Renewable and Sustainable Energy Reviews, 114, 109334.

[13] Huang, H., Bian, C., Wu, M., An, D., & Yang, S. (2024). A novel integrated SOC?SOH estimation framework for whole-life-cycle lithium-ion batteries. Energy, 288, 129801.

[14] Jafari, S., & Byun, Y. C. (2022). Xgboost-based remaining useful life estimation model with extended Kalman particle filter for lithium-ion batteries. Sensors, 22(23), 9522.

[15] Kim, J., Chun, H., Kim, M., Han, S., Lee, J. W., & Lee, T. K. (2021). Effective and practical parameters of electrochemical Li-ion battery models for degradation diagnosis. Journal of Energy Storage, 42, 103077.

[16] Lai, X., Wang, S., Ma, S., Xie, J., & Zheng, Y. (2020). Parameter sensitivity analysis and simplification of equivalent circuit model for the state of charge of lithium-ion batteries. Electrochimica Acta, 330, 135239.

[17] Lyu, C., Song, Y., Zheng, J., Luo, W., Hinds, G., Li, J., & Wang, L. (2019). In situ monitoring of lithium-ion battery degradation using an electrochemical model. Applied Energy, 250, 685-696.

[18] Marwa, B., Berriri, H., & Mimouni, M. F. (2025, February). Comparative Analysis of LSTM and Bayesian Neural Networks for Lithium-Ion Battery Capacity Degradation Prediction. In 2025 15th International Renewable Energy Congress (IREC) (pp. 1-6). IEEE.

[19] Meng, H., & Li, Y. F. (2019). A review on prognostics and health management (PHM) methods of lithium-ion batteries. Renewable and Sustainable Energy Reviews, 116, 109405.

[20] Nuhic, A., Terzimehic, T., Soczka-Guth, T., Buchholz, M., & Dietmayer, K. (2013). Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. Journal of Power Sources, 239, 680-688.

[21] Nvidia. (2022). Nvidia Jetson Nano Developer Kit. Available online: https://developer.nvidia.com/embedded/jetson-nano-developerkit (accessed on 1 February 2025).

[22] Raspberry Pi. (2021). Raspberry Pi Products. Available online: https://www.raspberrypi.com/products/ (accessed on 20 January 2025).

[23] Saw, L. H., Ye, Y., & Tay, A. A. (2016). Integration issues of lithium-ion battery into electric vehicles battery pack. Journal of Cleaner Production, 113, 1032-1045.

[24] Srihith, D., Lakshmi, P. V., Donald, A. D., Srinivas, T. A. S., & Thippanna, G. (2023). A Forest of Possibilities: Decision Trees and Beyond. Journal of Advancement in Parallel Computing, 6(3), 29-37.

[25] Sui, X., He, S., Vilsen, S. B., Meng, J., Teodorescu, R., & Stroe, D. I. (2021). A review of non-probabilistic machine learning-based state of health estimation techniques for Lithium-ion battery. Applied Energy, 300, 117346.

[26] Tang, X., Zou, C., Yao, K., Lu, J., Xia, Y., & Gao, F. (2019). Aging trajectory prediction for lithium-ion batteries via model migration and Bayesian Monte Carlo method. Applied Energy, 254, 113591.

[27] Wang, X., Hu, B., Su, X., Xu, L., & Zhu, D. (2024). State of health estimation for lithium-ion batteries using random forest and gated recurrent unit. Journal of Energy Storage, 76, 109796.

[28] Weiss, M., Ruess, R., Kasnatscheew, J., Levartovsky, Y., Levy, N. R., Minnmann, P., ... & Janek, J. (2021). Fast charging of lithium?ion batteries: a review of materials aspects. Advanced Energy Materials, 11(33), 2101126.

[29] Wu, J., Zhang, C., & Chen, Z. (2016). An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks. Applied Energy, 173, 134-140.

[30] Xie, Q., Liu, R., Huang, J., & Su, J. (2023). Residual life prediction of lithium-ion batteries based on data preprocessing and a priori knowledge-assisted CNN-LSTM. Energy, 281, 128232.

[31] Xiong, R., Li, L., & Tian, J. (2018). Towards a smarter battery management system: A critical review on battery state of health monitoring methods. Journal of Power Sources, 405, 18-29.

[32] Yao, J., Neupert, S., & Kowal, J. (2024). Cross-Stitch Networks for Joint State of Charge and State of Health Online Estimation of Lithium-Ion Batteries. Batteries, 10(6), 171.

[33] Yang, N., Hofmann, H., Sun, J., & Song, Z. (2023). Remaining useful life prediction of lithium-ion batteries with limited degradation history using random forest. IEEE Transactions on Transportation Electrification, 10(3), 5049-5060.

[34] Yuliani, A. R., Ramdan, A., Zilvan, V., Supianto, A. A., Krisnandi, D., Yuwana, R. S., ... & Pardede, H. (2021, October). Remaining useful life prediction of lithium-ion battery based on LSTM and GRU. In Proceedings of the 2021 International Conference on Computer, Control, Informatics and Its Applications (pp. 21-25).

[35] Zraibi, B., Mansouri, M., & Loukili, S. E. (2025). RUL prediction for lithium-ion batteries using improved-CGD hybrid model. International Journal of Dynamics and Control, 13(6), 237.