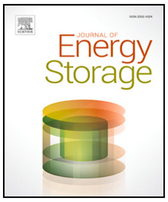
Contents lists available at ScienceDirect

Journal of Energy Storage

journal homepage: www.elsevier.com/locate/est



Research papers

Early prediction of battery remaining useful life using CNN-XGBoost model and Coati optimization algorithm

Vahid Safavi [a,*], Arash Mohammadi Vaniar [b], Najmeh Bazmohammadi [a], Juan C. Vasquez [a], Ozan Keysan [b], Josep M. Guerrero [a,c,d]

[a] *Center for Research on Microgrids (CROM), AAU Energy, Aalborg University, 9220 Aalborg East, Denmark*
[b] *Department of Electrical and Electronics Engineering, Middle East Technical University, 06800 Ankara, Turkey*
[c] *Center for Research on Microgrids (CROM), Department of Electronic Engineering, Technical University of Catalonia, 08034 Barcelona, Spain*
[d] *ICREA, Pg. Lluís Companys 23, 08010 Barcelona, Spain*

ARTICLE INFO

Keywords:
Lithium-ion batteries
Early remaining useful life prediction
Machine learning
XGBoost
CNN
Coati Optimization

ABSTRACT

Lithium-ion (Li-ion) batteries are essential for modern power systems but suffer from performance degradation over time. Accurate prediction of the remaining useful life (RUL) of these batteries is critical for ensuring the reliability and efficient operation of the power grid. On this basis, this paper presents a novel Coati-integrated Convolutional Neural Network (CNN)-XGBoost approach for the early RUL prediction of Li-ion batteries. This method incorporates CNN architecture to automatically extract features from the discharge capacity data of the battery via image processing techniques. The extracted features from the CNN model are concatenated with another set of features extracted from the first 100 cycles of measured battery data based on the charging policy information of the battery. This combined set of features is then fed into an XGBoost model to make the early RUL prediction. Additionally, the Coati Optimization Method (COM) is utilized for CNN hyperparameter tuning, to improve the performance of the proposed RUL prediction method. Numerical results reveal the effectiveness of the proposed approach in predicting the RUL of Li-ion batteries, where values of 106 cycles and 7.5% have been obtained for the RMSE and MAPE, respectively.

1. Introduction

Predicting the remaining useful life (RUL) of lithium-ion (Li-ion) batteries is essential for ensuring a reliable and efficient operation of power systems, given the influence of environmental conditions, cycling, and degradation. This degradation can lead to reduced capacity, increased internal resistance, and an increased risk of failure, potentially resulting in power outages, safety hazards, and economic losses [1]. Accurately predicting the RUL of Li-ion batteries allows power system operators to take proactive measures to prevent catastrophic failures, optimize maintenance schedules, and improve overall system reliability [2]. By utilizing RUL predictions, utilities can implement predictive maintenance strategies, such as scheduling battery replacements before they reach their end of life, reducing the risk of unexpected outages, and improving grid stability [3].

Predicting Li-ion battery lifetime with early-cycle data offers substantial advancements in battery production, utilization, and optimization. Manufacturers can expedite cell development, validate novel manufacturing processes, and categorize new cells based on their anticipated lifespan. Further, end users can also gauge the lifespan of their batteries. However, accurately predicting battery lifetime is challenging due to nonlinear degradation and variability, despite controlled operating conditions [4]. To address this challenge, several approaches such as physics-based, semi-empirical, and data-driven methods are developed. Therefore, an extensive literature review is conducted in the following subsection to identify the gap within this scope to be addressed by this paper.

1.1. Literature review

In this section, a literature survey is conducted for two types of approaches regarding the prediction of RUL of Li-ion batteries, namely Physic-Based (PB), and Data-Driven-Based (DDB) methods. PB models can be further classified into Electrochemical Model (EM), Semi-Empirical (SE), and Equivalent Circuit Model (ECM) [5]. On the other hand, DDB models utilize the battery performance and associated operating condition data to develop machine learning (ML)-based algorithms for RUL prediction purposes. Regarding PB models, the Persudo Two-Dimension (PTD) model is the most widely utilized EM to simulate

* Corresponding author.
E-mail address: vsa@et.aau.dk (V. Safavi).

https://doi.org/10.1016/j.est.2024.113176
Received 6 March 2024; Received in revised form 25 June 2024; Accepted 27 July 2024
Available online 3 August 2024

the behavior of the entire battery, including all the critical components that comprise Li-ion batteries [6,7]. A parameterized approach is proposed for PTD model by integrating the normalization, grouping, and sensitivity analysis to present the impacts of various parameters on the model accuracy, where current and voltage data are found to be effective features for more accurate prediction in [8]. Despite EM which require detailed knowledge of chemical reactions, ECM approaches such as Thevenin model [9], Rint model [10], and second-order electrical equivalent model [11] consider electrical components to model the battery behavior. In [12], a real-time framework is proposed to estimate the negative electrode potential and state-of-charge (SOC) of Li-ion batteries by considering a half-cell ECM on a 21 700 Li-ion cell. Furthermore, SE models are developed to derive a mathematical relationship between the input parameters and resultant predictions. This approach is divided into three categories: calendar life modeling [13], performance modeling [14], and cycle aging modeling [15].

The degradation performance of a battery is widely investigated using DDB models among scholars for the prediction of state-of-health (SOH) [16], end-of-life (EOL) [17], SOC [18], and RUL [19]. Accurate prediction of the battery RUL and EOL at different operating conditions is critical for the battery management system (BMS) to guarantee safe and efficient operation. This prediction assists in ensuring optimal battery utilization, maximizing lifespan, and preventing unexpected failures. In [20], a DDB framework is proposed by utilizing an automated feature selection to generate customized inputs for a Gaussian Process Regression (GPR) model for the estimation of battery health fluctuations, which leads to the proper prediction of EOL. The feature selection procedure in this methodology demonstrates flexibility in response to a wide range of inputs, prioritizing factors with significant impacts on battery degradation. In [21], a hybrid approach that includes variational mode decomposition, multi-kernel support vector regression, and sparrow search algorithm is proposed to enhance SOH estimation by utilizing the NASA dataset, but the proposed methodology is incompatible with RUL prediction. In [22], a combined detection and prediction model is proposed by employing unsupervised learning and extracting physics-informed features from an equivalent circuit model of a battery, which is tested on 65 batteries. The proposed model achieves over 90% accuracy in degradation stage detection and an RMSE value of 53.56% for life prediction performance. In [23], a moving window-based method is presented for in-situ battery life prediction and classification using ML techniques. By extracting features from partial charging data and employing GPR and SVM, this approach achieves EOL predictions with RMSE and MAPE values of 100 cycles and 10%, respectively.

In [24], a neural network (NN) is developed to predict the EOL of Li-ion batteries under various testing conditions such as varying temperature, charging/discharging current, and cut-off voltages, which highlights the potential of ML models in capturing the complex hidden features for predicting the cycle life. In [25], a residual-based convolutional neural network (CNN) model is presented for the RUL prediction, in which the residual network extracts the feature information of varying depths considering the features of the sparse dataset in a cloud computing environment. Results of the proposed methodology demonstrate a high degree of accuracy and reliability in predicting the RUL. In [26], a real-time framework is investigated by utilizing the classification and regression attributes of Support Vector Machine (SVM) for the RUL prediction of Li-ion batteries with various characteristics derived from voltage and temperature profiles. In [27], an improved extreme learning machine is developed for the improvement of RUL prediction of Li-ion batteries by using a feature extraction technique for Health Indicator (HI) that determines the deterioration of batteries, in which the Pearson correlation coefficient between each HI and capacity is extracted and tested on three different datasets.

In recent research, predicting the RUL of batteries has seen advancements, notably through the usage of novel hyperparameter tuning for ML and feature extraction methods from raw sensor data. Some scholars have employed optimization techniques to obtain optimal weights and thresholds for the parameter initialization of the ML models. In [28], particle swarm optimization (PSO) technique is used in a NN with backpropagation to predict the RUL and SOH of Li-ion batteries using two datasets, which considers HIs as the input features through the process of battery charging. Hyperparameter tuning is applied to optimize the number of epochs, dropout ratio, batch size, and number of hidden layers. Results demonstrate the effectiveness of the proposed model in terms of root-mean-squared error (RMSE) and mean absolute error (MAE) of SOH and RUL predictions. In [29], the Fruit-Fly Optimization (FFO) method is proposed with Fractional Brownian Motion (FBM) to predict the RUL of Li-ion batteries. Firstly, the FBM degradation model is presented to calculate the Hurst parameter, then, maximal likelihood estimation is utilized along with the FFO to optimize the Hurst parameter to perform the RUL prediction considering the capacity degradation data of the battery. In [30], a comprehensive strategy is proposed for battery RUL prediction, which incorporates an information entropy-based technique and an enhanced PSO algorithm for determining optimal degradation parameter values. Additionally, a Moving Average Filter (MAF) is employed to address noise and capacity degradation issues in the experimental data. The effectiveness of the proposed methodology is demonstrated using NASA and Maryland University datasets. The findings indicate that the algorithm achieves superior prediction accuracy while requiring less training data compared to DDB alternatives. In [31], authors present a DDB model that combines PSO method and feature enhancement using box-cox transformation to strengthen the correlation between features and the aging status of the battery to perform RUL prediction. The process of optimizing model parameters is carried out via PSO. Using actual Li-ion battery degradation data, the efficacy of this method is validated, and experimental results demonstrate the effectiveness of this method. In [32], authors employ the NASA dataset to develop a model of battery degradation by combining Support Vector Regression (SVR) with the Artificial Bee Colony (ABC) algorithm to predict the RUL of Li-ion batteries, in which ABC is utilized to optimize the SVR kernel parameters. The results indicate that the ABC algorithm outperforms the PSO algorithm in terms of parameter optimization.

Some researchers have applied a CNN model to enhance the performance of feature extraction methods. This approach has gained attraction due to the ability of CNNs to learn data patterns autonomously. In [33], a CNN-extreme gradient boosting (XGBoost) model is proposed for battery RUL prediction utilizing feature extension and time window processing, in which the effects of diverse operating conditions are considered. In [34], a CNN-light gradient-boosting machine (LightGBM) model is developed after preprocessing the raw data and extracting features using a CNN model. Then, the CNNs output is utilized as the input for the LightGBM model when generating RUL predictions using the C-MAPSS dataset. The findings show a substantial decrease in prediction error when CNN and LightGBM are combined, in comparison to the case in which CNN and LightGBM models are utilized independently with the same parameter configurations.

In [35], a dataset comprising 124 Li-ion cells, with cycle lifetimes spanning from 150 to 2300 cycles, is employed for early cycle life RUL prediction based on discharge voltage profiles obtained from early cycles that do not exhibit capacity degradation. Authors demonstrate that the most efficient models attain a test error of 9.1% by employing data from the initial 100 cycles to predict the cycle life, and when classifying cycle life into two categories using data from the initial five cycles, a test error of 4.9% is attained. In [36], RUL prediction of Li-ion batteries is investigated using multiple ML models, in which XGBoost reported as a top-performer among other ML models in terms of RMSE and mean absolute percentage error (MAPE). In [37], an early prediction approach combining two phases of NN integration and GPR is proposed. In the first phase, a NN model is trained with features of 124 Li-ion batteries to predict their initial cycle lifetime. Subsequently, GPR predicts the early RUL for each test set in the second phase. Evaluation of four batteries under various operational conditions shows that the RMSE for all capacity predictions is less than 1.2%. In [38], early cycle life RUL prediction of Li-ion batteries is proposed using deep learning with a parallel NN with two dimensions. This methodology commences with the implementation of a five-step preprocessing approach to generate input data and extract correlated features. The methodology ultimately results in the determination of the present cycle life and a prediction regarding the RUL of the battery. In the test set, which consists of information from 31 cells with 25 unique cycle profiles, the MAPE for predicting the RUL and early lifetime is reported at 1.47% and 2.85%, respectively. In [39], three regression models including Linear Regressor (LR), Bagging Regressor (BR), and Random Forest Regressor (RFR), are presented to predict the early lifespan of Li-ion batteries in electric vehicles. The RFR outperforms the other two models, achieving the lowest Mean Square Error (MSE) and RMSE values. In [40], predicting the cycle lives of Li-ion batteries via randomized trees using data from their initial cycles is developed for 121 commercial Li-ion batteries considering the feature extraction. In [41], a few-shot learning-based methodology is proposed that requires six cycles of charging data to predict the early lifetime of batteries. In this methodology, batteries are classified into long-life and short-life categories by autoencoder. Upon the classification outcomes, lifetime models are chosen to arrive at the ultimate prediction. Using an accelerated aging dataset comprised of 124 batteries, the framework's effectiveness in circumventing data limitations and enhancing the accuracy of early lifetime prediction is validated.

1.2. Novelty and contribution

According to the literature review, only two relevant studies have investigated the RUL prediction for Li-ion batteries using the CNN-XGBoost model. Therefore, a novel data-driven approach is proposed in this paper, which entails the development of a CNN model to extract features from the battery discharge capacity data via image processing techniques. The extracted features are concatenated with another set of features derived from measured battery data and a new feature proposed in this paper that is calculated from the charging policy of the battery. Additionally, we incorporate the Coati optimization method (COM) into the CNN model to derive optimal values for critical parameters, such as the number of filters (K_3), kernel size (K_4), and number of units in the dense layer (K_5), through an iterative process. To evaluate the performance of the developed models, two case studies are presented. The initial case investigates the XGBoost model in depth by training and validating it with distinct feature sets. In the second scenario, the CNN-XGBoost model is constructed with and without the COM integration. The numerical results highlight the efficacy of the proposed methodology in predicting the RUL of Li-ion batteries. The significant contributions made by this paper are as follows:

- Developing a CNN model for feature extraction via image processing.
- Integrating the COM within the CNN model to optimize critical parameters of the number of filters, kernel size, and number of units in the dense layer to improve the prediction accuracy.
- Proposing a new feature calculated from the charging policy of the battery.
- Proposing a new set of features by concatenating extracted features from the CNN model, the proposed new feature, and another set of features extracted from measured data to feed with XGBoost to enhance the effectiveness of the CNN-XGBoost model for battery RUL prediction.
- Implementing various case studies to investigate the effectiveness and limitations of the proposed methodology.

Table 1
Degradation data of Li-ion batteries with a rated capacity of 1.1 Ah and constant discharge current of 4C.

Battery group	Rated/End voltage (V)	Charge/Discharge rest time (s)	Environmental temperature (°C)
Train data	3.7/2.7	60/60	30
Test data-1	3.7/2.7	300/300	30
Test data-2	3.7/2.7	5/5	30 to 36

1.3. Paper organization

The rest of the paper is organized as follows: Section 2 presents the data processing method including data description, data preprocessing, data cleaning, normalization, feature extraction, and splitting data. Section 3 elaborates on the proposed methodology with a focus on the explanation of developed CNN, XGBoost, and COM. Subsequently, Section 4 presents the case studies and simulation results. Following this, Section 5 concludes the paper.

2. Data processing

This section contains a comprehensive and detailed explanation of the dataset utilized in our research, along with an in-depth examination of the preprocessing methods implemented to guarantee the data's integrity, dependability, and suitability for the proposed methodology. Furthermore, a comprehensive explanation of the preprocessing procedures performed is provided in the following subsections, including data description, and data preprocessing which includes data cleaning, normalization, feature extraction, feature selection, and splitting data.

2.1. Data description

In this paper, a dataset of 124 commercial lithium iron phosphate/graphite cells [35] from the A123 system (model APR18650M1A, 1.1 Ah nominal capacity) is considered. The batteries in the dataset are subjected to diverse fast-charging policies until they reach the EOL, defined as 80% of their initial capacity. This dataset is divided into three segments: Train data, Test data-1, and Test data-2, comprising 41, 43, and 40 lithium-ion batteries, respectively. The battery capacity degradation and distribution of battery cycle life are presented in Fig. 1.

This battery dataset was obtained in a controlled environment at 30 °C for Train data, and Test data-1, and ranged from 30 °C to 36 °C for Test data-2, with fast-charging conditions ranging between 3 C and 8 C as detailed in Table 1. It is structured into three phases with different charging rates of C1, C2, and a constant C3 with a value of 1 C. It should be mentioned that to discharge the battery, a fixed discharging rate of 4 C is applied until the cut-off voltage is reached as shown in Fig. 2. The dataset encompasses a wide range of charging policies, including varying charging times from 9 to 15 min and cycle life spanning from 150 to 2300 cycles. The parameters of each cycle such as temperature, current, voltage, and charge/discharge capacity, were meticulously recorded, enabling detailed insights into battery degradation characteristics.

2.2. Data preprocessing

Within the domain of data analytics, the initial raw measurement data is often contaminated with noise, which poses a significant obstacle to the application of ML models. Inputting such noisy data directly into the models' input layers may hinder the learning process and introduce errors due to the presence of bad data [42]. As a result, a critical component of our approach is data preprocessing, which is implemented to guarantee that the data is accurate, genuine, and structured in a way that facilitates optimal utilization by the ML models. The preprocessing procedure consists of the following five important

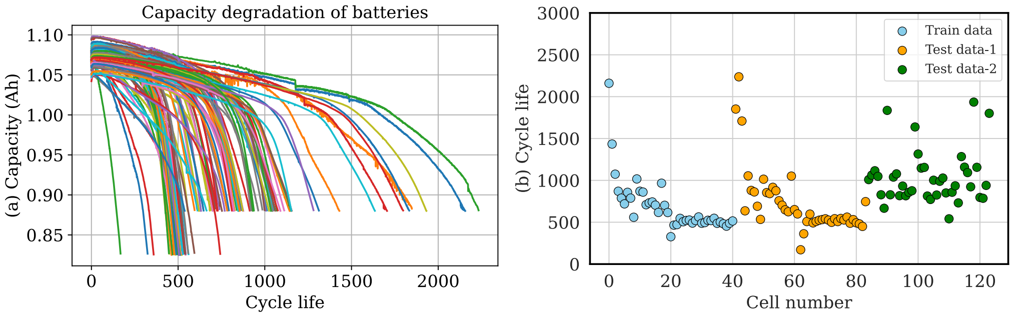


Fig. 1. Battery capacity fades for 124 cells in the dataset (a), and distribution of batteries cycle life (b).

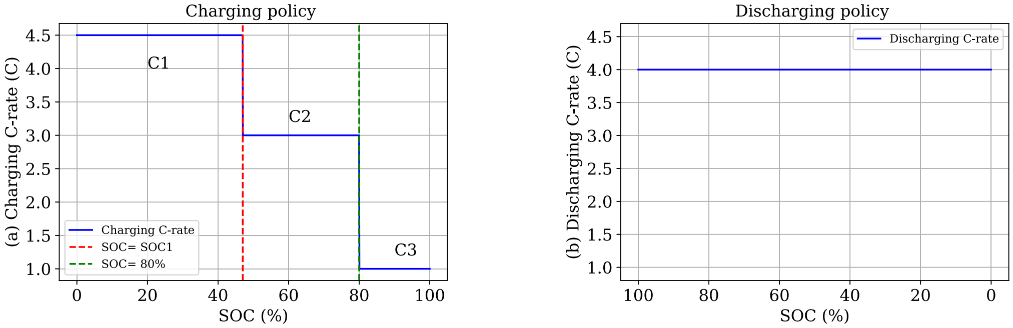


Fig. 2. Description of the charging policy “C1, SOC1, C2” (a) and constant discharging policy (b).

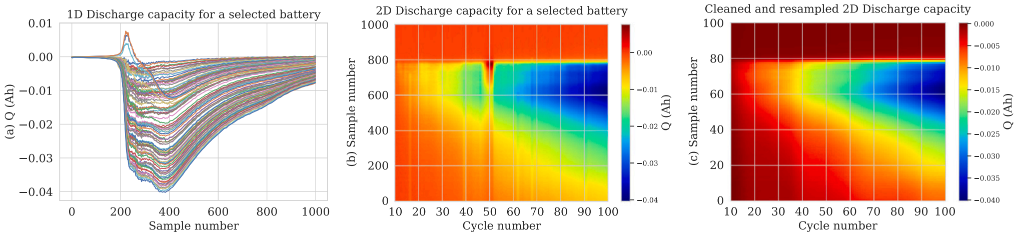


Fig. 3. Discharge capacity fades depicted in 1D (a) and 2D (b) for a selected battery, followed by 2D discharge capacity fades after cleaning and resampling data (c).

phases: building the 2D discharge capacity dataset, cleaning and resampling data, normalization, feature extraction, feature selection, and splitting. Every single one of these stages is critical in improving the unprocessed data, which in turn prepares it for the subsequent stages of our analytical pipeline. In doing so, we ultimately strengthen and increase the reliability of our predictive models.

2.2.1. Building the 2D discharge capacity dataset

In order to effectively utilize the CNN model and derive features from the discharging capacity (Q) data, a structured 2D dataset is required. The transformation of Q-V into a 2D dataset is depicted in Fig. 3(a, and b). Here, each data point is represented by a pair (X, y), where ‘X’ denotes the cycle number and ‘y’ represents the voltage sample number. The corresponding values (X, y) signify the discharging capacity in Ampere-hours (Ah).

2.2.2. Cleaning and resampling data

To enhance the accuracy of ML models, cleaning data is a crucial initial step in the data preprocessing pipeline. This process aims to refine unprocessed measurement data by rectifying noise, removing errors and inconsistencies, and eliminating outliers [43]. By focusing on specific issues such as overshooting or undershooting data in battery discharge capacity, data cleaning ensures that the data utilized by ML models is reliable and free from abnormalities, leading to improved model performance and more accurate predictions. To achieve these purposes, a cleaning data process is proposed, comprising the following steps:

- Thresholding: Modify input values greater than a pre-defined threshold value.
- Forward Difference Filtering: Calculate intensity differences between adjacent values in the forward direction and adjust input values to ensure differences are either zero or negative.
- Midpoint Averaging: Smooth the values by replacing each value with the average of its neighboring values. In all rows, from right to left, each value is replaced by the average of its immediate right and left neighbors. This process is repeated iteratively to ensure consistent smoothing across the entire dataset.

Midpoint Averaging technique can effectively reduce noise and improve data consistency in ML datasets. This method enhances feature extraction and model robustness by highlighting true patterns and minimizing the impact of anomalies. Consequently, it leads to more accurate and reliable predictive models.

By adding more data, the DL model has more parameters to manage, which makes the learning process longer and more complicated [44]. To tackle this problem, a resampling technique is used to average every ten discharge capacity data points following the cleaning process. Fig. 3(c) illustrates the 2D discharge capacity for a selected battery, after cleaning, noise reduction, and resampling. This approach is essential to manage the computational load and enhance model efficiency,

Table 2
Extracted features and their descriptions.

Feature name	Details of extracted features
F1	Variance = $\log\left(\left\lvert \frac{1}{p-1}\sum_{i=1}^{p}(\Delta Q(V)-\Delta\bar{Q}(V))^2\right\rvert\right)$
F2	$\log(\lvert \min(\Delta Q(V))\rvert)$
F3	Gradient of discharge curves over cycles 2 to 100.
F4	Intercept of a linear fit to capacity fade curves from cycles 2 to 100
F5	Discharge capacity at cycle 2
F6	Average charge time over first 5 cycles
F7	Minimum internal resistance
F8	Difference in internal resistance between cycle 100 and cycle 2
F9	C-rate in the first charging step (C1)
F10	C-rate in the second charging step (C2)
F11	Average weighted C-rate in each cycle

especially considering that the primary data size of 1000 × 90 with values between −0.04 to 0 significantly increases computational demands without improving the CNN model's prediction accuracy. Therefore, resampling is employed to reduce the image dimensions and create a balance between data resolution and computational efficiency.

2.2.3. Normalization

Normalization is a crucial step in data preprocessing to ensure that different features of the data are on the same scale and have a consistent range. This process improves the performance of ML algorithms by facilitating learning patterns and relationships in the data [45]. There are various normalization techniques available, namely min–max normalization, Z-score, and decimal scaling normalization, from which the min–max normalization is utilized in this paper. Min–Max normalization is a simple and widely used normalization technique that scales the data linearly to a specific range, typically [0, 1] or [−1, 1]. The mathematical formulation of min–max normalization is provided by Eq. (1).

$$x_{norm} = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

where x is the original data value before normalization, x_{norm} is the normalized data value, $min(x)$ and $max(x)$ are the minimum and maximum values of x in the dataset, respectively.

The outputs of normalization, serve as the input for the CNN model to calculate the features from each 2D data.

2.2.4. Feature extraction

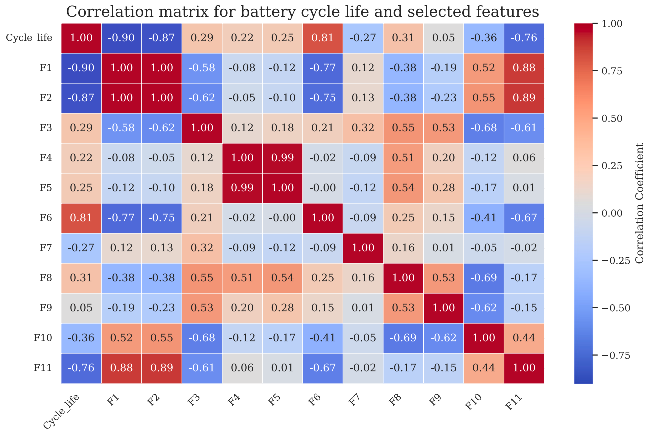
Feature extraction is an essential component of the data preprocessing pipeline, which aims to enhance the efficacy of ML models by identifying and extracting pertinent information from the dataset. The procedure entails the transformation of data into a more compact and informative format, thereby decreasing the dimension while maintaining critical patterns and attributes [46]. The extracted features in this paper are presented in Table 2. Features $F1$ to $F8$ are obtained from [35,37]. The decline in the Li-ion battery capacity is mirrored in $F1$, $F2$, $F3$, $F4$, and $F5$ is calculated using (2) and (3) as follows:

$$\Delta Q(V) = \frac{Q_{100}(V) - Q_{10}(V)}{p}, \tag{2}$$

$$\Delta\bar{Q}(V) = \frac{1}{p}\sum_{i=1}^{p}\Delta Q(V), \tag{3}$$

where $\Delta Q(V)$ represents the change in charge capacity at a given voltage, V. $Q_{100}(V)$ and $Q_{10}(V)$ denote the charge capacities at 100th and 10th cycles, respectively. $\Delta\bar{Q}(V)$ represents the average change in charge capacity at the given voltage V. The symbol p denotes the number of trials or repetitions.

Additionally, we propose features $F9$ to $F11$ that are extracted from the battery charging policies. $F9$ and $F10$ denote the C-rate during the initial and intermediate stages of battery charging, respectively. $F11$ is formulated as (4) to present the average weighted C-rate (AWCR) using C_1, C_2, C_3, and the battery SOC extracted from the battery's charging policy.



	Cycle_life	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
Cycle_life	1.00	-0.90	-0.87	0.29	0.22	0.25	0.81	-0.27	0.31	0.05	-0.36	-0.76
F1	-0.90	1.00	1.00	-0.58	-0.08	-0.12	-0.77	0.12	-0.38	-0.19	0.52	0.88
F2	-0.87	1.00	1.00	-0.62	-0.05	-0.10	-0.75	0.13	-0.38	-0.23	0.55	0.89
F3	0.29	-0.58	-0.62	1.00	0.12	0.18	0.21	0.32	0.55	0.53	-0.68	-0.61
F4	0.22	-0.08	-0.05	0.12	1.00	0.99	-0.02	-0.09	0.51	0.20	-0.12	0.06
F5	0.25	-0.12	-0.10	0.18	0.99	1.00	-0.00	-0.12	0.54	0.28	-0.17	0.01
F6	0.81	-0.77	-0.75	0.21	-0.02	-0.00	1.00	-0.09	0.25	0.15	-0.41	-0.67
F7	-0.27	0.12	0.13	0.32	-0.09	-0.12	-0.09	1.00	0.16	0.01	-0.05	-0.02
F8	0.31	-0.38	-0.38	0.55	0.51	0.54	0.25	0.16	1.00	0.53	-0.69	-0.17
F9	0.05	-0.19	-0.23	0.53	0.20	0.28	0.15	0.01	0.53	1.00	-0.62	-0.15
F10	-0.36	0.52	0.55	-0.68	-0.12	-0.17	-0.41	-0.05	-0.69	-0.62	1.00	0.44
F11	-0.76	0.88	0.89	-0.61	0.06	0.01	-0.67	-0.02	-0.17	-0.15	0.44	1.00

Fig. 4. Pearson correlation Heatmap for battery cycle life and selected features in train data.

$$AWCR = \frac{K_1 C_1 \mathrm{SOC} + K_2 C_2 (80 - \mathrm{SOC}) + C_3(100 - 80)}{100} \tag{4}$$

where C_1, C_2, and C_3 denote the C-rate during the initial, intermediate, and final stages of battery charging, respectively. The parameter SOC signifies the state of charge at which the transition from C_1 to C_2 occurs. Additionally, K_1 and K_2 represent coefficient factors derived through optimization techniques aimed at maximizing the correlation between the AWCR and cycle life of the battery.

2.2.5. Feature selection

Feature selection facilitates the choice of a subset of features that make it a substantial contribution to the predictive capability of ML models. To accomplish this, Pearson correlation [47] is utilized for selecting features. The Pearson correlation coefficient indicates how changes in one variable are correlated with changes in another by quantifying the linear relationship between the two variables, and its mathematical representation is provided by (5).

$$r = \frac{\mathrm{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{5}$$

where, r is the Pearson correlation coefficient, $\mathrm{cov}(X, Y)$ is the covariance of X and Y, σ_X is the standard deviation of X, and σ_Y is the standard deviation of Y.

The result of the Pearson correlation demonstrates the substantial inter-feature correlations and the correlation of features with battery cycle life, as presented in Fig. 4. The four features (F1, F2, F6, and F11) were chosen based on their strong correlation (exceeding 75%) with cycle life in the training data. These correlations are depicted in Fig. 5, showing this significant linear relationship. This high correlation suggests that these features are likely to be predictive of cycle life,

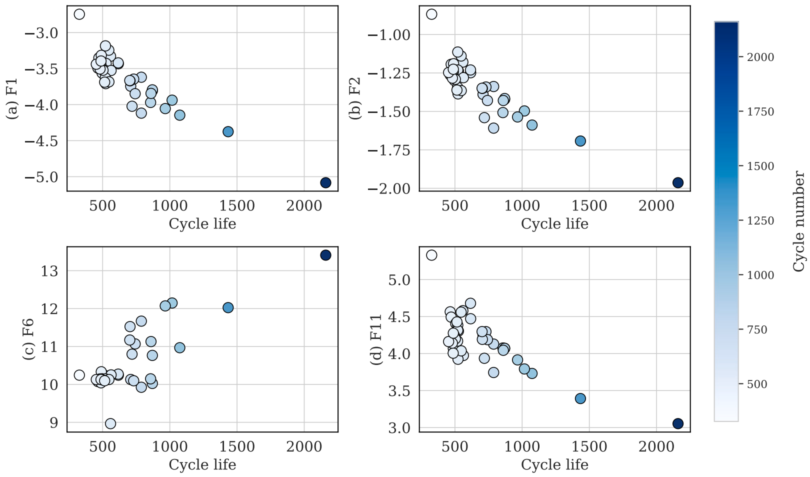


Fig. 5. Correlation of the selected features (F1 (a), F2 (b), F6 (c), and F11 (d)) and battery life cycle for train data.

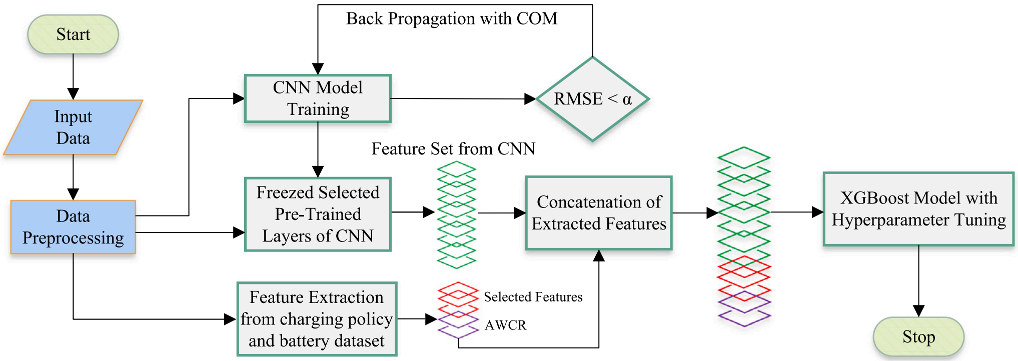


Fig. 6. Structure of the proposed Coati-CNN-XGBoost ML model.

justifying their selection. The analysis reveals that features F1 and F2 have correlation coefficients of −0.9 and −0.85, respectively, while the proposed feature F11, calculated with the optimized values of 0.55 and 0.45 for K_1 and K_2, respectively, demonstrates a correlation coefficient of −0.72 with cycle life for the training data. The objective of extracting these features is to optimize the performance of the proposed prediction model, decrease the dimension, and enhance the overall model interpretability. The chosen features will form the basis for the subsequent phases of model development and analysis.

2.2.6. Splitting data

The dataset, consisting of 124 battery data records, is divided into three groups, as shown in Fig. 1. The first group, Train data, containing 41 battery data records, is used for training and validation. Using a split ratio of 0.8 for the training/validation set ensures a fair distribution for model evaluation and comparison with other research papers. The second and third groups, with 43 and 41 battery data records respectively, are designated as test data-1 and test data-2. In this paper, the training/validation set is employed for model training and hyperparameter tuning, while the test data-1 is utilized to evaluate the performance of the trained model.

3. Methodology

In this section, The proposed methodology for predicting the RUL of Li-ion batteries is introduced. In this architecture, to address this challenge, we have developed a CNN is developed that adeptly extracts features from training data. This data is partitioned into 80% for training and 20% for validation. After training and validation, the CNN model is fine-tuned by discarding the last two layers and extracting features from the preceding layer, resulting in a set of features. The CNN-derived feature set is then concatenated with the selected features derived from the measured data and the new proposed feature extracted from the battery charging policies. This combined feature set is then fed into an XGBoost model for early RUL prediction. To improve the accuracy of the proposed methodology, the COM is used to fine-tune hyperparameters for the CNN model and the feature extraction process. COM aids in determining the optimal values of K_3, K_4, K_5 for the CNN model. In the subsequent subsections, detailed mathematical representations and explanations of the employed methods are provided, encompassing CNN, XGBoost, COM, hyperparameter tuning, and evaluation metrics. The architecture of the proposed methodology is represented in Fig. 6.

3.1. CNN

CNN models are a subset of deep learning models that are desirable for processing grid-like data, such as images. CNNs are frequently employed for object detection, image classification, and other visual data-intensive tasks. CNN is constructed primarily through the convolution operation to identify patterns in data. In order to execute the convolution operation, a filter is applied to the input data. After relocating the filter, which is a compact matrix of weights, the dot product between the input data and the filter is computed at each position. When the convolution operation is performed, a feature map is produced. Following this, a nonlinear activation function, such as the rectified linear unit (ReLU), is applied to the feature map. By introducing nonlinearity into the model via the activation function, it becomes capable of learning complex patterns. Typically, the activation function and convolution operation are iterated numerous times, resulting in

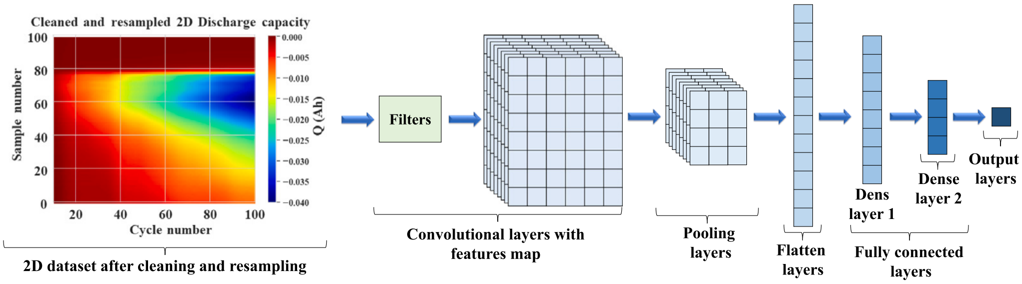


Fig. 7. Structure of CNN model.

filters that progress in complexity with each subsequent layer [48]. Typically, a fully connected layer serves as the concluding component of the CNN, performing the task of classifying the input data. Given an input data matrix X of size $m \times n$, and a filter matrix W of size $k \times l$, the convolution operation is defined by (6):

$$Y_{ij} = \sum_{p=0}^{k-1} \sum_{q=0}^{l-1} W_{pq} X_{i+p,j+q} \tag{6}$$

where Y is the output feature map, and i and j are the indices of the output element. The convolution operation can be thought of as a sliding window that is applied to the input data. The filter is placed at the top-left corner of the input data, and the dot product of the filter and the input data is calculated. The filter is then moved one pixel to the right, and the dot product is calculated again. This process is repeated until the filter has been applied to the entire input data. The feature map contains a representation of the features that have been extracted from the input data that can then be used to classify the input data.

Fig. 7 illustrates the preprocessed 2D dataset, after cleaning, resizing, and masking procedures, that are prepared for feeding the CNN model. In this figure, the outputs of the dense layer, calculating the features from each 2D data instance, are concatenated with four other selected features. This concatenated feature set serves as the input for the subsequent XGBoost model to predict the RUL of batteries.

3.2. XGboost model

XGBoost is a type of gradient-boosting machine that is designed for speed and accuracy, and is commonly used for regression and classification tasks. XGBoost works by building a series of decision trees. Each decision tree is built on a subset of the data, and the predictions from the individual trees are combined to make a final prediction. XGBoost uses a gradient-boosting algorithm to train the decision trees. Gradient boosting is an iterative algorithm that builds a model by repeatedly adding new decision trees to the model. Each new decision tree is built to correct the errors of the previous trees [49]. The objective function that XGBoost minimizes is defined in (7):

$$L(y, \hat{y}) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \Omega(\hat{f}) \tag{7}$$

where y is the true label, $\hat{y}$ is the predicted label, l is the loss function, and Ω is a regularization term. The regularization term Ω is used to control the complexity of the model is defined as follows:

$$\Omega(\hat{f}) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{m} w_j^2 \tag{8}$$

where $\hat{f}$ is the predicted function, T is the number of leaves in the tree, m is the number of features, γ is the regularization parameter, λ is the L2 regularization parameter, and w_j is the weight of the jth leaf.

3.3. Coati optimization method

The COM is a metaheuristic optimization algorithm and draws inspiration from the foraging behavior of coatis. Coatis are mammalian predators that search the ground for food with their extended snouts. They possess an acute sense of scent and are capable of locating sustenance concealed underground. As a population-based algorithm, the COM maintains a pool of candidate solutions. Every candidate solution signifies a prospective resolution to the optimization problem. The algorithm refines the candidate solutions iteratively through the implementation of a sequence of operators, which consist of [50]:

- **Exploration:** The exploration operator is used to generate new candidate solutions. This operator can be any type of random search algorithms, such as uniform or Gaussian random sampling.
- **Exploitation:** The exploitation operator is used to improve the existing candidate solutions. This operator can be any type of local search algorithms, such as hill climbing or simulated annealing.
- **Selection:** The selection operator is used to select the best candidate solutions from the population. This operator can be any type of selection algorithms, such as roulette wheel or tournament selection.

The COM terminates when a stopping criterion is met. The stopping criterion can be based on the number of iterations, the time limit, or the convergence of the algorithm. The solution update mechanism of the COM can be represented mathematically as follows:

$$X_{t+1} = X_t + \alpha(X_t - X_{best}) + \beta(X_t - X_{rand}) \tag{9}$$

where X_t is the current population, X_{t+1} is the next population, X_{best} is the best candidate solution found so far, X_{rand} is a randomly selected candidate solution, and α and β are parameters that control the exploration and exploitation rates. Due to the distinguishing features of COM resulting in enhanced robustness, reduced algorithm complexity, effective incorporation of information, and high convergence speed [50], it is used in this paper to optimize hyperparameters of the proposed ML model as will be explained in the following subsections.

3.4. Hyperparameter tuning

During the construction of an ML model, optimizing hyperparameters becomes an essential procedure to improve the efficiency of the model. The process of hyperparameter optimization entails determining the optimal values for the parameters that regulate the behavior of the model. The hyperparameters in question comprise various aspects, including but not limited to the learning rate, number of trees, estimators, maximum depth, minimal child weight, and regularization parameters. Various methodologies are available for optimizing hyperparameters, with random search, Bayesian optimization, and grid search frequently utilized techniques. These methodologies could present computational obstacles, particularly when confronted with vast search spaces. Our study utilized grid search, which assesses every conceivable combination of hyperparameters within predetermined ranges. The efficacy of XGBoost is improved by creating a hyperparameter-tuned variant, which is accomplished via Grid Search CV [51]. The hyperparameters of the XGBoost model that are optimized in this paper include the

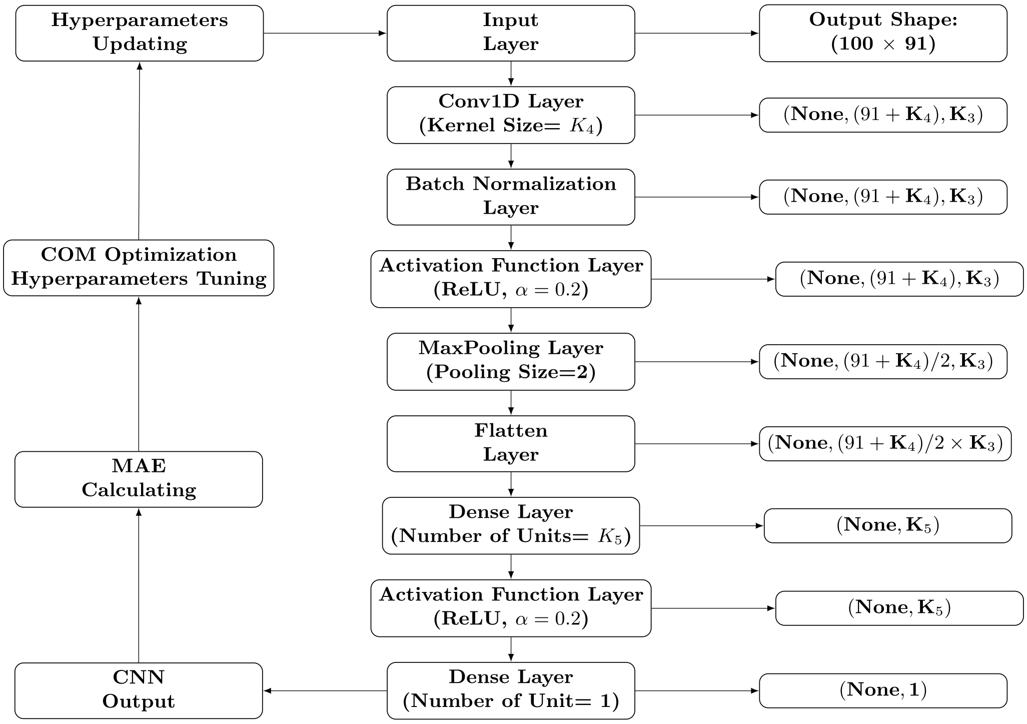

Hyperparameters Updating
Input Layer
Output Shape: (100 × 91)
Conv1D Layer (Kernel Size= K_4)
$(\mathbf{None}, (91+\mathbf{K}_4), \mathbf{K}_3)$
Batch Normalization Layer
$(\mathbf{None}, (91+\mathbf{K}_4), \mathbf{K}_3)$
COM Optimization Hyperparameters Tuning
Activation Function Layer (ReLU, $\alpha = 0.2$)
$(\mathbf{None}, (91+\mathbf{K}_4), \mathbf{K}_3)$
MaxPooling Layer (Pooling Size=2)
$(\mathbf{None}, (91+\mathbf{K}_4)/2, \mathbf{K}_3)$
Flatten Layer
$(\mathbf{None}, (91+\mathbf{K}_4)/2 \times \mathbf{K}_3)$
MAE Calculating
Dense Layer (Number of Units= K_5)
$(\mathbf{None}, \mathbf{K}_5)$
Activation Function Layer (ReLU, $\alpha = 0.2$)
$(\mathbf{None}, \mathbf{K}_5)$
CNN Output
Dense Layer (Number of Unit= 1)
$(\mathbf{None}, \mathbf{1})$

Fig. 8. Architecture of the Coati-CNN model with subsequent processing steps.

number of trees, maximum depth of each tree, learning rate, and the number of samples and features in each tree.

According to Fig. 8, hyperparameters of CNN are indicated as K_3, K_4, and K_5. In CNNs, K_3 plays a pivotal role in feature extraction from input images, as each filter, applied as a small matrix, generates distinct feature maps. Determining the appropriate number of filters in a convolutional layer significantly influences the network's ability to capture diverse aspects of input data. Alongside this, K_3 dictates the receptive field of neurons within a convolutional layer. Typically, common kernel sizes such as 3 or 5 are employed. Furthermore, in neural network architecture, K_4, which represents the number of neurons within a fully connected layer, establishes intricate connections between successive layers. In this paper, we propose a COM to determine the optimal values of hyperparameters K_3, K_4, and K_5 in the CNN model. The architecture of our proposed model, along with the subsequent processing steps, is illustrated in Fig. 8. To train and validate the COM-CNN model, the dataset is partitioned into 41 training samples, from which 33 samples are used for training and 8 samples for validation. Through the optimization process, the optimal values for K_3, K_4, and K_5 have been found to be 121, 5, and 10, respectively.

3.5. Evaluation metrics

The comparative analysis in this paper employs two key performance metrics—MAPE and RMSE—widely acknowledged for evaluating the precision of Li-ion battery RUL predictions. MAPE quantifies the average absolute percentage error between actual and predicted values. On the other hand, RMSE gauges the average squared difference between actual and predicted values [52]. Mathematical expressions representing RMSE and MAPE are provided in (10) and (11):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{10}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{11}$$

where y_i and $\hat{y_i}$ are the actual and predicted values for the RUL of the battery, respectively.

4. Case study and numerical results

In this section, two case studies are presented to evaluate the performance of the proposed methodology, as gauged by the evaluation metrics introduced in Section 3.5. Despite the prevalent use of data-driven ML models for predicting the RUL of Li-ion batteries, their performance falls short, even when incorporating multiple features during the training phase. To substantiate this assertion, two case studies are investigated, each addressing the deficiencies observed and unveiling a highly efficient solution. In the first case study, a hyperparameter-tuned XGBoost model is developed, exploring diverse combinations of features presented in Table 2. Accordingly, three different feature combinations (FCs) are utilized for training and testing the XGBoost model: FC1, encompassing features F1 to F10; FC2, spanning features F1 to F11; and FC3, comprising selected features F1, F2, F6, and F11. This exploration allows for the evaluation of the model's performance across various feature configurations to address the importance and significance of selected features and the new proposed AWCR feature. In the second case study, a CNN model is developed to extract the additional set of features through advanced image processing techniques. Subsequently, these extracted features are integrated with F1, F2, F6, and F11 as the input feature set for the XGBoost model. Moreover, a COM is embedded within the CNN, offering fine-tuned adjustments to hyperparameters such as K_3, K_4, and K_5. Consequently, the second case study involves a comprehensive assessment of the CNN-XGBoost model, both with and without the adaptive COM. The subsequent subsections provide the numerical findings derived from the proposed methodology across these two case studies, summarized as follows:

- **Case 1:** Early RUL prediction with XGBoost model with three different FCs.
- **Case 2:** Early RUL prediction with CNN-XGBoost model, with and without the COM for hyperparameter tuning and the selected set of features from Case 1.

Table 3
RMSE and MAPE values for RUL prediction of test data with different features combination.

Model	Features	RMSE (cycle)	MAPE (%)
XGBoost	FC1	116	9.1
	FC2	114	9.3
	FC3	111	9.0

4.1. Case 1

In this subsection, the results of the developed XGBoost model with three distinct sets of FCs are presented. The results are presented prediction via XGBoost are showcased in Fig. 9, showing the training and prediction outputs for each of the FCs mentioned above. Fig. 9(a) illustrates the alignment between the XGBoost training data and the actual data to validate the training performance of the proposed model. Additionally, Fig. 9(b, c, and d) provide the prediction results obtained, showcasing the model's adeptness in closely capturing the fluctuations observed in the actual data. Furthermore, the model shows resilience in handling outliers within the actual data.

Table 3 provides RMSE and MAPE values corresponding to each FC. Notably, for FC2, the RMSE and MAPE values stand at 114 cycles and 9.3%, respectively. These results underscore the potential for improvement in the overall performance of the XGBoost model with the incorporation of the proposed AWCR feature, and noticeable enhancements in both RMSE and MAPE metrics. Moreover, our investigation extends to the comparison between FC3 and FC2, where it becomes apparent that despite reducing the number of features to merely 4 in FC3, the predictive efficacy in terms of battery RUL remains comparable to FC2, which incorporates 11 features. This observation emphasizes the efficacy of the proposed feature selection strategy in FC3, paving the way for streamlined yet equally effective predictive modeling. Since the proposed methodology has been verified by the results of Case 1, we will proceed to the next case with the knowledge of using FC3 in the developed CNN-XGBoost model.

4.2. Case 2

In this subsection, three cases are compared: the CNN model, CNN-XGBoost model integrated with the COM and the CNN-XGBoost model without hyperparameter-tuning. The RUL predictions for these cases are illustrated in Fig. 10, revealing that the integration of CNN and COM significantly enhances the model's performance compared to both Case 1 and the CNN-XGBoost model without integration of COM. This improvement can be attributed to several factors. Firstly, leveraging a CNN to extract features from the discharge capacity data proves to be advantageous. Additionally, concatenating specific features such as F1, F2, F6, and AWCR with the features extracted by the CNN model enriches the representation of the data, enhancing the model's predictive capability. This feature augmentation strategy allows the model to consider a broader range of information relevant to the RUL prediction task. Moreover, the incorporation of COM for hyperparameter tuning of the CNN model further improves its performance, leading to more accurate predictions.

To quantify the performance improvement, RMSE and MAPE values are presented in Table 4, comparing the proposed CNN-XGBoost models with and without COM against available benchmarks. In [35], three models—the "Variance", "Discharge", and "Full" models—based on the number of selected features or grouped cells by cycle life are proposed. The results are reported in two formats: including and excluding an outlier sample that does not match other followed patterns. In [53], a Hybrid LSVR model achieves a MAPE value of 8.2%, but only for Test Data-1. In [54], a hybrid DL model is introduced, exploring various feature selection and ensembling methods to enhance battery life prediction accuracy, achieving a MAPE of 8.5%. In [55], a model with three convolution layers is developed to improve early RUL prediction, achieving a MAPE of 10%. Meanwhile, the proposed COM-CNN-XGBoost method achieves a MAPE of 7.5%, surpassing not only the ML models developed in this study but also other established benchmarks. By incorporating COM into the CNN-XGBoost model, feature selection and hyperparameter tuning are optimized, leading to an improvement in the performance of RUL prediction.

Table 4
RMSE and MAPE values for RUL prediction of test data-1 using the proposed method.

Benchmark	Model	RMSE (cycle)	MAPE (%)
Severson et al. [35]	Variance	138 (138)	14.7 (13.2)
	Discharge	91 (86)	13 (10.1)
	Full	118 (100)	14.1 (7.5)
Alipour et al. [53]	LSVR	(152)	(8.2)
Xu et al. [54]	Ensemble learning	(114)	(8.5)
Ma et al. [55]	CNN	(90)	(10)
This paper	CNN	279 (222)	17.6 (12.4)
	XGBoost	115 (111)	11.9 (9.0)
	CNN-XGBoost	117 (107)	12.9 (8.1)
	COM-CNN-XGBoost	113.6 (106)	**11.3 (7.5)**

The parenthetical values for Test data-1 represent the results after excluding a battery data outlier.

Table 5
Comparing MAPE values for RUL prediction of test data using different validation data split rates.

Model	Validation data		MAPE (%)	
	Split rate	Number	Minimum	Maximum
CNN-XGBoost with COM	0.1	4	8.3	9.1
	0.2	8	**7.5**	8.5
	0.3	12	8.1	8.9

Finally, it is important to highlight the sensitivity of the CNN-based prediction method to the number of training and validation data. The limitation of this proposed method is that the variations in the composition or size of these datasets can impact the performance of the RUL prediction model and consequently lead to fluctuations in the predictive accuracy when applying the model to the test data. To illustrate this sensitivity, the values of MAPE for different selections of validation data are shown in Table 5. The results indicate that the MAPE values of RUL prediction range from 7.5 to 9.1 cycles for different split rates. The primary reason for this sensitivity is that the current dataset consists of only 41 data points for training and validation. CNN, being a type of DL model, typically requires a much larger amount of data to effectively learn and generalize patterns. With limited data, the model may not capture the underlying complexities of the battery data. Therefore, to enhance the robustness and accuracy of the proposed method, it is crucial to use a larger dataset.

5. Conclusion

In this paper, an innovative COM-integrated CNN-XGBoost methodology designed for predicting the RUL of Li-ion batteries was introduced. This approach employed a CNN architecture to extract features from the discharge capacity data of the battery through image processing techniques. These features extracted from the CNN model were then concatenated with another set of features derived from the battery charging policy and measured data. The aggregated feature set becomes the input layer for the developed XGBoost model. Afterward, this developed CNN-XGBoost model is augmented by the COM, in which it fine-tunes the CNN's hyperparameters namely the number of filters, kernel size, and number of units in the dense layer. Obtained results affirmed the effectiveness of this proposed approach, with the COM-integrated CNN-XGBoost model demonstrating RMSE and MAPE values of 106 cycles and 7.5%, respectively. To the best of the authors'

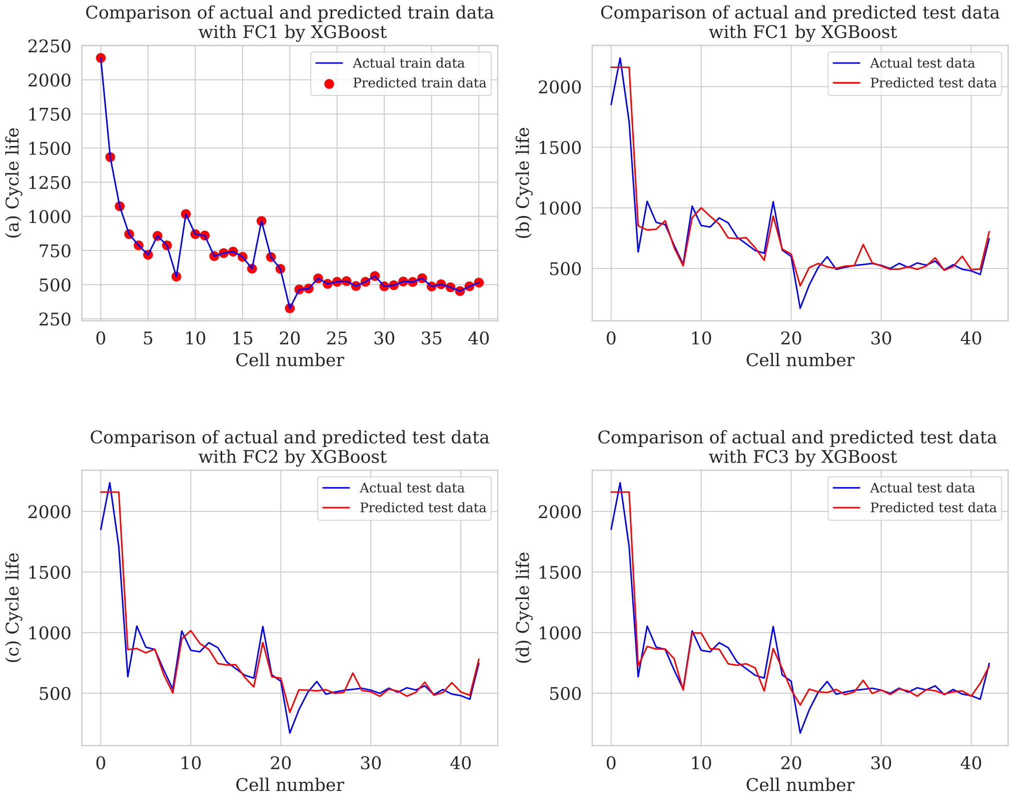


Fig. 9. Early prediction of battery RUL for train dataset with FC1 (a), and test dataset with FC1 (b), FC2 (c), and FC3 (d).

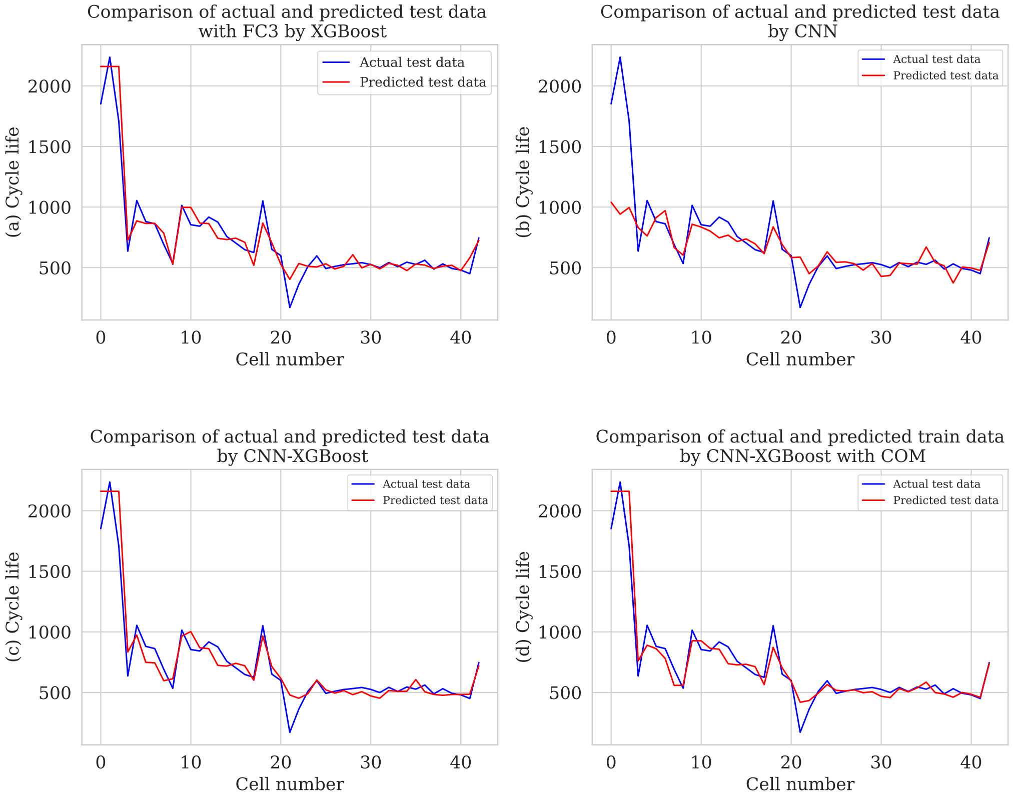


Fig. 10. Early prediction of battery RUL with XGBoost model (a), CNN model (b) and CNN-XGBoost model (c), CNN-XGBoost model with COM (d) on test datasets.

knowledge, the presented results are the lowest values in terms of RMSE and MAPE for RUL prediction of Li-ion batteries compared to reported data-driven methods in the literature on the same datasets. However, the sensitivity of the model to the limited training data shows the need for a larger dataset to enhance the robustness and accuracy of RUL prediction of batteries.

This model could be applied to electric vehicles or other battery datasets with partial discharge or unstable discharge patterns, but this would require more complex data preprocessing to create the 2D image dataset and the acquisition of a larger dataset in real applications. Future work will focus on addressing these challenges by developing a novel data preprocessing pipeline to handle the randomness in daily discharging patterns and by sourcing more complex datasets. This will enable the application of the proposed model in more realistic and varied scenarios, thereby improving its robustness and practical utility.

Abbreviations

The following abbreviations are used in this paper:

Li-ion	Lithium-ion
SOH	State-of-health
SOC	State-of-charge
DOD	Depth of Discharge
PB	Physic-Based
DDB	Data-Driven-Based
EM	Electrochemical Model
SE	Semi-Empirical
ECM	Equivalent Circuit Model
PTD	Persudo Two-Dimension
BMS	Battery Management System
GPR	Gaussian Process Regression
HI	Health Indicator
FBM	Fractional Brownian Motion
FFO	Fruit-Fly Optimization
PSO	Particle swarm optimization
ABC	Artificial Bee Colony
RFR	Random Forest Regressor
LR	Linear Regressor
BR	Bagging Regressor
RUL	Remaining useful life
COM	Coati Optimization Method
C-rate	Current Rate
MC	Monte Carlo
ML	Machine learning
DL	Deep learning
NN	Neural network
DNN	Deep neural network
CNN	Convolutional neural network
CNN-XGBoost	Convolutional neural network-Extreme Gradient Boosting
TCNN	Temporal convolutional neural network
RNN	Recurrent neural network
LSTM	Long-Short-Term Memory
XGBoost	Extreme Gradient Boosting
LightGBM	Light Gradient-Boosting Machine
TCN	Temporal Convolutional Network
MLP	Multi-Layer Precptron
RF	Random Forest
SVR	Support Vector Regression
GRU	Gated recurrent unit
SVM	Support Vector Machine
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error

CRediT authorship contribution statement

Vahid Safavi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Arash Mohammadi Vaniar:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Najmeh Bazmohammadi:** Writing – review & editing, Writing – original draft, Visualization, Resources, Investigation, Formal analysis, Data curation, Conceptualization. **Juan C. Vasquez:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Formal analysis, Conceptualization. **Ozan Keysan:** Writing – review & editing, Supervision, Resources, Project administration, Formal analysis, Conceptualization. **Josep M. Guerrero:** Writing – review & editing, Supervision, Resources, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported by VILLUM FONDEN, Denmark under the VILLUM Investigator Grant (no. 25920): Center for Research on Microgrids (CROM).

References

[1] Y. Zheng, N.A. Weinreich, A. Kulkarni, Y. Che, H. Sorouri, X. Sui, R. Teodorescu, Sensorless state of temperature estimation for smart battery based on electrochemical impedance, in: 2023 25th European Conference on Power Electronics and Applications, EPE'23 ECCE Europe, IEEE, 2023, pp. 1–8.

[2] L. Ren, L. Zhao, S. Hong, S. Zhao, H. Wang, L. Zhang, Remaining useful life prediction for lithiumion battery: A deep learning approach, IEEE Access 6 (2018) 50587–50598.

[3] C. Zhang, S. Zhao, Y. He, An integrated method of the future capacity and rul prediction for lithium-ion battery pack, IEEE Trans. Veh. Technol. 71 (3) (2021) 2601–2613.

[4] S.F. Schuster, T. Bach, E. Fleder, J. Müller, M. Brand, G. Sextl, A. Jossen, Nonlinear aging characteristics of lithium-ion cells under different operational conditions, J. Energy Storage 1 (2015) 44–53.

[5] W. Guo, Z. Sun, S.B. Vilsen, J. Meng, D.I. Stroe, Review of grey box lifetime modeling for lithium-ion battery: Combining physics and data-driven methods, J. Energy Storage 56 (2022) 105992.

[6] M. Doyle, T.F. Fuller, J. Newman, Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell, J. Electrochem. Soc. 140 (6) (1993) 1526.

[7] T.F. Fuller, M. Doyle, J. Newman, Simulation and optimization of the dual lithium ion insertion cell, J. Electrochem. Soc. 141 (1) (1994) 1.

[8] Z. Khalik, M. Donkers, J. Sturm, H.J. Bergveld, Parameter estimation of the doyle–fuller–newman model for lithium-ion batteries by parameter normalization, grouping, and sensitivity analysis, J. Power Sources 499 (2021) 229901.

[9] S. Abu-Sharkh, D. Doerffel, Rapid test and non-linear model characterisation of solid-state lithiumion batteries, J. Power Sour. 130 (1–2) (2004) 266–274.

[10] M.-K. Tran, A. Mevawala, S. Panchal, K. Raahemifar, M. Fowler, R. Fraser, Effect of integrating the hysteresis component to the equivalent circuit model of lithium-ion battery for dynamic and non-dynamic applications, J. Energy Storage 32 (2020) 101785.

[11] S. Wang, P. Takyi-Aninakwa, Y. Fan, C. Yu, S. Jin, C. Fernandez, D.-I. Stroe, A novel feedback correction-adaptive kalman filtering method for the whole-life-cycle state of charge and closedcircuit voltage prediction of lithium-ion batteries based on the second-order electrical equivalent circuit model, Int. J. Electr. Power Energy Syst. 139 (2022) 108020.

[12] C. Zhang, T. Amietszajew, S. Li, M. Marinescu, G. Offer, C. Wang, Y. Guo, R. Bhagat, Real-time estimation of negative electrode potential and state of charge of lithium-ion battery based on a half-cell-level equivalent circuit model, J. Energy Storage 51 (2022) 104362.

[13] Y. Liu, K. Xie, Y. Pan, H. Wang, Y. Li, C. Zheng, Simplified modeling and parameter estimation to predict calendar life of li-ion batteries, Solid State Ion. 320 (2018) 126–131.

[14] J. de Hoog, J.-M. Timmermans, D. Ioan-Stroe, M. Swierczynski, J. Jaguemont, S. Goutam, N. Omar, J. Van Mierlo, P. Van Den Bossche, Combined cycling and calendar capacity fade modeling of a nickel-manganese-cobalt oxide cell with real-life profile validation, Appl. Energy 200 (2017) 47–61.

[15] X. Han, M. Ouyang, L. Lu, J. Li, A comparative study of commercial lithium ion battery cycle life in electric vehicle: Capacity loss estimation, J. Power Sources 268 (2014) 658–669.

[16] C. Chang, Y. Wu, J. Jiang, Y. Jiang, A. Tian, T. Li, Y. Gao, Prognostics of the state of health for lithium-ion battery packs in energy storage applications, Energy 239 (2022) 122189.

[17] B. Gou, Y. Xu, X. Feng, State-of-health estimation and remaining-useful-life prediction for lithiumion battery using a hybrid data-driven method, IEEE Trans. Veh. Technol. 69 (10) (2020) 10854–10867.

[18] H. Meng, Y.-F. Li, A review on prognostics and health management (phm) methods of lithium-ion batteries, Renew. Sustain. Energy Rev. 116 (2019) 109405.

[19] S. Jafari, Y.-C. Byun, S. Ko, A novel approach for predicting remaining useful life and capacity fade in lithium-ion batteries using hybrid machine learning, IEEE Access (2023).

[20] S. Greenbank, D. Howey, Automated feature extraction and selection for data-driven models of rapid battery capacity fade and end of life, IEEE Trans. Ind. Inform. 18 (5) (2021) 2965–2973.

[21] Y. Chen, W. Duan, Y. He, S. Wang, C. Fernandez, A hybrid data driven framework considering feature extraction for battery state of health estimation and remaining useful life prediction, Green Energy Intell. Transp. (2024) 100160.

[22] M. Zhao, Y. Zhang, H. Wang, Battery degradation stage detection and life prediction without accessing historical operating data, Energy Storage Mater. (2024) 103441.

[23] Y. Zhang, M. Zhao, R. Xiong, Online data-driven battery life prediction and quick classification based on partial charging data within 10 min, J. Power Sources 594 (2024) 234007.

[24] L. Su, M. Wu, Z. Li, J. Zhang, Cycle life prediction of lithium-ion batteries based on data-driven methods, ETransportation 10 (2021) 100137.

[25] Q. Zhang, L. Yang, W. Guo, J. Qiang, C. Peng, Q. Li, Z. Deng, A deep learning method for lithiumion battery remaining useful life prediction based on sparse segment data via cloud computing system, Energy 241 (2022) 122716.

[26] M.A. Patil, P. Tagade, K.S. Hariharan, S.M. Kolake, T. Song, T. Yeo, S. Doo, A novel multistage support vector machine based approach for li ion battery remaining useful life estimation, Appl. Energy 159 (2015) 285–297.

[27] T. Tang, H. Yuan, The capacity prediction of li-ion batteries based on a new feature extraction technique and an improved extreme learning machine algorithm, J. Power Sources 514 (2021) 230572.

[28] Y. Ma, M. Yao, H. Liu, Z. Tang, State of health estimation and remaining useful life prediction for lithium-ion batteries by improved particle swarm optimization-back propagation neural network, J. Energy Storage 52 (2022) 104750.

[29] H. Wang, W. Song, E. Zio, A. Kudreyko, Y. Zhang, Remaining useful life prediction for lithium-ion batteries using fractional brownian motion and fruit-fly optimization algorithm, Measurement 161 (2020) 107904.

[30] B. Long, X. Gao, P. Li, Z. Liu, Multi-parameter optimization method for remaining useful life prediction of lithium-ion batteries, IEEE Access 8 (2020) 142557–142570.

[31] J. Peng, Z. Zheng, X. Zhang, K. Deng, K. Gao, H. Li, B. Chen, Y. Yang, Z. Huang, A data-driven method with feature enhancement and adaptive optimization for lithium-ion battery remaining useful life prediction, Energies 13 (3) (2020) 752.

[32] Y. Wang, Y. Ni, S. Lu, J. Wang, X. Zhang, Remaining useful life prediction of lithium-ion batteries using support vector regression optimized by artificial bee colony, IEEE Trans. Veh. Technol. 68 (10) (2019) 9543–9553.

[33] X. Zhang, P. Xiao, Y. Yang, Y. Cheng, B. Chen, D. Gao, W. Liu, Z. Huang, Remaining useful life estimation using cnn-xgb with extended time window, IEEE Access 7 (2019) 154386–154397.

[34] Z. Ma, H. Zeng, J. Guo, T. Gu, S. Mao, T. Yang, The application of cnn-lightgbm algorithm in remaining useful life prediction, in: 2020 7th International Conference on Information Science and Control Engineering, ICISCE, IEEE, 2020, pp. 1411–1418.

[35] K.A. Severson, P.M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M.H. Chen, M. Aykol, P.K. Herring, D. Fraggedakis, et al., Data-driven prediction of battery cycle life before capacity degradation, Nat. Energy 4 (5) (2019) 383–391.

[36] V. Safavi, A. Mohammadi Vaniar, N. Bazmohammadi, J.C. Vasquez, J.M. Guerrero, Battery remaining useful life prediction using machine learning models: A comparative study, Information 15 (3) (2024) http://dx.doi.org/10.3390/info15030124.

[37] Z. Wei, C. Liu, X. Sun, Y. Li, H. Lu, Two-phase early prediction method for remaining useful life of lithium-ion batteries based on a neural network and gaussian process regression, Front. Energy (2023) 1–16.

[38] D. Chen, W. Zhang, C. Zhang, B. Sun, X. Cong, S. Wei, J. Jiang, A novel deep learning-based life prediction method for lithium-ion batteries with strong generalization capability under multiple cycle profiles, Appl. Energy 327 (2022) 120114.

[39] V.T. Ha, P.T. Giang, Experimental study on remaining useful life prediction of lithium-ion batteries based on three regression models for electric vehicle application, Appl. Sci. 13 (13) (2023) 7660.

[40] B. Celik, R. Sandt, L.C.P. dos Santos, R. Spatschek, Prediction of battery cycle life using earlycycle data, machine learning and data management, Batteries 8 (12) (2022) 266.

[41] X. Sui, S. He, Y. Zheng, Y. Che, R. Teodorescu, Early prediction of lithium-ion batteries lifetime via few-shot learning, in: IECON 2023-49th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2023, pp. 1–6.

[42] C.V.G. Zelaya, Towards explaining the effects of data preprocessing on machine learning, in: 2019 IEEE 35th International Conference on Data Engineering, ICDE, IEEE, 2019, pp. 2086–2090.

[43] I.F. Ilyas, T. Rekatsinas, Machine learning and data cleaning: Which serves the other? ACM J. Data Inf. Qual. (JDIQ) 14 (3) (2022) 1–11.

[44] V. Safavi, N. Bazmohammadi, J.C. Vasquez, J.M. Guerrero, Battery state-of-health estimation: A step towards battery digital twins, Electronics 13 (3) (2024) 587.

[45] D. Singh, B. Singh, Investigating the impact of data normalization on classification performance, Appl. Soft Comput. 97 (2020) 105524.

[46] S. Khalid, T. Khalil, S. Nasreen, A survey of feature selection and feature extraction techniques in machine learning, in: 2014 Science and Information Conference, IEEE, 2014, pp. 372–378.

[47] I.M. Nasir, M.A. Khan, M. Yasmin, J.H. Shah, M. Gabryel, R. Scherer, R. Damaševičius, Pear- son correlation-based feature selection for document classification using balanced training, Sensors 20 (23) (2020) 6793.

[48] L. Alzubaidi, J. Zhang, A.J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M.A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions, J. Big Data 8 (2021) 1–74.

[49] S.S. Azmi, S. Baliga, An overview of boosting decision tree algorithms utilizing adaboost and xgboost boosting strategies, Int. Res. J. Eng. Technol. 7 (5) (2020) a.

[50] M. Dehghani, Z. Montazeri, E. Trojovská, P. Trojovskỳ, Coati optimization algorithm: A new bio- inspired metaheuristic algorithm for solving optimization problems, Knowl.-Based Syst. 259 (2023) 110011.

[51] E. Ndiaye, T. Le, O. Fercoq, J. Salmon, I. Takeuchi, Safe grid search with optimal complexity, in: International Conference on Machine Learning, PMLR, 2019, pp. 4771–4780.

[52] T. Chai, R.R. Draxler, Root mean square error (rmse) or mean absolute error (mae), Geosci. Model Dev. Discuss. 7 (1) (2014) 1525–1534.

[53] M. Alipour, S.S. Tavallaey, A.M. Andersson, D. Brandell, Improved battery cycle life predic- tion using a hybrid data-driven model incorporating linear support vector regression and gaussian, ChemPhysChem 23 (7) (2022) e202100829.

[54] Q. Xu, M. Wu, E. Khoo, Z. Chen, X. Li, A hybrid ensemble deep learning approach for early prediction of battery remaining useful life, IEEE/CAA J. Autom. Sin. 10 (1) (2023) 177–187.

[55] G. Ma, Z. Wang, W. Liu, J. Fang, Y. Zhang, H. Ding, Y. Yuan, A two-stage integrated method for early prediction of remaining useful life of lithium-ion batteries, Knowl.-Based Syst. 259 (2023) 110012, 25.