A Generalizable Method for Capacity Estimation and RUL Prediction in Lithium-ion Batteries

Yixiu Wang,[†] Jiangong Zhu,[‡] Liang Cao,[†] Jianfeng Liu,[¶] Pufan You,[§] Bhushan Gopaluni,[*,†] and Yankai Cao[*,†]

[†]*Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC, V6T 1Z3, Canada*

[‡]*Clean Energy Automotive Engineering Center, Tongji University, 201804, Shanghai, China*

[¶]*Amazon, Seattle, WA, 98109, USA*

[§]*Department of Statistics, University of Manitoba, Winnipeg, MB, R3T 2N2, Canada*

E-mail: bhushan.gopaluni@ubc.ca; yankai.cao@ubc.ca

Abstract

Data-driven methods have attracted much attention in capacity estimation and remaining useful life (RUL) prediction of Lithium-ion batteries. However, existing studies rely on complex machine learning models (e.g., Gaussian Process Regression, Neural Networks, etc) that are applicable to specific observed operating conditions, and the prediction accuracy can be affected by different usage scenarios. This paper proposes to adopt a linear and robust machine learning technique, partial least squares regression, for battery capacity estimation and RUL prediction based on the partial incremental capacity curve. The features can be easily obtained by interpolation of the measured charging profiles without data smoothing, and the bootstrapping technique is used to give confidence intervals of predictions, which helps to evaluate the robustness and reliability of the model. The proposed method is validated on three battery datasets with different operating conditions provided by NASA. We train the model on one battery and test its performance on the other two batteries without changing the model weights. Experimental results show that the suggested classical method exhibits greater generalizability, compared to complex and sophisticated methods proposed in the literature.

1. Introduction

Lithium-ion batteries, with their superior high-energy and high-power density characteristics, have been widely used in many fields such as electric vehicles, portable electronic devices, and power grids, and are widely considered to be the best energy storage solution.[1] However, the inevitable degradation poses a challenge that cannot be overlooked.[2] This degradation becomes increasingly evident over time, manifesting as a decrease in battery performance, which could potentially lead to increased operational costs. To ensure the efficient and safe usage of Lithium-ion batteries, establishing accurate models to estimate the battery's State of Health (SOH) and predict its remaining useful life (RUL) has become crucial.

Battery capacity is an important indicator of SOH, which can be obtained experimentally through the integration of the discharging current with respect to time over a complete discharging process. However, measuring capacity is challenging in practical applications as a complete discharging curve is seldom available for battery health monitoring due to uncertainties in operational conditions.[3] Over the past decades, extensive research has been dedicated to capacity estimation using measurements from the voltage and current. These methods fundamentally rely on the relationship between the electricity charged or discharged by the battery and the voltage variations during these processes.[4] Broadly, these techniques are categorized into two main types, namely model-based methods and data-driven methods. A prevalent model in the model-based capacity estimation methods is the equivalent circuit model (ECM),[5] which simulates the behavior of a battery using basic electronic components like resistors and capacitors. After formulating the battery's state-space equations through ECM, several recursive adaptive filters such as the extended Kalman filter (EKF)[6,7] and particle filter (PF)[8] are adopted. These filters facilitate the identification of model parameters and subsequently update the battery's capacity. For instance, Plett et al.[6,9] constructed a battery cell model in a discrete-time state-space form, considering the quickly varying state quantities like the State of Charge (SOC), and slowly changing time-varying parameters of the system such as capacity. Then, they proposed the utilization of a dual EKF for the joint estimation of states and parameters. The experimental results confirm that this approach allows the capacity estimation to converge to accurate values, maintaining minor variations over time, even under a complex Urban Dynamometer Driving Schedule (UDDS) cycle. However, despite the high accuracy achieved by the model-based capacity estimation methods, it involves substantial computational processes, making it less suitable for practical online applications.

In recent years, data-driven methods have become popular in the research field of batteries with the development of machine learning and artificial intelligence, which can reflect the intrinsic correlation between the measurements and the battery capacity without expert knowledge of aging mechanisms. Extracting valuable features from measurements is the first step to build a data-driven model, and much attention has been focused on extracting health features from charging curves since the charging process of a battery is expected to occur regularly. Specifically, considering the state-of-the-art literature, there are typically three sources of features. These include features from the Constant Current-Constant Voltage (CC-CV) charging curves,[10,11] features from the relaxation process curves after the charging process,[3,12] and features from the CC charging curves.[13–15] Yang et al.[10] adopted Gaussian Process Regression (GPR) for capacity estimation with four specific features extracted from the CC-CV charge curve, including the time of CC model duration, the time of CV mode duration, the slope of the curve at the end of CC charge mode and the vertical slope at the corner of the cc charge curve. Guo et al.[11] extracted 14 health features from the charging process and then proposed to employ the Relevance Vector Machine (RVM) for capacity estimation. While these methods have achieved high accuracy, they require the use of a whole charging curve to obtain health features, which is not guaranteed during the vehicle operation. Zhu et al.[3] proposed using statistical features of the relaxation voltage curve after the charging process for capacity estimation, while Baghdadi et al.[12] utilized the open-circuit voltage after a 30-minute rest as a feature. These methods have shown good results, but ensuring adequate rest after a full charging process in actual battery usage is challenging. As for the CC charging curve, incremental capacity (IC) analysis is a common-used technique for feature extraction, and Pei et al.[13] proposed to use the partial charged capacity from the threshold (IC = 2 Ah/10mV) till to the end of the charge as the feature. However, one limitation of this method is that the entire CC curve is required to smooth the IC curve and calculate the partial charge capacity. Considering that partial charging is more common in practical applications, methods based on feature extraction from partial charging curves have attracted significant interest in the field of battery capacity estimation. Robert et al.[14] proposed a GPR with a Matérn (5/2) kernel function for battery capacity estimation based on the time vectors extracted from the partial constant current (CC) charging curve, while Yi et al.[15] used random forest regression (RFR) with the relative capacity values from the partial CC charging curve for accurate capacity estimation. However, the complex structure of these models often makes them suitable only for specific observed operating conditions, and a large amount of experimental data under various conditions is required to train the model, which can be time-consuming to collect in practice. Therefore, there is a need to develop a generalizable regression model for accurate capacity estimation.

Besides capacity estimation, another critical task in battery health monitoring is to predict the RUL, which is the number of charging/discharging cycles that a cell can undergo before the capacity decays to a specific value Typically, there are two approaches for RUL prediction. One is to predict the capacity degradation trajectory and count the number of cycles to the end of life as RUL, and the other is to directly predict the value of RUL. Many previous studies have used different machine learning techniques such as Support Vector Machine (SVM),[16–18] GPR,[19–21] and NNs,[22–24] to predict future values of capacity and RUL based on previous capacity trajectory. However, one disadvantage of these models is that they require nearly half of the capacity trajectory to obtain an accurate RUL prediction and can not be used in the early-cycle stage of the battery. In recent years, Kristen et al.[25] and Yu et al.[26] proposed to use the features from the discharging curves of the first 100 cycles to directly predict RUL, and the results showed high accuracy. One weakness of these methods is that they can only give one RUL prediction for each cell, i.e., the cycle life of a new cell, and can not dynamically update the RUL prediction at different aging stages. Furthermore, the complete discharge curves for multiple cycles are not guaranteed in practice. Zhang et al.[27] introduced an RUL prediction method by combining the electrochemical impedance spectroscopy (EIS) spectrum with GPR, and it can predict the RUL of a battery at each cycle. However, the real-time acquisition of the EIS spectrum is still a challenge in a real-life usage scenario of electric vehicles. It is vital to develop a model to predict RUL based on readily measurable variables, such as voltage, current, etc.

This paper proposes to use a linear and robust machine learning technique, partial least squares regression (PLSR), for capacity estimation and RUL prediction based on the partial incremental capacity (IC) curve. The proposed method is realized and tested on a dataset of three battery cells (#5, #7, #18) provided by NASA. Experimental results show that the model trained on 80% of the data samples of cell #5 can not only give accurate capacity estimation and RUL prediction for the remaining 20% data of cell #5, but also achieve good performance in capacity estimation and RUL prediction for both cell #7 and cell #18 without changing the model weights. The principal contributions of this study can be summarized as follows:

- The proposed method can estimate capacity and predict RUL based on the same features.

- The features used in the model can be easily measured without the need for additional devices or information from the full cycle.

- The proposed model demonstrates good generalizability under different operating conditions where the battery exhibits various cycling characteristics.

- Confidence intervals for the predictions are provided using the bootstrapping technique, which can be used to evaluate the robustness and reliability of the model.

The remainder of the paper is organized as follows. The cycling dataset utilized in this study is presented in Section 2, followed by the presentation of the proposed methodology in Section 3. The results of capacity estimation and RUL prediction are thoroughly discussed in Section 4 and finally, Section 5 closes this work with some conclusions.

2. Cycling Dataset

To assess the performance of the proposed method in terms of capacity estimation and RUL prediction, this study utilizes the cycling dataset from the Prognostics Center of Excellence(PCoE) at Ames Research Center, NASA.[28] The dataset consists of three commercial 18650-size Lithium-ion batteries from the same material system, specifically cells number #5, #7, #18. It should be noted that we did not use cell #6 from the same group because its capacity range differs from the other three batteries. The selected batteries were subjected to three separate operational procedures: charging, discharging, and impedance measurement, all conducted at a constant room temperature of 24 °C.

Throughout the charging phase, a CC mode was first employed, charging the batteries at 1.5A until achieving a voltage of 4.2V, succeeded by a constant voltage (CV) charging phase maintained at 4.2V until the current reduced to 20mA. For the discharging phase, a CC mode was utilized, operating at 2A until the battery voltage declined to 2.7V, 2.2V, and 2.5V for cells #5, #7, and #18, respectively. Battery capacity was obtained by integrating the discharge current starting from a fully charged state to the state where the battery voltage dropped to 2.7V. An EIS frequency sweep ranging from 0.1Hz to 5kHz was conducted for impedance measurements, which can provide insight into the battery's internal parameters. The repeated cycling of the charging and discharging processes expedited battery aging. The experiment was stopped once the capacity dropped to the end-of-life threshold, marked by a 30% reduction in nominal capacity (from 2Ah to 1.4Ah).

Figure 1 illustrates in detail how the capacity of the three battery cells in the experiment changes with the number of cycles. It is evident that as the cycle number increases, the battery capacity gradually diminishes, revealing irreversible physical and chemical changes occurring internally during the continuous charging and discharging processes. Moreover, each battery cell exhibits distinct degradation characteristics and performances throughout the process. It is noteworthy that slight fluctuations in battery capacity have been observed, possibly due to measurement errors encountered during the experiment. Additionally, different cells display varied capacity values under identical cycling numbers, indicating that relying solely on the number of cycles to evaluate battery capacity is neither accurate nor practical. For a more precise estimation of battery capacity and prediction of RUL, more other relevant cycling information needs to be taken into account, such as voltage and current measurements collected during the charging process.

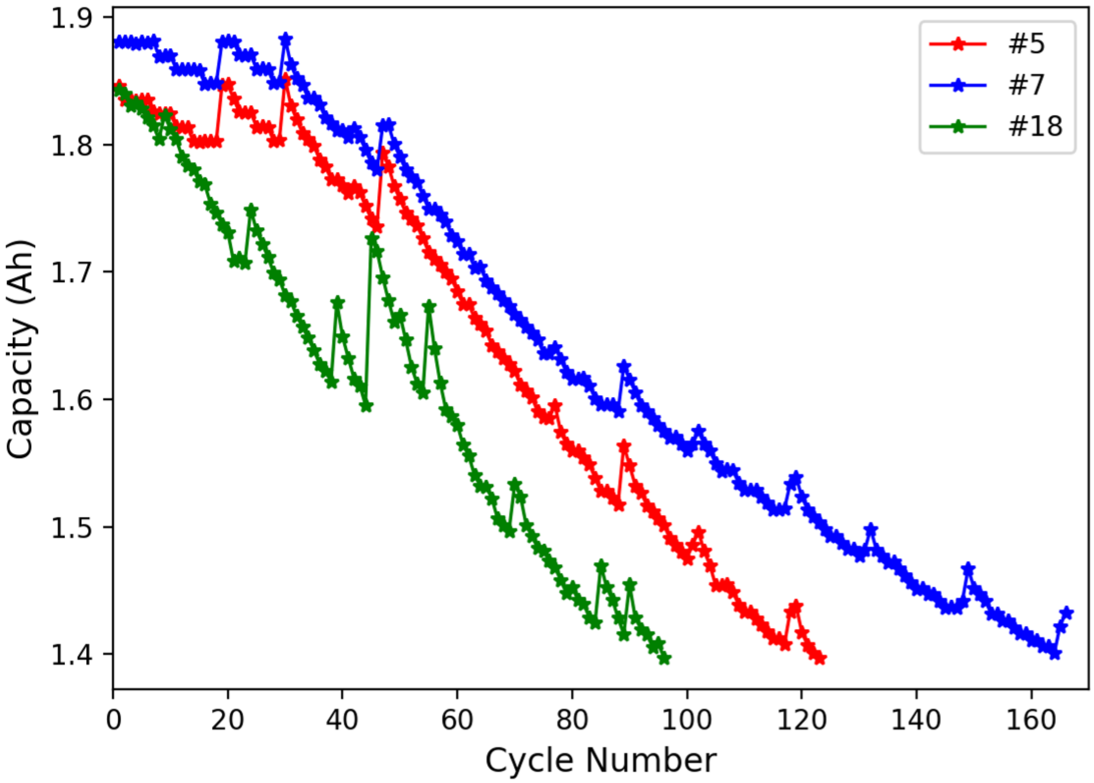


Figure 1: Capacity change of three experimental batteries. Reproduced or adapted with permission from.[29] Copyright 2022, IEEE.

3. Methodology

3.1 Partial Incremental Capacity Curve for Feature Extraction

Incremental capacity (IC) analysis is a valuable technique for investigating the degradation mechanism of lithium-ion batteries. It involves differentiating the change in battery capacity over the change in terminal voltage, with each peak of the IC curve representing a distinct electrochemical reaction occurring within the cell. A multitude of previous studies have effectively utilized the peak position[30] and peak intensity[31] of IC curves as features to estimate battery capacity. However, the practical application of IC analysis is inherently vulnerable to data perturbations, leading to elusive peaks due to pervasive measurement noises. Conventional approaches often necessitate the application of suitable data smoothing methods, such as the Gaussian filter, to acquire a more streamlined IC curve, but its application in real-time is challenging due to the current battery management system (BMS) having restricted computational capabilities. To overcome this obstacle, this paper proposes a novel approach aimed at estimating the battery capacity and predicting the RUL by directly leveraging the IC values for a specific voltage region, effectively bypassing the need for data smoothing processes.

In reality, BMS samples current and voltage at equal time intervals, resulting in varying lengths of sampling points at different charging times. In other words, there is no guarantee of having the same sampling points for a specific voltage region over all cycles. In order to have fixed-length input features for the model, an interpolation-based IC curve acquisition algorithm is proposed in this paper. Primarily, taking into consideration the CC charging profile shown in Figure 2 (a), it can be denoted as $\mathcal{D} = \{(t_i, I_i, V_i), i = 1, 2, \cdots, n\}$, where I_i and V_i are the measured current and voltage at the sampling time point t_i, and n is the total number of sampling points. Then, focusing on a specific voltage region from V_l to V_h, a process of discretization is applied with intervals of ΔV to get the desired voltage vector $\hat{\mathcal{V}} = \{V_l, V_l + \Delta V, \cdots, V_h\} = \{\hat{V}_i, i = 1, 2, \cdots, k\}$, where $\hat{}$ represents discrete values and the number of discrete points k can be calculated by $k = \lfloor (V_h - V_l)/\Delta V + 1 \rfloor$. Subsequent to this, the time $\hat{t}_i$ to attain the desired $\hat{V}_i$ can be obtained by interpolating the two nearest voltage measurements V_i and V_{i+1} in $\mathcal{D}$, depicted by:

$$\frac{\hat{V}_i - V_i}{V_{i+1} - V_i} = \frac{\hat{t}_i - t_i}{t_{i+1} - t_i} \tag{1}$$

$$\hat{t}_i = \frac{\hat{V}_i - V_i}{V_{i+1} - V_i}(t_{i+1} - t_i) + t_i \tag{2}$$

In a manner akin to the above, $\hat{I}_i$ is computed as per the following expression:

$$\hat{I}_i = \frac{\hat{V}_i - V_i}{V_{i+1} - V_i}(I_{i+1} - I_i) + I_i \tag{3}$$

Upon deriving the desired current vector $\hat{\mathcal{I}} = \{\hat{I}_i, i = 1, 2, \cdots, k\}$ and time vector $\hat{\mathcal{T}} = \{\hat{t}_i, i = 1, 2, \cdots, k\}$, IC at $\hat{V}_i$ can be approximated by

$$\left.\frac{dQ}{dV}\right|_{\hat{V}_i} \approx \left.\frac{\Delta Q}{\Delta V}\right|_{\hat{V}_i} = \frac{\hat{I}_i(\hat{t}_{i+1} - \hat{t}_i)}{\Delta V} \tag{4}$$

Figure 2 (b) shows the IC curves for cycle 1 and cycle 100 of cell #5 when ΔV is 0.002V. As shown in the figure, it can be noticed that the IC values at cycle 1 are generally higher than those at cycle 100, and the curves show substantial variation in the voltage range from 3.8V to 4.2V, indicating the possibility of using IC values to estimate the capacity and predict RUL. We will discuss on how to select the specific voltage range in detail later.

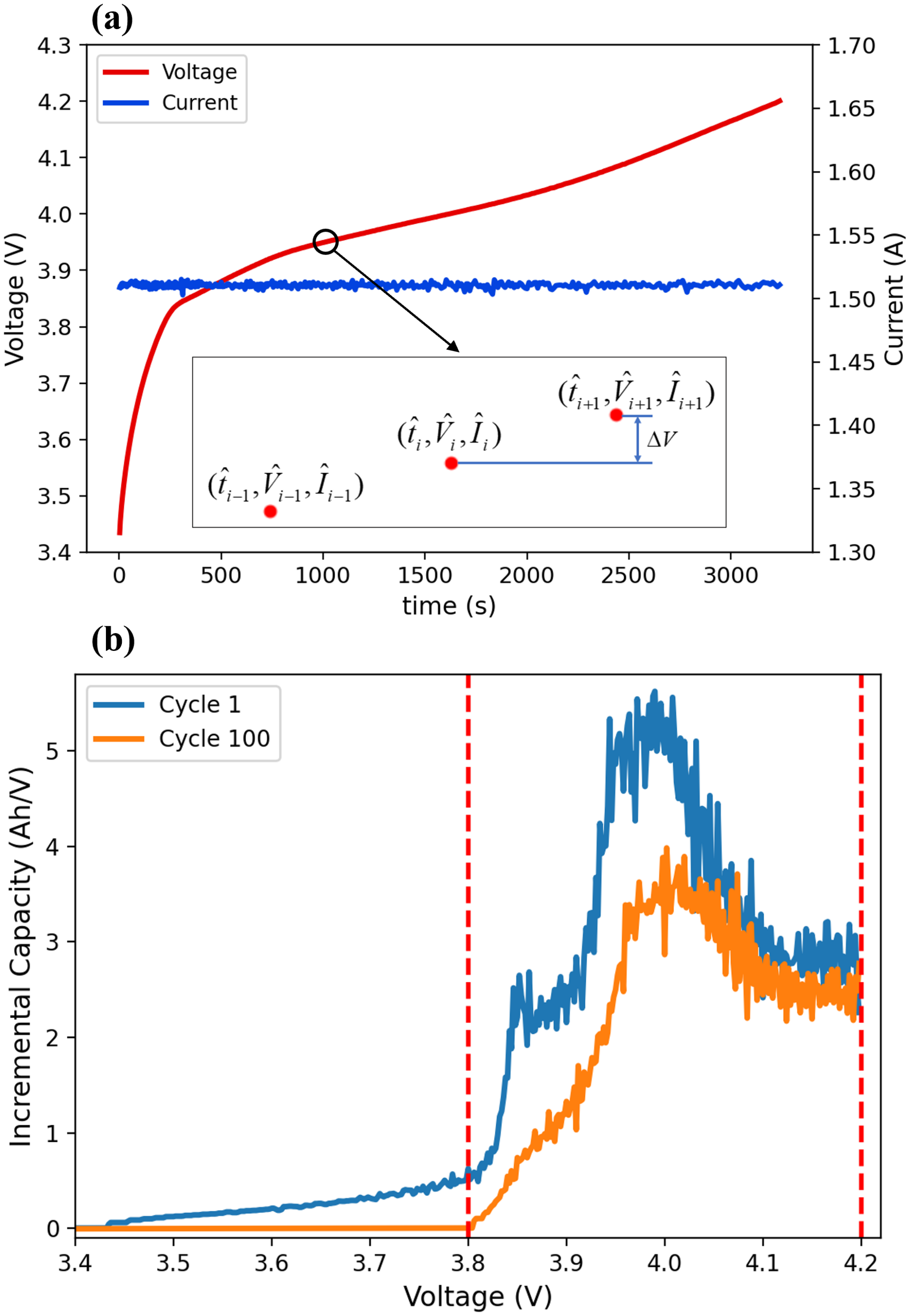


Figure 2: Illustration of feature extraction: (a) Constant current charging profile for cycle 1 of cell #5. (b) Incremental capacity curves for cycle 1 and 100 of cell #5.

3.2 Partial Least Squares Regression

PLSR[32] serves as an advanced extrapolation of the conventional multiple linear regression (MLR), exhibiting excellent performance particularly when dealing with a multitude of variables and few observations or significant collinearity amongst variables. A detailed description of PLSR, with MLR as the basis, is presented in this section.

In addressing a regression problem, considering that we are given a feature matrix X comprised of elements x_{ij}, along with a target y constituted of elements y_i, where i ranging from 1 to n denotes the index of samples, and j ranging from 1 to p represents different features, MLR helps to create a linear connection between these features and the target, denoted by

$$y = Xb + e \tag{5}$$

where b is the regression coefficient to be estimated, and e is the residual. Employing the classical least squares method, one can get b by

$$b = (X^T X)^{-1} X^T y \tag{6}$$

This equation effectively derives an unbiased estimation of b under the conditions of the Gauss–Markov theorem. However, in a setting with limited samples and substantial collinearity, deriving a credible estimation of b remains challenging. PLSR emerges as a remedy, projecting X and y into a realm of reduced dimensionality to facilitate the execution of regression.

The procedure of PLSR begins with the identification of the first component c_1 of X with $c_1 = Xw_1$. The solution is achieved by maximizing $Cov(c_1, y)$, leading to

$$w_1 = \frac{X^T y}{||X^T y||},\ c_1 = Xw_1 = \frac{XX^T y}{||X^T y||} \tag{7}$$

Regression is subsequently enacted separately upon X and y relative to c_1, yielding

$$\begin{cases} X = c_1 p_1^T + X_1 \\ y = c_1 r_1 + y_1 \end{cases} \tag{8}$$

where the coefficients are

$$\begin{cases} p_1 = \dfrac{X^T c_1}{||c_1||^2} \\ r_1 = \dfrac{y^T c_1}{||c_1||^2} \end{cases} \tag{9}$$

In addition, X_1 and y_1 are the residuals unexplained by the first component. By substituting X and y with X_1 and y_1, and iteratively applying the preceding procedure, subsequent components c_i along with their weights w_i, p_i and $r_i$can be acquired, where $i = 2, \cdots, m$ and m is the number of components.

Considering that

$$\begin{aligned} c_i &= X_{i-1} w_i = X_{i-2}(E - w_{i-1} p_{i-1}^T) w_i \\ &= X \prod_{k=1}^{i-1} (E - w_k p_k^T) w_i \end{aligned} \tag{10}$$

we can let

$$w_i^* = \prod_{k=1}^{i-1} (E - w_k p_k^T) w_i \tag{11}$$

then we can have $c_i = X w_i^*$ and

$$\begin{aligned} y &= c_1 r_1 + c_2 r_2 + \cdots + c_m r_m + y_m \\ &= X w_1^* r_1 + X w_2^* r_2 + \cdots + X w_m^* r_m + y_m \\ &= X \left(\sum_{i=1}^{m} w_i^* r_i \right) + y_m = Xb + y_m \end{aligned} \tag{12}$$

where $b = \sum_{i=1}^{m} w_i^* r_i$ is the coefficient of PLSR.

3.3 Cross-Validation

In machine learning, it is common to train a model on a training set and then test its performance on an independent validation set to select the hyperparameters of the model. However, this method carries the risk of overfitting when only a limited amount of data is available for training. Cross-validation (CV)[33] addresses this issue by splitting the data into different subsets and training and testing the model on different subsets in each iteration. Specifically, as shown in Figure 3, the dataset is first randomly divided into K equal subsets (also called folds), and in each iteration, one of the subsets is selected as the validation set, while the remaining K-1 subsets form the training set. The model is trained on the training set and then tested on the validation set to evaluate its performance. This process is repeated K times, with each subset used for testing exactly once. The results from each iteration are averaged to provide an accurate estimate of the model's performance.

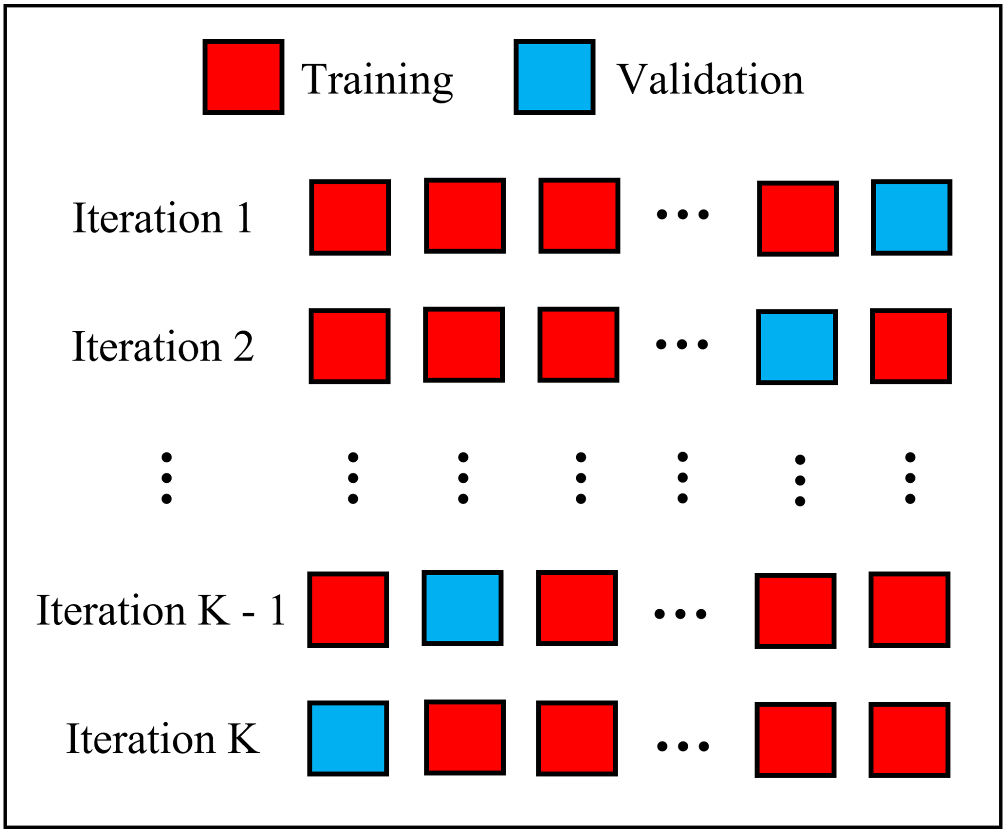


Figure 3: Illustration of K-fold cross-validation.

3.4 Bootstrapping

Although the proposed PLSR can provide the estimation of capacity and the prediction of RUL, it cannot give the confidence interval (CI) for predictions, and we cannot evaluate the robustness of the model. Bootstrapping[34] is a statistical method used to estimate the performance distribution of a machine learning model by resampling with replacement from a given dataset, without depending on data distribution assumptions. Concretely, numerous random samples (known as bootstrap samples) are drawn with replacements from the original dataset to form the bootstrap dataset. A model is trained on each of these bootstrap datasets, and the performance of these models on the independent testing dataset is used to estimate the error distribution of the proposed model structure.

In this work, we employ bootstrapping to provide CIs for capacity estimation and RUL prediction. By creating CIs, we can get a range of the model's prediction results and understand their uncertainty better. To be more specific, we randomly sample 80% of the training dataset each time to form the bootstrap dataset for training the model and record the prediction error on the testing dataset. This process is repeated 3000 times to obtain an error matrix with 3000 elements. The mean error is then calculated using all elements in the error matrix to give an estimation of the model's average performance. Additionally, the values from the 2.5th percentile to the 97.5th percentile of the error matrix give the 95% CI of the model, and one can expect the model's performance to fluctuate within this range.

4. Results and Discussions

4.1 Metrics

To evaluate the predictive performance of the proposed model, Root-Mean-Squared-Error (RMSE) is used in the paper, and it is given by

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2} \tag{13}$$

In our paper, 2 different RMSEs are proposed to describe the accuracy of capacity estimation and RUL prediction, respectively:

1) RMSE of capacity estimation (RMSE-Q) is used to evaluate the performance of the proposed model on estimating the capacity. Here, $\hat{Y}_i$ is the predicted capacity given by the model, Y_i is the corresponding measured capacity, and N is the total number of testing samples. Also, please note that for the ease of understanding the results, the capacity values have been normalized with respect to the nominal capacity, so the RMSE-Q is a percentage and has no units.

2) RMSE of RUL prediction (RMSE-RUL) is used to evaluate the performance of the proposed model on predicting RUL. Here, $\hat{Y}_i$ is the predicted RUL from the model, Y_i is the

observed RUL from the experimental data, and N is the total number of testing samples. In this study, the observed RUL is obtained as the difference between the cycle life and the current cycle number, and its unit is cycle, so the unit of RMSE-RUL is also cycle.

4.2 Influence of Voltage Range and the Number of Components

In this work, partial IC curve is used for battery capacity estimation and RUL prediction. The specific voltage range chosen for feature extraction significantly affects the model accuracy, since different voltage ranges contain different amounts of information. In addition, the hyperparameter of proposed PLSR, the number of components, determines how much information is used for model prediction. Choosing an appropriate number of components can significantly improve the model's prediction performance. To investigate the influence of the voltage range for feature extraction and the number of components of PLSR on model accuracy, we utilized the 5-fold CV to determine the optimal voltage range for feature extraction and the number of components in the PLSR. The candidate voltage ranges contain 3.8V-4.0V, 3.9V-4.1V, and 4.0V-4.2V, and the candidate number of principal components ranges from 1 to 10, resulting in a total of 30 different combinations. For each possible combination of the voltage range and the number of components, the 5-fold CV process is applied and the average error is calculated across all 5 iterations.

Figure 4 shows the 5-fold CV results for RMSE-Q and RMSE-RUL under the 30 combinations mentioned in section 3.3, respectively. Here we use the randomly selected 80% of the cycling data of cell #5 as the training/validation dataset. Since we use the same length of voltage range (0.2V) and the same voltage discretization interval ΔV (0.002V), the input features of the model are 100 IC values for any cycle and the output is the corresponding capacity and RUL for that cycle.

As shown in Figure 4 (a), it is clear that using features extracted from the voltage range of 4.0V-4.2V leads to high RMSE-Q, with almost all results exceeding 2.5%, indicating that the IC curve from 4.0V to 4.2V provides very little information for capacity estimation. In contrast, most of RMSE-Q for the voltage ranges of 3.8V-4.0V and 3.9V-4.1V are below 1%, and it is generally observed that the accuracy of the voltage range of 3.8V-4.0V is slightly better than that of 3.9V-4.1V. In addition, it can be noticed from the blue curve for the voltage range of 3.8V-4.0V that RMSE-Q first demonstrates a downward trend, followed by an upward trend as the number of components increases. This is reasonable as too few components can cause a large amount of information to be lost and thus reduce the model performance, while too many components can make the model more prone to overfitting. Overall, the proposed PLSR performs best on capacity estimation when using the features extracted from the voltage range of 3.8V-4.0V and 4 components, with an RMSE-Q of 0.69%. Figure 4 (b) presents a similar trend to Figure 4 (a), where the highest accuracy for RUL prediction is achieved using IC curves in the voltage range of the 3.8V-4.0V, followed by 3.9V-4.1V and the worst for 4.0V-4.2V. The best model performance is obtained at the voltage range of 3.8V-4.0V and 4 components, with an RMSE-RUL of 6.87 cycles. Therefore, the specific voltage range is selected to be 3.8V-4.0V and the number of components is chosen to be 4 in the following discussion.

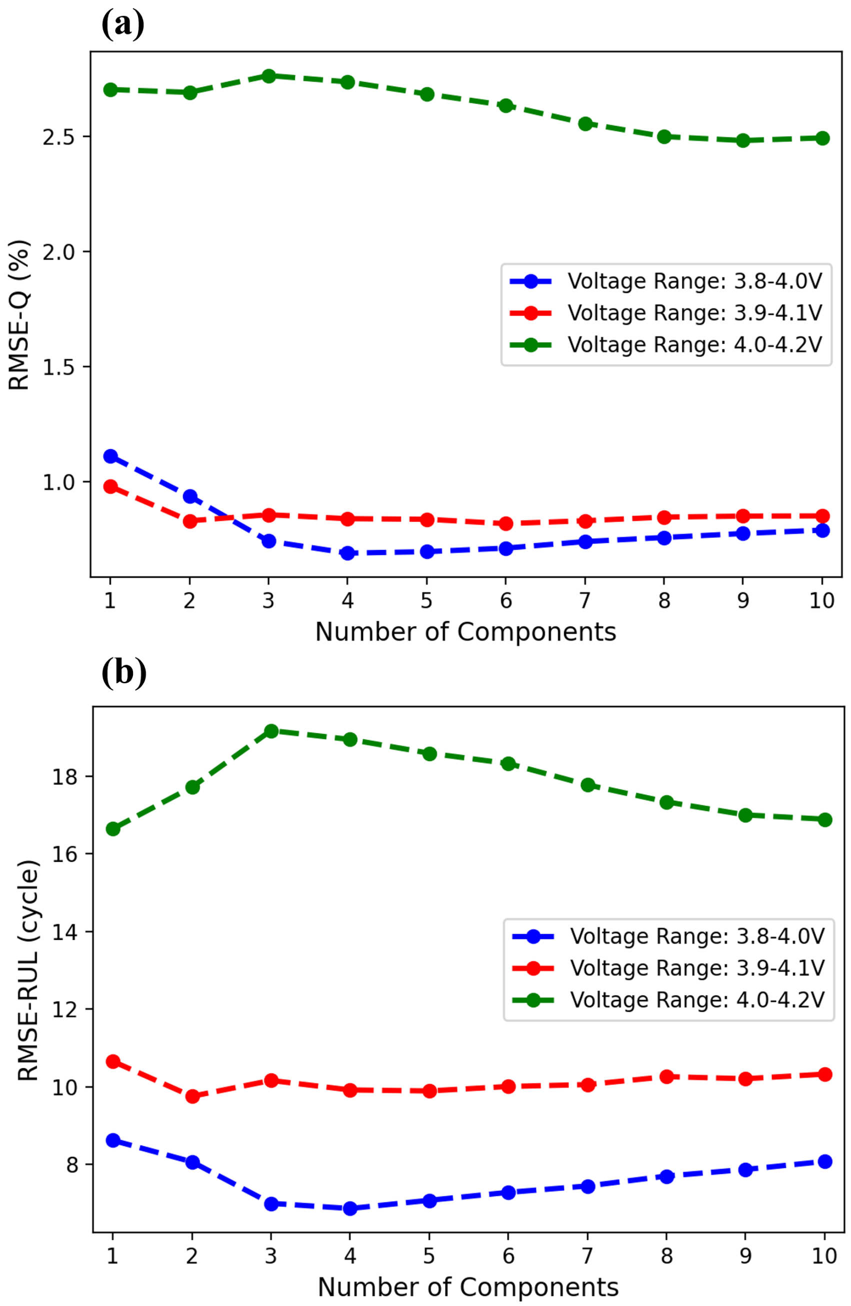


Figure 4: Performance comparison of PLSR with different combinations of the voltage range and the number of components for (a) capacity estimation and (b) RUL prediction.

4.3 Performance of Capacity Estimation

The effectiveness of the proposed PLSR model on battery capacity estimation is benchmarked with the other four commonly used machine learning models, including MLR,[35] Support Vector Regression (SVR),[36] RFR,[37] and Long Short-Term Memory Network (LSTM network).[38] MLR is used to model the relationship between two or more independent variables (features) and a dependent variable (output). In MLR, the output is modeled as a linear combination of features, as well as a constant term known as the intercept. SVR is a special case of SVM in regression, which was designed to find the hyperplane that minimizes the distance between the predicted values and the true values. In this work, we introduce the *rbf* kernel to make the SVR handle nonlinear relationships between the input features and the output. RFR is an ensemble learning method that combines multiple decision trees to make predictions and then averages the results of all decision trees to get the output. Each decision tree in the forest is trained on a subset of the data and a subset of the features to reduce overfitting. As suggested in Li's work,[15] we use an RFR with 500 decision trees for capacity estimation, and the number of random features for each decision tree is 1/3 of the number of all features. LSTM network is a special case of the recurrent neural network, which are designed to process sequential data. All models are trained on a randomly selected 80% of the cycling data of cell #5 and then are tested on the remaining 20% of the data. Moreover, to verify the generalizability of the proposed models, the models trained by data of cell #5 are also tested on data from cell #7 and cell #18 without changing model weights. Here, one cycle is one sample for capacity estimation, and the input features to models are the 100 IC values over a voltage range of 3.8V to 4.0V, and the output is the corresponding capacity.

The correlation analysis is an essential first step in building a data-driven model because collinearity among features can severely affect the performance of the model. Figure 5 displays the Pearson correlation coefficient obtained by using the training data. It can be noted that these features are highly correlated, with correlation coefficients all above 0.6 and most reaching 0.9, implying severe collinearity among the features. It is easy to understand as features are time-series IC values. However, such strong collinearity can hinder some models from finding the correct regression coefficients, thereby reducing the prediction performance of the models, especially for MLR. Therefore, we proposed using Principal

Component Analysis (PCA)[39] and Kernel PCA (KPCA)[40] to reduce the correlation among the features for MLR, SVR, RFR, and the LSTM network. PCA is a statistical technique for reducing the dimensionality of a high-dimensional dataset. It involves transforming the original variables into a new set of uncorrelated variables, known as principal components, which are ordered by the amount of variance they explain in the data. KPCA is an extension of the conventional PCA, allowing for the exploration of non-linear data structures through the use of the kernel trick. In our study, we applied KPCA with the "rbf" kernel, aiding in the uncovering of the intricate non-linear patterns present within the dataset.

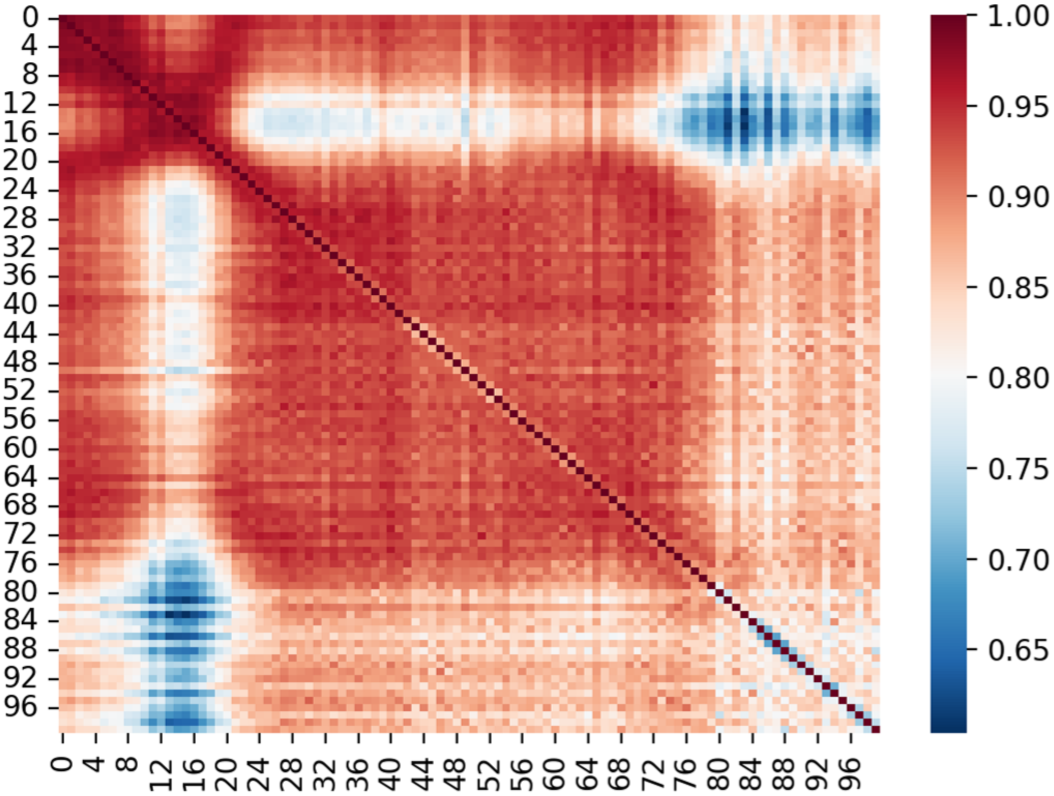


Figure 5: The result of correlation analysis between features.

The number of principal components in PCA and KPCA is crucial for the accuracy of the model. To determine the appropriate number of principal components, a 5-fold CV is applied for MLR to the training dataset. Table 1 presents a performance comparison with varying numbers of principal components. As illustrated in the table, RMSE-Q for both

PCA and KPCA initially exhibits a decreasing trend as the number of principal components increases but subsequently begins to slightly rise, which is consistent with the influence of the number of components on PLSR. Based on the results in the table, for the task of capacity estimation, we retained 6 principal components for PCA and 9 principal components for KPCA in our following discussions.

Table 1: Comparison of PCA and KPCA with different number of principal components

Number of	RMSE-Q (%)		RMSE-RUL(cycle)	
principal components	PCA	KPCA	PCA	KPCA
1	1.12	2.17	8.68	14.89
2	0.97	2.23	8.59	15.14
3	0.97	1.53	8.59	8.84
4	0.94	1.18	8.47	8.22
5	0.91	1.21	7.83	8.24
6	**0.89**	0.75	**7.68**	**6.50**
7	0.89	0.76	7.69	6.57
8	0.90	0.76	7.80	6.72
9	0.93	**0.73**	7.82	6.81
10	0.94	0.74	7.91	6.99

Table 2 compares the performance of different models on capacity estimation. We can notice that all models perform well in estimating capacity of cell #5, in which PLSR has the better performance with an RMSE-Q of 0.59% and 95% CI of [0.51%, 0.69%] compared to other benchmark models, and RFR with PCA has the worst performance with an RMSE-Q of 1.92% and 95% CI of [1.49%, 2.52%], suggesting that the IC values are good features for capacity estimation. Meanwhile, a comparison of the performance of the benchmark models with and without PCA shows that PCA significantly improves the performance of the MLR and the LSTM network, but has no improvement or even decreases the accuracy of the SVR and RFR. One potential reason is that MLR is very sensitive to collinearity, while SVR and RFR, as nonlinear models, can handle collinearity to some extent, and the information loss caused by PCA can make these models perform worse. For the LSTM network, dimensionality reduction through PCA can significantly simplify the model structure, which aids in reducing the risk of the model being overfitted. It can also be observed that KPCA enhances the accuracy of MLR. However, its performance is inferior to that of PCA, suggesting the existence of underlying linear relationships between the features and the capacity.

When testing the generalizability of models to other cells, it is found in the table that the SVR, RFR and the LSTM network can estimate the capacity of cell #7 well, but failed to estimate the capacity of cell #18, especially the LSTM network has an RMSE-Q of 4.82%. In contrast, MLR shows good accuracy on both cell #7 and cell #18, but still performed worse than the proposed PLSR with an RMSE-Q of 1.16% and 1.66%, respectively. Overall, the proposed PLSR shows the best performance across 3 test cells. Additionally, it is found that both PCA and KPCA do not improve the capacity estimation accuracy of the MLR on cell #7 and cell #18. One possible reason is that the distribution of features of cell #5 is different from that of other cells, and hence the PCA and KPCA transformation obtained from training datasets on cell #5 may lose some information that is helpful for capacity estimation. Finally, it is worth noting that the accuracy of the MLR reported here is much higher than in our previous conference paper.[29] This is because we have 100 features but only 98 training samples, and the number of features exceeds the number of samples making MLR severely overfitted and very sensitive to the samples. Our previous work only conducted the simulation once, which produced biased results, but bootstrapping in this work can provide a more reliable evaluation of the model.

To further illustrate the effectiveness of the proposed model, we also compare the method proposed in Richardson's work,[41] which also uses partial charging curves for capacity estimation. To be specific, they discretized the charging curve at the same interval in a specific voltage range and identified the values of time at equispaced voltage points to time vectors, which were used as features for capacity estimation. Subsequently, they proposed to use GPR with a Matérn (5/2) kernel function to map the time vector and capacity and achieved a high estimation accuracy. In this study, we use the same voltage range as PLSR to ensure that the results are comparable, and Table 1 shows the performance of GPR on the NASA dataset. It can be observed that GPR demonstrates the best capacity estimation result on cell #5 with an RMSE-Q of 0.50% and 95% CI of [0.32%, 0.86%], but the accuracy on cell #7 is worse than PLSR, and its performance on cell #18 is the worst of all models with an RMSE-Q of 5.09%, indicating that the complex structure of GPR makes it prone to overfitting.

Table 2: Comparison of RMSE-Q (%) of different models

Model	Cell #5		Cell #7		Cell #18	
	Mean	95% CI	Mean	95% CI	Mean	95% CI
MLR	1.07	[0.68, 1.69]	1.40	[1.04, 1.98]	1.94	[1.48, 2.90]
MLR-PCA	0.71	[0.60, 0.82]	1.47	[1.39, 1.56]	2.20	[1.91, 2.64]
MLR-KPCA	0.75	[0.68, 0.85]	1.48	[1.34, 1.61]	3.25	[3.01, 3.42]
SVR	0.71	[0.57, 0.89]	1.41	[1.24, 1.67]	2.25	[2.04, 2.49]
SVR-PCA	1.59	[1.33, 2.07]	2.38	[2.13, 2.72]	6.83	[6.46, 7.17]
SVR-KPCA	0.75	[0.66, 0.89]	1.57	[1.44, 1.75]	3.38	[3.20, 3.62]
RFR	0.82	[0.64, 1.02]	1.82	[1.68, 1.98]	4.56	[3.87, 5.37]
RFR-PCA	1.92	[1.49, 2.52]	2.72	[2.39, 3.14]	4.35	[3.16, 5.76]
RFR-KPCA	1.70	[1.22, 2.38]	2.40	[2.07, 2.89]	3.90	[3.46, 4.48]
LSTM network	1.04	[0.59, 3.23]	1.85	[1.34, 4.24]	4.82	[2.84, 11.21]
LSTM network-PCA	0.60	[0.44, 0.88]	1.28	[1.12, 1.46]	3.40	[2.79, 4.10]
LSTM network-KPCA	1.17	[0.97, 1.43]	2.02	[1.83, 2.25]	5.52	[5.08, 6.03]
PLSR	**0.59**	[0.51, 0.69]	**1.16**	[1.00, 1.36]	**1.66**	[1.38, 2.37]
GPR[41]	**0.50**	[0.32, 0.86]	1.75	[1.27, 2.38]	5.09	[2.54, 7.82]

Figure 6 presents a comprehensive comparison between measured and estimated capacity values, including the 95% CI, across various testing samples, as well as the errors as a function of cycle number. It is noteworthy that the estimated values are obtained by averaging the predictions of 3000 bootstrap models. In Figure 6 (a), the testing results for cell #5 are depicted, where the proposed PLSR model appears highly accurate in estimating capacity, evidenced by the alignment of most measured capacities within the 95% CI of the predictions. Figure 6 (b) illustrates how errors fluctuate with cycle numbers, showing all errors remain within 1%. Figure 6 (c) and (d) present testing results for cell #7, revealing some limitations in the model's performance on capacity estimation in the early cycles due to differing cycling characteristics compared to cell #5. Despite this, the PLSR model effectively tracks capacity changes over the battery's lifespan, with most estimated values closely aligning with actual measurements. This is further supported in Figure 6 (d), which shows the model's enhanced capacity estimation accuracy in mid-cycles compared to early cycles. Additionally, as observed in Figure 6 (c), many observations fall outside the 95% CI. The results are still meaningful, as our primary objective with the 95% CI is to demonstrate the robustness and uncertainty in the model's predictions. If we have a very wide CI, we may be lucky to get a prediction that is very close to the observed value, but in many cases, the predictions may all be poor and unreliable, depending on the data samples used for training. Conversely, a narrow and close-to-truth interval like the one in Figure 6 (c) implies that even if observations are not within the 95% CI, the predictions from any individual bootstrap model are likely to be acceptable, regardless of the specific training samples chosen. This aspect enhances the model's practicality for real-world applications. Lastly, Figure 6 (e) displays testing results for cell #18, indicating accurate capacity estimations for most samples. Compared to the other two cells, the widest 95% CI for cell #18 suggests that it has the largest variance in the predictions, implying notable differences in data distribution compared to cell #5, which is aligned with the subpar performance of other benchmark models on cell #18. Figure 6 (f) emphasizes PLSR's competency, maintaining a capacity error below 2% across most cycles. However, it performs poorly when the capacity suddenly increases, with the maximum error even reaching close to 8%, indicating the impact of abnormal fluctuations in the data on model accuracy.

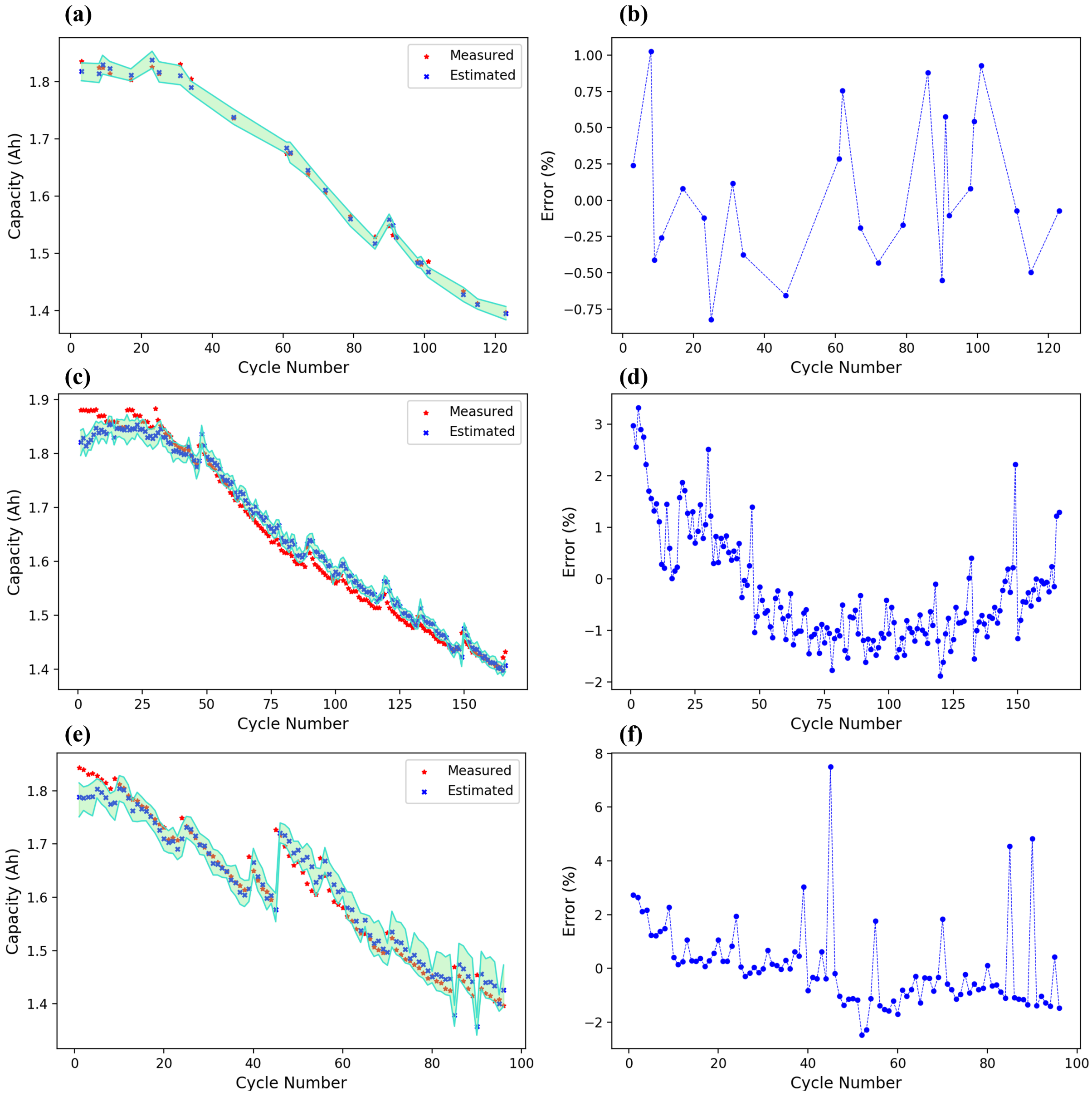


Figure 6: Testing results of PLSR with 4 components on capacity estimation: estimated capacity (a) and error (b) for cell #5, estimated capacity (c) and error (d) for cell #7, estimated capacity (e) and error (f) for cell #18.

4.4 Performance of RUL Prediction

Similarly to capacity estimation, the effectiveness of PLSR on RUL prediction is benchmarked with MLR, SVR, RFR, and the LSTM network, and PCA and KPCA are used to reduce the correlation among the features. We maintain the same data split for training and testing as in the capacity estimation to ensure consistency of the results. The input features to models are still the 100 IC values over a voltage range of 3.8V to 4.0V, but the output is the corresponding RUL. Here, RUL is calculated by subtracting the current cycle number from the battery's cycle life. For instance, the cycle life of cell #5 is 123 cycles, so the RUL at the 10th cycle is 113 cycles.

We first conduct a 5-fold CV for MLR on the training dataset to select the number of principal components for both PCA and KPCA in the task of RUL prediction, and results are displayed in Table 1. Similar to RMSE-Q, RMSE-RUL noticeably decreases at first as the number of principal components increases, and then slightly ascends afterward. The optimal results for both PCA and KPCA are achieved with 6 principal components. Therefore, we retain 6 principal components for the subsequent discussions.

Table 3 compares the performance of different models on RUL prediction. We can see that all models except SVR can accurately predict the RUL of testing samples from cell #5, in which PLSR has the best performance with an RMSE-RUL of 5.97 cycles and 95% CI of [4.77, 7.51] cycles, indicating that IC values are helpful for RUL prediction. Meanwhile, it can be noticed that PCA and KPCA significantly improve the RUL prediction performance of MLR and the LSTM network, but reduce the accuracy of SVR and RFR, which is consistent with our findings in capacity estimation. Notably, the LSTM network with PCA achieves an RMSE-RUL of 5.91 cycles, demonstrating a precision comparable to that of PLSR, with a broader 95% CI of [3.26, 9.37] cycles.

Further, all models are tested on the other two cells to evaluate their generalizability. As shown in Table 3, PLSR exhibited the best RUL prediction accuracy for cell #7 with an RMSE-RUL of 26.58 cycles, followed by the LSTM network with an RMSE-RUL of 27.83 cycles and MLR with an RMSE-RUL of 27.91 cycles, while SVR performs the worst with an RMSE-RUL of 42.74 cycles. However, an interesting finding is that SVR shows the best RUL prediction performance for cell #18 with an RMSE-RUL of 19.11 cycles. The inconsistent results suggest that SVR is not a stable model for RUL prediction. When PCA and KPCA are utilized for feature reduction, it is observed that PCA slightly enhances the prediction accuracy of MLR on cell #7, but reduces the accuracy for other models. For cell #18, both PCA and KPCA significantly improve the performance of the LSTM network, achieving an RMSE-RUL of 13.33 cycles and 16.08 cycles, respectively, outperforming PLSR with an RMSE-RUL of 21.06 cycles. However, considering the capacity estimation and RUL prediction for the three cells, PLSR emerges as the best model.

Table 3: Comparison of RMSE-RUL (cycle) of different models

Model	Cell #5		Cell #7		Cell #18	
	Mean	95% CI	Mean	95% CI	Mean	95% CI
MLR	9.05	[6.00, 12.97]	27.91	[23.45, 32.84]	20.63	[13.75, 30.20]
MLR-PCA	6.72	[5.86, 8.19]	27.08	[24.97, 28.96]	19.68	[15.49, 25.50]
MLR-KPCA	7.01	[6.18, 8.24]	28.52	[26.79, 30.15]	19.22	[17.53, 21.32]
SVR	23.22	[20.85, 24.62]	42.74	[38.67, 44.30]	**19.11**	[17.49, 21.26]
SVR-PCA	32.44	[31.29, 34.53]	48.30	[44.79, 53.18]	30.40	[27.03, 35.19]
SVR-KPCA	32.51	[31.80, 33.98]	48.61	[45.81, 52.47]	27.73	[24.81, 32.09]
RFR	6.11	[5.01, 8.13]	29.57	[28.08, 31.26]	26.85	[21.77, 31.64]
RFR-PCA	11.00	[8.31, 14.61]	33.39	[30.73, 36.52]	29.39	[19.65, 35.49]
RFR-KPCA	9.18	[6.44, 13.58]	31.17	[28.81, 34.06]	20.02	[17.99, 22.37]
LSTM network	6.48	[5.03, 8.63]	27.83	[25.83, 29.98]	25.91	[13.03, 36.65]
LSTM network-PCA	**5.91**	[3.26, 9.37]	29.85	[26.58, 33.47]	**13.33**	[10.93, 17.72]
LSTM network-KPCA	8.40	[6.58, 11.50]	29.84	[26.95, 32.68]	16.08	[13.57, 22.25]
PLSR	**5.97**	[4.77, 7.51]	**26.58**	[24.97, 28.22]	21.06	[15.26, 27.12]
EM[42]	34.70	[4.81, 71.35]	**15.60**	[0.53, 45.23]	61.14	[30.82, 97.63]

We also compare the proposed PLSR with the empirical model (EM) proposed in Wang's work.[42] EM is an empirical mathematical formulation obtained by fitting accelerated aging experimental data to describe the battery capacity fade, usually composed of simple function combinations such as polynomial functions and exponential functions. A general EM for capacity fade can be expressed as

$$Q_{\text{loss}} = \alpha \cdot x_{cyc}^{\beta} \tag{14}$$

where Q_{loss} is the capacity fade between the initial capacity and the current capacity,

x_{cyc} is the number of cycles the cell has undergone, α and β are coefficients that describe the rate of capacity loss by various factors such as current, temperature, etc. Here, since cell #5, #7, and #18 have the same charging/discharging current and cycling temperature, we assume that they have the same α and β.

To perform RUL prediction using EM, we first need to fit (14) with the training data and obtain α and β. Then, we can simulate the capacity fade curves of the three cells based on their initial capacities, which allows us to calculate the cycle life and RUL at a specific cycle. Table 3 shows the performance of EM in terms of RUL prediction. Although EM performs best for cell #7 with an RMSE-RUL of 15.60 cycles, it does not perform as well as the proposed PLSR for both cell #5 and cell #18. More importantly, 95% CI of EM is very wide for these three cells, indicating the instability of the empirical model in RUL prediction.

Figure 7 displays the comparison between the measured RUL and the predicted RUL with 95% CI for the testing samples, alongside error changes relative to cycle number. From Figure 7 (a) and (b), we can see that the proposed model can accurately predict the RUL for samples from cell #5 and all of the errors are within 15 cycles. Figure 7 (c) demonstrates that the predicted RUL for cell #7 from PLSR is less than the measured RUL, while Figure 7 (e) shows that the predicted RUL for cell #18 is greater than the measured RUL. This is due to the different cycling characteristics of the three cells, and we only use data from cell #5 to train the model. Our findings from Figure 7 (c) and (e) are consistent with the battery degradation patterns shown in Figure 1. Additionally, as illustrated in Figure 7 (d), the error significantly diminishes as the number of cycles increases, suggesting that cell #7 exhibits a similar degradation mode to cell #5 in the later cycles. The largest errors are observed in the early cycles, aligning with our findings presented in Figure 6 (d). Regarding cell #18, Figure 7 (f) depicts the error fluctuating as the cycle number changes. The most substantial prediction errors emerge between the 40th and 60th cycles, corresponding to the unexpected surge in capacity observed in the capacity curve.

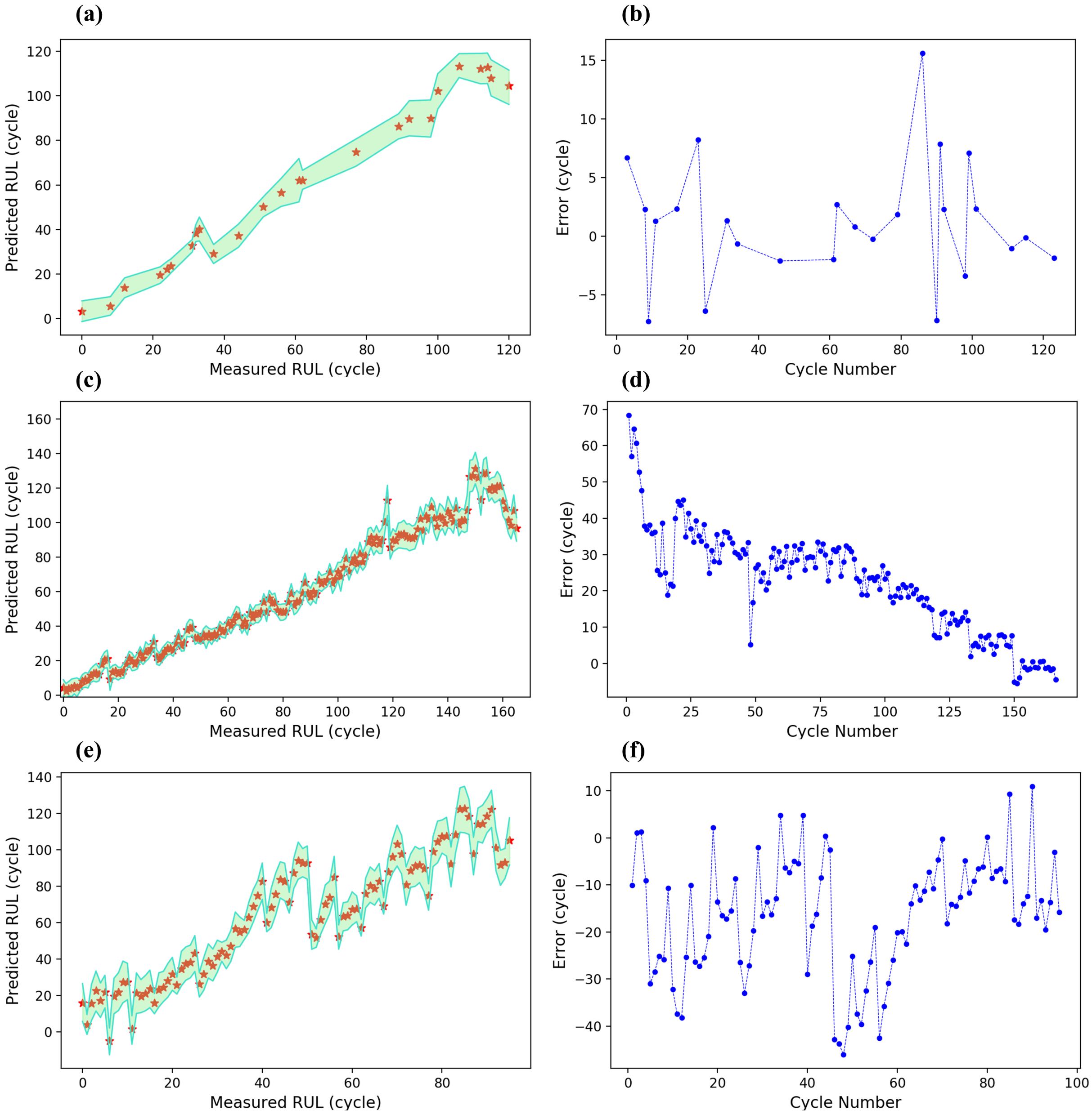


Figure 7: Testing results of PLSR with 4 components on RUL prediction: predicted RUL (a) and error (b) for cell #5, predicted RUL (c) and error (d) for cell #7, predicted RUL (e) and error (f) for cell #18.

5. Conclusions

This paper introduces a method using the PLSR model to perform battery capacity estimation and RUL prediction through partial IC curves. Firstly, an interpolation-based approach is developed to acquire the IC curve, allowing for the approximation of the IC values from the charging curves within a 3.8-4.0V voltage range. Subsequently, a PLSR model with 4 components is trained to learn the relationship between IC values and the outputs such as capacity and RUL. By using the bootstrapping technique, the proposed model can provide CIs for capacity estimation and RUL prediction, allowing us to evaluate the robustness and reliability of the model. A dataset of three battery cells (#5, #7, #18) from NASA is used to validate the proposed method. Experimental results show that the PLSR model with 4 components trained on 80% data samples of cell #5 can achieve an RMSE-Q of 0.59% and 95% CI of [0.51%, 0.69%] for capacity estimation and an RMSE-RUL of 5.97 cycles and 95% CI of [4.77, 7.51] cycles for RUL prediction for the remaining 20% data samples. Moreover, the trained model obtains an RMSE-Q of 1.16% and an RMSE-RUL of 26.58 cycles for data samples of cell #7 and an RMSE-Q of 1.66% and an RMSE-RUL of 21.06 cycles for data samples of cell #18 without changing the model weights. The success of this work has advanced the development of battery capacity estimation and RUL prediction using linear models.

Acknowledgment

Yankai Cao acknowledges funding from the discovery program of the Natural Science and Engineering Research Council of Canada under grant RGPIN-2019-05499. Bhushan Gopaluni would like to acknowledge the financial support from NSERC Discovery grant. We are thankful for the computing resources provided by SciNet (www.scinethpc.ca) and Compute Canada (www.computecanada.ca).

References

(1) Bresser, D.; Hosoi, K.; Howell, D.; Li, H.; Zeisel, H.; Amine, K.; Passerini, S. Perspectives of automotive battery R&D in China, Germany, Japan, and the USA. *Journal of*

Power Sources **2018**, *382*, 176–178.

(2) Han, X.; Lu, L.; Zheng, Y.; Feng, X.; Li, Z.; Li, J.; Ouyang, M. A review on the key issues of the lithium ion battery degradation among the whole life cycle. *eTransportation* **2019**, *1*, 100005.

(3) Zhu, J.; Wang, Y.; Huang, Y.; Bhushan Gopaluni, R.; Cao, Y.; Heere, M.; Mühlbauer, M. J.; Mereacre, L.; Dai, H.; Liu, X.; others Data-driven capacity estimation of commercial lithium-ion batteries from voltage relaxation. *Nature Communications* **2022**, *13*, 2261.

(4) Waag, W.; Fleischer, C.; Sauer, D. U. Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles. *Journal of Power Sources* **2014**, *258*, 321–339.

(5) Liaw, B. Y.; Nagasubramanian, G.; Jungst, R. G.; Doughty, D. H. Modeling of lithium ion cells—A simple equivalent-circuit model approach. *Solid State Ionics* **2004**, *175*, 835–839.

(6) Plett, G. L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation. *Journal of Power Sources* **2004**, *134*, 277–292.

(7) Zou, C.; Manzie, C.; Nešić, D.; Kallapur, A. G. Multi-time-scale observer design for state-of-charge and state-of-health of a lithium-ion battery. *Journal of Power Sources* **2016**, *335*, 121–130.

(8) Schwunk, S.; Armbruster, N.; Straub, S.; Kehl, J.; Vetter, M. Particle filter for state of charge and state of health estimation for lithium–iron phosphate batteries. *Journal of Power Sources* **2013**, *239*, 705–710.

(9) Plett, G. L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Modeling and identification. *Journal of Power Sources* **2004**, *134*, 262–276.

(10) Yang, D.; Zhang, X.; Pan, R.; Wang, Y.; Chen, Z. A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve. *Journal of Power Sources* **2018**, *384*, 387–395.

(11) Guo, P.; Cheng, Z.; Yang, L. A data-driven remaining capacity estimation approach for lithium-ion batteries based on charging health feature extraction. *Journal of Power Sources* **2019**, *412*, 442–450.

(12) Baghdadi, I.; Briat, O.; Gyan, P.; Vinassa, J. M. State of health assessment for lithium batteries based on voltage–time relaxation measure. *Electrochimica Acta* **2016**, *194*, 461–472.

(13) Pei, P.; Zhou, Q.; Wu, L.; Wu, Z.; Hua, J.; Fan, H. Capacity estimation for lithium-ion battery using experimental feature interval approach. *Energy* **2020**, *203*, 117778.

(14) Richardson, R. R.; Birkl, C. R.; Osborne, M. A.; Howey, D. A. Gaussian process regression for in situ capacity estimation of lithium-ion batteries. *IEEE Transactions on Industrial Informatics* **2018**, *15*, 127–138.

(15) Li, Y.; Zou, C.; Berecibar, M.; Nanini-Maury, E.; Chan, J. C.-W.; Van den Bossche, P.; Van Mierlo, J.; Omar, N. Random forest regression for online capacity estimation of lithium-ion batteries. *Applied Energy* **2018**, *232*, 197–210.

(16) Wang, D.; Miao, Q.; Pecht, M. Prognostics of lithium-ion batteries based on relevance vectors and a conditional three-parameter capacity degradation model. *Journal of Power Sources* **2013**, *239*, 253–264.

(17) Patil, M. A.; Tagade, P.; Hariharan, K. S.; Kolake, S. M.; Song, T.; Yeo, T.; Doo, S. A novel multistage Support Vector Machine based approach for Li ion battery remaining useful life estimation. *Applied Energy* **2015**, *159*, 285–297.

(18) Gao, D.; Huang, M. Prediction of remaining useful life of lithium-ion battery based on multi-kernel support vector machine with particle swarm optimization. *Journal of Power Electronics* **2017**, *17*, 1288–1297.

(19) Liu, D.; Pang, J.; Zhou, J.; Peng, Y.; Pecht, M. Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression. *Microelectronics Reliability* **2013**, *53*, 832–839.

(20) He, Y.-J.; Shen, J.-N.; Shen, J.-F.; Ma, Z.-F. State of health estimation of lithium-ion batteries: A multiscale Gaussian process regression modeling approach. *AIChE Journal* **2015**, *61*, 1589–1600.

(21) Richardson, R. R.; Osborne, M. A.; Howey, D. A. Gaussian process regression for forecasting battery state of health. *Journal of Power Sources* **2017**, *357*, 209–219.

(22) Zhang, Y.; Xiong, R.; He, H.; Pecht, M. G. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Transactions on Vehicular Technology* **2018**, *67*, 5695–5705.

(23) Li, X.; Zhang, L.; Wang, Z.; Dong, P. Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and Elman neural networks. *Journal of Energy Storage* **2019**, *21*, 510–518.

(24) Mao, L.; Xu, J.; Chen, J.; Zhao, J.; Wu, Y.; Yao, F. A LSTM-STW and GS-LM fusion method for lithium-ion battery RUL prediction based on EEMD. *Energies* **2020**, *13*, 2380.

(25) Severson, K. A.; Attia, P. M.; Jin, N.; Perkins, N.; Jiang, B.; Yang, Z.; Chen, M. H.; Aykol, M.; Herring, P. K.; Fraggedakis, D.; others Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy* **2019**, *4*, 383–391.

(26) Zhang, Y.; Peng, Z.; Guan, Y.; Wu, L. Prognostics of battery cycle life in the early-cycle stage based on hybrid model. *Energy* **2021**, *221*, 119901.

(27) Zhang, Y.; Tang, Q.; Zhang, Y.; Wang, J.; Stimming, U.; Lee, A. A. Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using machine learning. *Nature Communications* **2020**, *11*, 1706.

(28) Saha, B.; Goebel, K. Battery data set. *NASA AMES prognostics data repository* **2007**,

(29) Wang, Y.; Zhu, J.; Cao, L.; Gopaluni, B.; Cao, Y. Online Capacity Estimation of Lithium-ion Batteries by Partial Incremental Capacity Curve. 2022 IEEE Vehicle Power and Propulsion Conference (VPPC). 2022; pp 1–6.

(30) Li, Y.; Abdel-Monem, M.; Gopalakrishnan, R.; Berecibar, M.; Nanini-Maury, E.; Omar, N.; van den Bossche, P.; Van Mierlo, J. A quick on-line state of health estimation method for Li-ion battery with incremental capacity curves processed by Gaussian filter. *Journal of Power Sources* **2018**, *373*, 40–53.

(31) Weng, C.; Cui, Y.; Sun, J.; Peng, H. On-board state of health monitoring of lithium-ion batteries using incremental capacity analysis with support vector regression. *Journal of Power Sources* **2013**, *235*, 36–44.

(32) Wold, S.; Sjöström, M.; Eriksson, L. PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems* **2001**, *58*, 109–130.

(33) Arlot, S.; Celisse, A. A survey of cross-validation procedures for model selection. *Statistics Surveys* **2010**, *4*, 40–79.

(34) Efron, B. *Breakthroughs in Statistics: Methodology and Distribution*; Springer: Verlag New York, 1992; Vol. II; pp 569–593.

(35) Montgomery, D. C.; Peck, E. A.; Vining, G. G. *Introduction to Linear Regression Analysis*; John Wiley & Sons: Hoboken, New Jersey, 2021; pp 67–128.

(36) Drucker, H.; Burges, C. J.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. *Advances in Neural Information Processing Systems* **1996**, *9*.

(37) Breiman, L. Random forests. *Machine learning* **2001**, *45*, 5–32.

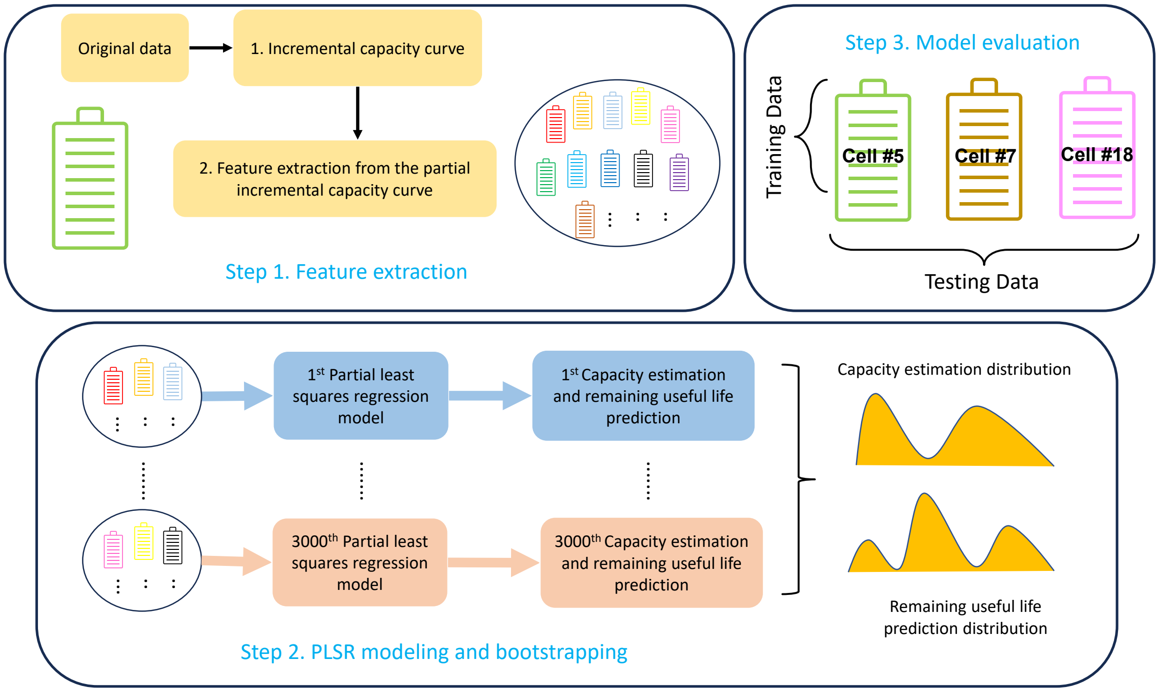
(38) Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Computation* **1997**, *9*, 1735–1780.

(39) Bro, R.; Smilde, A. K. Principal component analysis. *Analytical Methods* **2014**, *6*, 2812–2831.

(40) Schölkopf, B.; Smola, A.; Müller, K.-R. Kernel principal component analysis. International Conference on Artificial Neural Networks. 1997; pp 583–588.

(41) Richardson, R. R.; Birkl, C. R.; Osborne, M. A.; Howey, D. A. Gaussian process regression for in situ capacity estimation of lithium-ion batteries. *IEEE Transactions on Industrial Informatics* **2018**, *15*, 127–138.

(42) Wang, J.; Liu, P.; Hicks-Garner, J.; Sherman, E.; Soukiazian, S.; Verbrugge, M.; Tataria, H.; Musser, J.; Finamore, P. Cycle-life model for graphite-LiFePO4 cells. *Journal of Power Sources* **2011**, *196*, 3942–3948.


Original data
1. Incremental capacity curve
2. Feature extraction from the partial incremental capacity curve
Step 1. Feature extraction
Step 3. Model evaluation
Training Data
Cell #5
Cell #7
Cell #18
Testing Data
1st Partial least squares regression model
1st Capacity estimation and remaining useful life prediction
3000th Partial least squares regression model
3000th Capacity estimation and remaining useful life prediction
Capacity estimation distribution
Remaining useful life prediction distribution
Step 2. PLSR modeling and bootstrapping