Journal of Energy Storage 52 (2022) 104936

Contents lists available at ScienceDirect

Journal of Energy Storage

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

Research papers

Early-stage lifetime prediction for lithium-ion batteries: A deep learning framework jointly considering machine-learned and handcrafted data features

Zicheng Fei [a], Zijun Zhang [a,*], Fangfang Yang [b], Kwok-Leung Tsui [c], Lishuai Li [a]

[a] School of Data Science, City University of Hong Kong, Hong Kong, China
[b] School of Intelligent Systems Engineering, Sun Yat-sen University, China
[c] Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, United States

ARTICLE INFO

Keywords:
Lithium-ion battery
Lifetime prediction
Early stage
Health feature extraction
Convolutional neural networks
Data-driven models

ABSTRACT

Predicting the battery lifetime at its early stage is a promising technology for accelerating the battery development, production, and design optimization. However, it is a challenging task for most existing prediction methods because information is too limited in early life cycles, and the early-cycle capacity data exhibits a weak correlation with the target battery lifetime. In this paper, to realize an accurate battery lifetime prediction via data obtained from just first few life cycles, we propose a three-stage deep learning framework. First, we develop an emerging two-channel data feature engineering process, which jointly consider a convolutional neural network based latent feature extraction and domain knowledge based handcrafted features. Next, a wrapper feature selection method is adopted to further compress the dimension of developed features via eliminating linearly correlated ones. Finally, processed data features are fed into a data-driven model to realize the early-stage battery lifetime prediction. Results of computational experiments show that our proposed joint consideration of machine-learned features and handcrafted features can improve early-stage battery lifetime predictions via comparing with state-of-the-art benchmarks. We also show that the proposed framework can be generalized to batteries cycled under different operation conditions.

1. Introduction

The early-stage battery lifetime prediction method is meaningful to accelerate the battery development, production, and design optimization [1]. For instance, using only data of the first 100 cycles, if developers can rapidly predict the lifetime of a lithium-ion battery (LIB) of 2500 cycles, the remaining 2400 testing cycles will be saved, and the development process will be significantly accelerated by 96%. In the emerging multi-step fast-charging field, the early-stage battery lifetime prediction can help quickly verify the effect of various fast-charging protocols, figure out the optimal charge policy, and subsequently expedite the battery optimization process. During the production process, the early-stage battery lifetime prediction can also facilitate manufacturers to rapidly verify the quality of manufacturing processes via a few diagnostic cycles, and detect the abnormal battery product in advance. Therefore, it is meaningful to develop methods for accurate early-stage battery lifetime predictions.

Battery lifetime prediction methods reported in existing literature can be generally divided into two groups, model-based methods and data-driven methods. Model-based methods develop mathematical models that incorporates battery operating conditions and physico-chemical mechanisms to depict the battery ageing pattern. Next, they constantly update model parameters through advanced filtering algorithms, e.g., the Kalman filter [2] and particle filter [3], to perform battery remaining useful life (RUL) predictions. Yang et al. [4] studied the correlation between the coulombic efficiency (CE) and the loss of lithium inventory, developed a CE-based model to depict degradation patterns of LIBs, as well as applied the particle filter algorithm to predict the battery RUL. Ma et al. [5] used an exponential model to characterize battery degradation trends and proposed a Gauss-Hermite particle filter to online update model parameters and predict the battery RUL. Micea et al. [6] utilized a second-order polynomial model to fit the battery ageing trend and predict the battery RUL. Xing et al. [7] presented an ensemble model, which fused an exponential model and a polynomial

* Corresponding author.
E-mail address: zijzhang@cityu.edu.hk (Z. Zhang).

https://doi.org/10.1016/j.est.2022.104936
Received 20 February 2022; Received in revised form 22 April 2022; Accepted 18 May 2022
Available online 2 June 2022

Table 1
Summary of data-driven studies on battery lifetime prediction.

Data-driven studies	Feature type		Model type		Task type	
	Handcrafted features	Deeply learned features	Classical machine learning models	Deep learning models	SOH estimation/RUL prediction	Early-stage lifetime prediction
[8–12,14,17–20]	✓		✓		✓	
[13,15,16,27]	✓			✓	✓	
[21]		✓	✓		✓	
[22]		✓		✓	✓	
[26]	✓			✓	✓	
[1,23,25]	✓		✓			✓
[24]	✓			✓		✓
This work		✓	✓	[a]		✓

[a] Deep learning models are compared in the prediction model development.

model, to characterize battery ageing trends, and next applied the particle filter to online predict the battery RUL. Model-based methods have shown successes on battery lifetime prediction; yet, they still have following limitations: 1) The underlying battery degradation model would significantly influence the lifetime prediction performance; 2) It is hard for model-based methods to make accurate predictions at the early ageing stage because of the inappreciable capacity degradation signs in early cycles [1].

In contrast, data-driven methods do not employ explicit mathematical forms to capture the complicated degradation pattern of LIBs. Data-driven models regarded as black-boxes output the battery lifetime prediction by feeding extracted battery health features into different data-driven models. Features considered in existing data-driven battery prognostic studies can be grouped into five types: 1) Features derived from raw voltage/current/temperature-time curves, such as the constant current (CC) and constant voltage (CV) charging duration [8], the increase of charge voltage within a fixed time interval [9], the slope of the end of CC charging curves [10], the change rate of the surface temperature [11], etc.; 2) Features derived from the incremental capacity (IC) and differential voltage (DV) analysis. For example, positions of IC peaks and valleys [12], the height, width, and area of raw IC peaks [13] and modified IC peaks [14], the difference of selected IC values [15]; 3) Directly measured variables regarded as features, such as the capacity [16], measurements from the electrochemical impedance spectroscopy [17,18]; 4) Features based on statistical metrics, such as the sample entropy [19], fuzzy entropy [13], and Shannon entropy [20] of a voltage sequence; as well as 5) Features extracted based on deep learning models, such as the convolutional neural network (CNN) [21,22] and recurrent neural network [22]. Recently, a few state-of-the-art studies also combine features of multiple types to achieve more accurate predictions. For example, in [1] [23] [24] [25,26], by analyzing the evolution of the battery voltage curve, temperature, and internal resistance profiles, a set of features covering multiple feature types were extracted and verified. In data-driven studies, these handcrafted and deeply learned features were next fed into different data-driven models, such as the elastic net [1], support vector machine (SVM), Gaussian process regression (GPR) [23], stacked denoising autoencoder [24], gradient boosting regression tree (GBRT) [25], random forest [21], deep neural networks [22,26], etc., to perform direct battery lifetime predictions or to extrapolate the battery lifetime through the SOH estimation. Table 1 summarizes related data-driven works in the literature, in terms of different feature types, model types and target tasks, as well as compares this work with existing related works from these aspects.

Predicting the battery lifetime at its early stage with limited early-cycle data is of great importance to the fast performance evaluation during the battery development and optimization. However, the early-stage battery lifetime prediction is a challenging task for most existing RUL prediction methods. Existing studies generally require 40%–70% historical data along the entire battery degradation trajectory to estimate model parameters or train data-driven models [28]. In early life cycles, the capacity degradation of LIB is negligible, indicating a weak correlation between the battery lifetime and the early-cycle capacity data. Therefore, to realize an accurate early-stage lifetime prediction via battery data of just first few cycles, effective lifetime-correlated features and advanced prediction techniques need to be explored.

There are so far two main categories of feature extraction techniques: 1) traditional domain knowledge based handcrafted features; and 2) recent deep learning based automatic feature extraction. As shown in Table 1, existing studies of battery lifetime predictions mostly consider the first category, which manually design features based on profiles of the battery voltage, capacity, temperature, etc. Handcrafted features have proven their values on battery lifetime predictions. Yet, such practice requires deep domain knowledge and significant efforts on a careful feature design, which can be a time-consuming trial-and-error process. Recent advances in deep learning have introduced powerful tools for automatic and in-depth feature extractions, such as the CNN. Apart from the success in image fields, the CNN has also been extended for extracting features from numerical data in many studies, such as the aero-engine RUL prediction [29], battery state-of-charge (SOC) estimation [30] etc. However, the adaption of CNN-based feature engineering into the early-stage battery lifetime prediction task has been seldom studied. The comparison and integration of handcrafted features and machine learned features into this task has not been discussed yet.

In this paper, a novel three-stage data-driven framework is developed for achieving more accurate and robust early-cycle LIB lifetime predictions. At the first stage, a two-channel data feature engineering process, which integrates latent features automatically extracted by a CNN and domain knowledge-based handcrafted features, is proposed. A tensor structure integrating the first-100-cycle raw voltage, current, and temperature data during battery charge-discharge processes is developed as the input of the CNN. A modified CNN architecture is applied to extract latent features via capturing local spatial and temporal patterns from the tensor input. At the second stage, a wrapper feature selection process is adopted to compress the feature dimension by eliminating features with high collinearities. At the third stage, selected meaningful features are fed into a data-driven model to finally produce the early-stage battery lifetime prediction. In computational studies, three datasets covering different battery materials and operation conditions are considered. Five sets of experiments are conducted to verify the value the proposed framework. First, the effectiveness of each stage of the framework is discussed. Next, the robustness of the framework is verified. Finally, the superiority of the framework is validated by comparing with a set of state-of-the-art benchmarks.

Main contributions of this study are summarized as follows:

1) **Data-driven framework**: A three-stage data-driven framework is proposed for accurately predicting LIB lifetime via battery data collected from only first 100 cycles, which is a pioneering study in the field.
2) **Data feature engineering process**: A novel feature engineering process effectively integrating CNN extracted latent features and handcrafted features is developed.

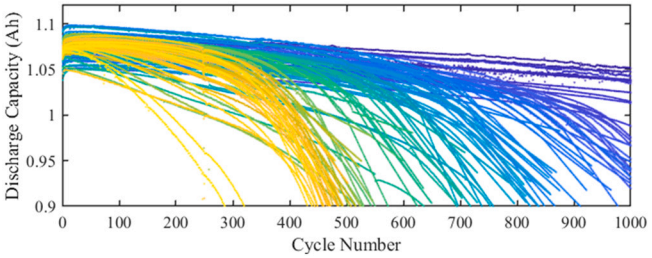

Fig. 1. Capacity fade curves of cell samples in Dataset A in first 1000 cycles.

Table 2
Operation conditions of selected cells from dataset A [1].

Barcode	Charge Condition	Lifetime
EL150800460514	'3.6C (80%) - 3.6C'	1852
EL150800460468	'5.4C (40%) - 3.6C'	1054
EL150800460525	'7C (30%) - 3.6C'	742
EL150800463229	'8C (15%) - 3.6C'	966
EL150800460518	'1C (4%) - 6C'	300
EL150800460501	'4.65C (69%) - 6C'	527
EL150800737276	'5.6C (19%) - 4.6C'	817
EL150800737334	'5C (67%) - 4C'	1935

3) **Battery data usage paradigm**: A novel tensor-like data structure is constructed as the input for the modified CNN architecture, which integrates three parameters, the voltage, current, and temperature, during battery charge-discharge processes.
4) **Prediction improvement**: Experimental results verify that our proposal of joining CNN features and handcrafted features could bring more accurate early-stage battery lifetime prediction than existing state-of-the-art benchmarks. And the proposed framework presents a decent generalization ability to different LIB materials and operation conditions.

2. Data description

In this study, three datasets considering LIBs of different materials and operated under different conditions are studied.

1) *Dataset A*: This dataset [1] provided by Massachusetts Institute of Technology is so far the largest dataset for the long-term degradation study of commercial LIBs. It contains 124 lithium iron phosphate (LFP) cell samples cycled under 72 various operation conditions. Fig. 1 demonstrates the capacity degradation curves of all samples in first 1000 cycles. As depicted in Fig. 1, degradation trajectories of these cells differ greatly, and their lifetimes range from 300 to 2300 cycles. The lifetime is defined as the cycle number when the capacity fades to 80% of the nominal capacity 1.1 Ah.

In a 30 °C temperature chamber, all cells were first charged with a current of C1 rate until the SOC reached S1 (≤80%), and next charged with C2 rate until the SOC reached 80%. This process is denoted as "C1 (*S*1)-C2". Next, the cells were charged from 80% to 100% SOC with a 1C constant current/constant voltage (CCCV) charging mode until the voltage reached 3.6 V. Subsequently, the cells were identically discharged with 4C rate until the voltage dropped to 2.0 V. Detailed charge conditions and lifetimes of eight selected cells are listed in Table 2. During cycling, various parameters (e.g., voltage, current, temperature, IR, etc.) were continuously measured and recorded in both short-term time dimension (within a single cycle) and long-term cycle dimension (along multiple cycles).

2) *Dataset B*: The second dataset is the CS2 dataset provided by The University of Maryland [2]. It contains prismatic lithium cobalt oxide (LiCoO2)/graphite cells with a nominal capacity of 1.1 Ah. In this paper, two cell samples, CS2_33 and CS2_34, are studied and labeled as cell B1 and B2. At the room temperature, the cells first underwent a 0.5C constant current (CC) charging process until the voltage reached 4.2 V, and next went through a 4.2 V constant voltage (CV) charging process until the current dropped to 0.05A. Subsequently, the cells underwent a 0.5C CC discharge process until the voltage decreased to 2.7 V. During cycling, the voltage and current parameters were continuously measured and recorded.

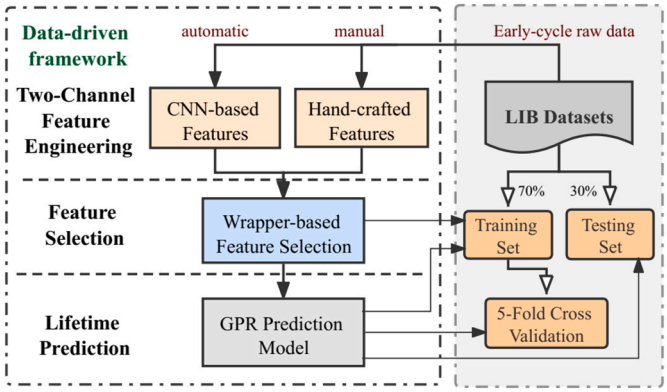

Fig. 2. The three-stage deep learning based framework for the early-stage battery lifetime prediction.

3) *Dataset C*: This dataset is obtained from our own laboratory using Arbin BT2000 battery tester. It contains two LFP/graphite cell samples with a nominal capacity of 1.2 Ah. In this study, the two samples are labeled as cell C1 and C2. In a 45 °C temperature chamber, the cells successively went through a 1C-rate CCCV charging process and a 1C-rate CC discharging process. The upper and lower cutoff voltages are 3.6 V and 2.0 V, respectively. During cycling, the voltage and current signals were continuously recorded.

3. A data-driven framework for early-stage battery lifetime prediction

In this section, a three-stage framework is proposed for predicting the LIB lifetime at its early stage. The overall framework is illustrated in Fig. 2, which is composed of the two-channel feature engineering stage, feature selection stage, and lifetime prediction stage.

3.1. Stage 1: two-channel feature engineering

In the first stage, raw battery signals including the charge and discharge voltage (V), current (I), temperature (T), capacity, and internal resistance obtained from the first 100 life cycles are utilized to extract effective lifetime-correlated features.

3.1.1. CNN-based automatic feature extraction

To reveal the degradation of LIBs in early life cycles, we plot the first-100-cycle full charge and discharge V/I/T data of a cell sample from Dataset A in Fig. 3. As shown in Fig. 3, the early-cycle profiles of V/I/T exhibit certain spatial variation tendency as the battery degrades. For example, the whole curve of T (Fig. 3(c)) moves towards the upper left as the cycle increases. Physically speaking, as the battery degrades, it would be faster to reach a higher surface temperature during cycling. This indicates that the evolution of V/I/T data in early cycles can provide effective spatial information for the battery degradation diagnosis and lifetime prediction [23].

Based on these patterns, we utilize the first-100-cycle charge and discharge data of *V/I/T* to construct a tensor-like data structure as the input for CNN as shown in Fig. 4. For the *i*-th battery sample, the input tensor is $\boldsymbol{x}_i = (V_i, I_i, T_i) \in R^{100\times100\times3}$. Data of *V/I/T* are stored in channel 1/2/3 of the tensor, respectively. In each channel, the horizontal dimension corresponds to the cycle number ranging from cycle 1–100, while the vertical dimension corresponds to 100 equal-time-interval

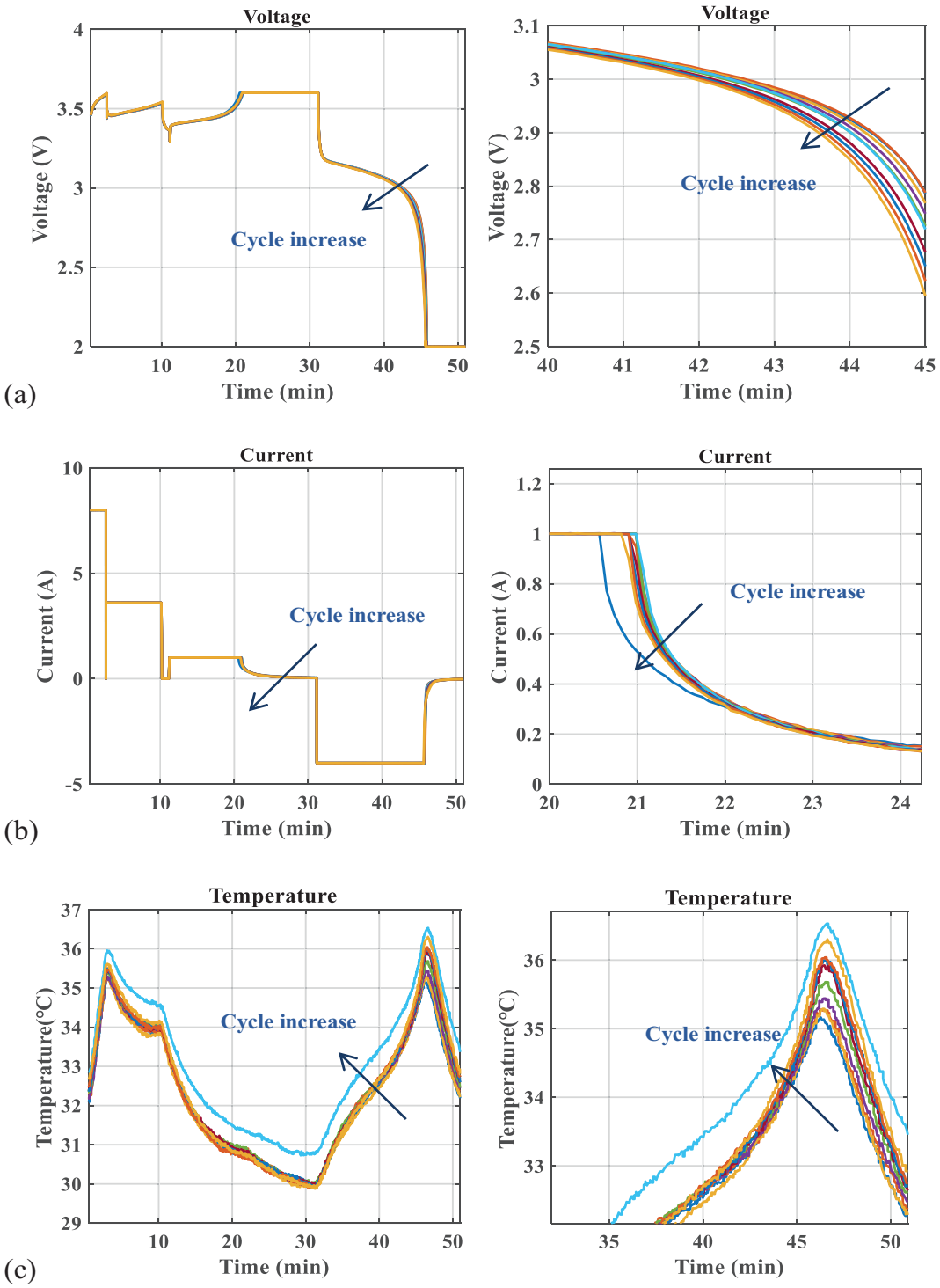

Fig. 3. Cycle-to-cycle evolution of (a) voltage (V); (b) current (I); (c) temperature (T) in first 100 cycles. Figures on the left show the full curve, while figures on the right show corresponding zoomed-in curves.

sampled values of the raw $V/I/T$ data in a full charge and discharge cycle. Given a training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$, y_i is the lifetime of the i-th battery sample, the input of CNN is $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_m] \in R^{m\times 100\times 100\times 3}$.

Training an effective CNN structure from the scratch is computationally expensive [31]. Recent studies have shown that feature representations learned by existing CNNs well trained on large scale datasets can be effectively transferred to other tasks [32]. Therefore, to evaluate whether a pre-trained CNN can perform well on our task, in this paper, we adopt the structure and weights of a famous pre-trained CNN model, VGG16, which has already learned rich feature representations from the ImageNet dataset. Based on a lot of experimental analysis, we discover that latent feature representations learned from the first seven layers of VGG16 are most predictive for the battery lifetime. Therefore, we adopt this CNN architecture and add a global average pooling layer at the end to form our own CNN feature extractor, denoted as a function $\varphi(\cdot)$.

As shown in Fig. 5, the CNN feature extractor $\varphi(\cdot)$ contains five convolutional layers, two max pooling layers, and one global average pooling layer, which are stacked as {*conv, conv, max-pooling, conv, conv,*

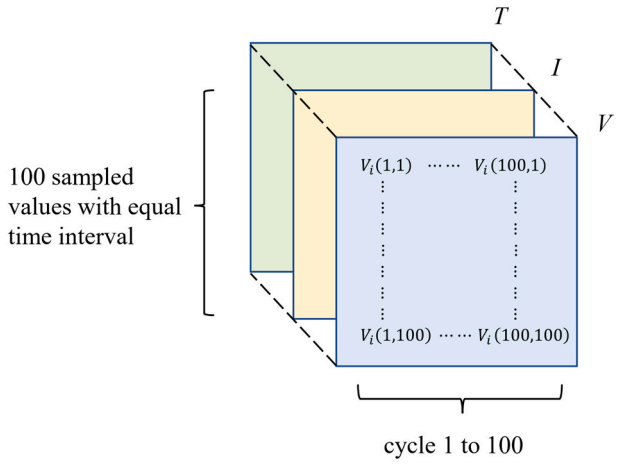

Fig. 4. Input tensor structure for CNN.

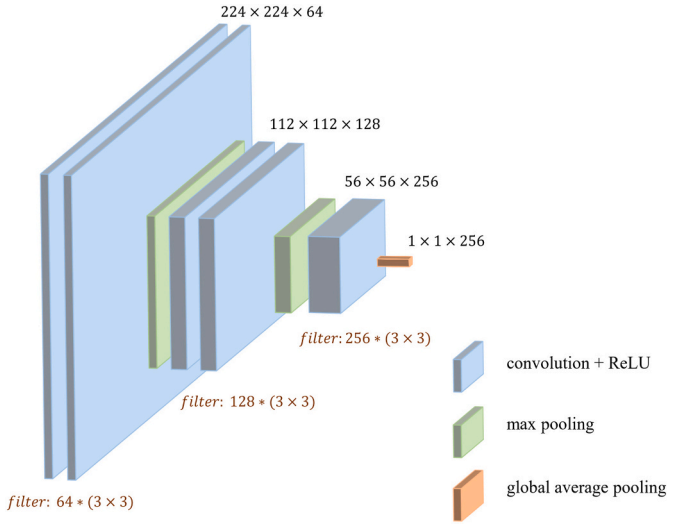

Fig. 5. Structure of the CNN feature extractor.

max-pooling, conv, global average pooling}. To match the fixed input size 224 × 224 × 3 of $\varphi(\cdot)$, we pad $X \in R^{m\times 100\times 100\times 3}$ with zero and obtain $\widehat{X} = [\widehat{x}_1, \widehat{x}_2, \ldots, \widehat{x}_i, \ldots, \widehat{x}_m] \in R^{m\times 224\times 224\times 3}$. For each input $\widehat{x}_i$, the generated feature map s^r of the r-th convolutional layer is:

$$s^r = A\left(W_q^r \odot \widehat{x}_i^{\,r-1} + B_q^r\right), \quad r = 1, 2, \ldots, 5 \tag{1}$$

$$A(s) = \begin{cases} s, & if\ s > 0 \\ 0, & if\ s \le 0 \end{cases} \tag{2}$$

where $\boldsymbol{W}_q^r$ and B_q^r are weights and bias of the q-th kernel in the r-th convolutional layer, and $\widehat{x}_i^{\,r-1}$ is the feature map generated from the previous convolutional layer. The $A(\bullet)$ is a nonlinear ReLU activation function in each convolutional layer. Max pooling layers with a window size of 2 × 2 are utilized to compress the spatial feature size. At the end, the global average pooling layer pools the feature map over all spatial locations.

Through these convolution and pooling operations as well as nonlinear transformations, correlations of battery *V/I/T* signals over different early life cycles and over different periods within a cycle as well as correlations among *V/I/T* over the tensor depth can be effectively captured. The latent features generated from the CNN channel is $z_c = \varphi(\widehat{X}) \in R^{m\times d_1}, d_1 = 256$.

Table 3
Hand-crafted Features for the battery lifetime prediction.

Feature category	No.	Feature
Charge-related features	F1	Time duration of CV charging period, difference between cycle 10 and 100
	F2	Charging time, difference between cycle 10 and 100
	F3	Average charge time from cycle 10 to 100
Discharge-related features	F4	Covering area of IC peak, difference between cycle 10 and 100
	F5	Variance of $\Delta Q_{100-10}(V)$
	F6	Minimum of $\Delta Q_{100-10}(V)$
	F7	Mean of $\Delta Q_{100-10}(V)$
	F8	Maximum of $\Delta Q_{100-10}(V)$
	F9	Skewness of $\Delta Q_{100-10}(V)$
	F10	Kurtosis of $\Delta Q_{100-10}(V)$
Capacity-related features	F11	p_1 of the linear model, fitted to the cycle 80–100 capacity degradation curve
	F12	p_2 of the linear model, fitted to the cycle 80–100 capacity degradation curve
	F13	p_3 of the SRT model, fitted to the cycle 80–100 capacity degradation curve
	F14	p_4 of the SRT model, fitted to the cycle 80–100 capacity degradation curve
Temperature-related features	F15	Temperature peak during discharge process, difference between cycle 2 and 100
	F16	Time of temperature peak during discharge process, difference between cycle 2 and 100
	F17	Minimum temperature within a cycle, difference between cycle 2 and 100
	F18	Average temperature within a cycle, difference between cycle 2 and 100
IR-related features	F19	Internal resistance, difference between cycle 2 and 100
	F20	Average internal resistance from cycle 2 to 100

Note: CV = constant voltage; IC = incremental capacity; SRT = square root of time.

3.1.2. Domain knowledge based handcrafted features

In this study, we manually crafted 20 features $z_H \in R^{m\times d_2}$, $d_2 = 20$, based on battery degradation data of first 100 cycles to perform early-stage lifetime predictions. The z_H is designed to characterize battery degradation patterns from various aspects and are grouped into five categories, as listed in Table 3. The crafted features are inspired by [1,23,25].

As discussed before, evolution curves of some measured battery parameters show certain spatial variation trend as the battery ages. Fig. 6 demonstrates the evolution of (a) CCCV charging curves; (b) IC curves; (c) Discharge capacity versus voltage ($Q(V)$) curves; and (d) IR curves of a battery sample at multiple cycles. We observe that as the cycle number increases, these curves tend to move towards certain directions as annotated in Fig. 6. For instance, the CCCV charging curve shifts towards the upper left as the battery ages because the battery would more quickly reach the charging cutoff voltage and therefore the duration of CV charging mode would become longer. Such variation trends potentially reflect battery ageing dynamics and could help predict the battery lifetime at an early stage. Based on these patterns, the 20 features are manually designed.

Details of some features are elaborated as below:

(a) In features F5 ~ F10, $Q_i(V)$ means the discharge capacity versus voltage at cycle i, and $\Delta Q_{i-j}(V)$ denotes the transformation between cycle i and j, i.e., $Q_i(V) - Q_j(V)$. F5 ~ F10 are summary statistics, i.e., the variance, mean, minimum, maximum, kurtosis, and skewness, of $\Delta Q_{100-10}(V)$, denoted as $Q_{100}(V) - Q_{10}(V)$. Detailed expressions of these statistics have been offered in [1].

(b) F11 and F12 are two parameters, p_1 and p_2, of a linear model fitting the capacity degradation curve as described in Eq. (3). F13 and F14 are two parameters, p_3 and p_4, of a typical semi-empirical model, the square root of time (SRT) model [33], as described in Eq. (4).

$$C_l^{linear} = p_1 \cdot l + p_2 \tag{3}$$

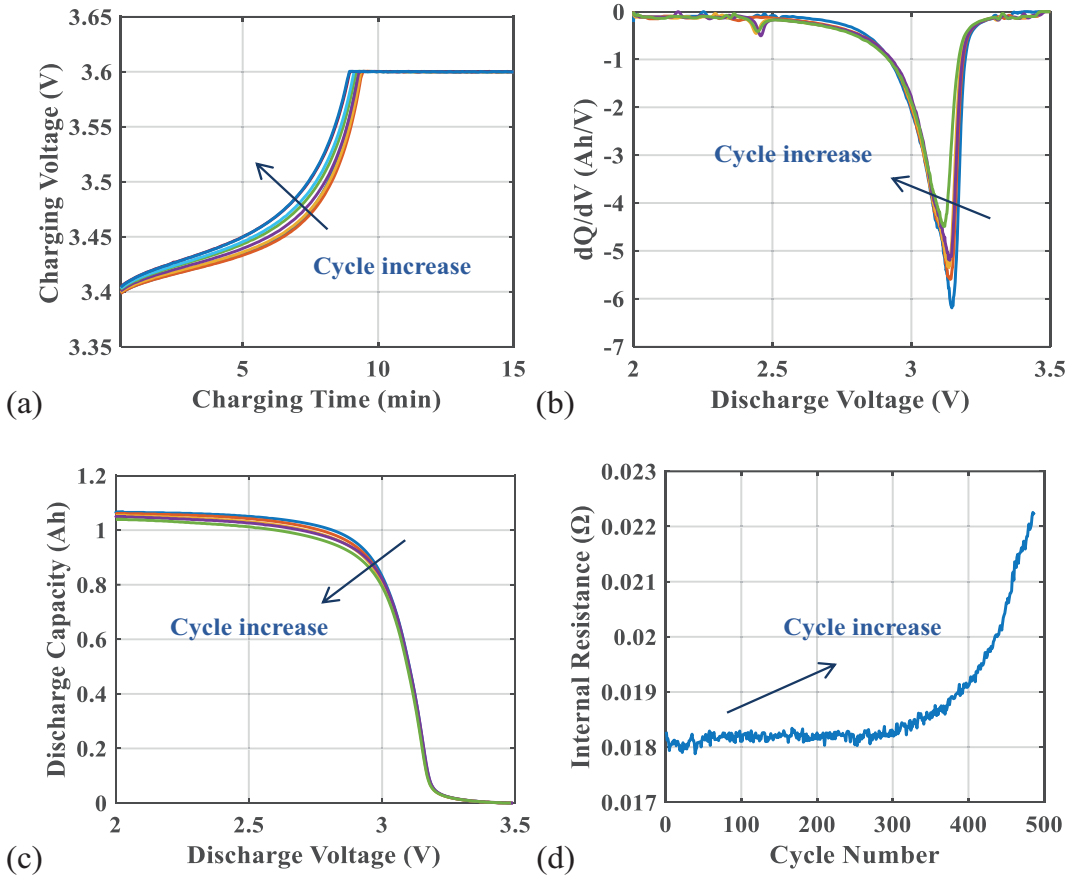

Fig. 6. Evolution curves of multiple cycles of a battery sample: (a) CCCV charging curves; (b) IC curves; (c) Discharge capacity vs. discharge voltage (*Q(V)*) curves; (d) Internal resistance vs. cycle number curves.

Table 4
Correlation between extracted features and the battery lifetime.

Twelve most correlated features	CNN feature channel	Handcrafted feature channel
1	0.75	0.92
2	0.59	0.91
3	0.58	0.76
4	0.56	0.63
5	0.56	0.58
6	0.55	0.52
7	0.55	0.51
8	0.55	0.51
9	0.54	0.48
10	0.54	0.45
11	0.53	0.42
12	0.51	0.42

$$C_l^{SRT} = p_3 \cdot \sqrt{l} + p_4 \tag{4}$$

where C_l^{linear} and C_l^{SRT} are the estimated discharge capacity at cycle l from the linear model and the SRT model, respectively.

In the two-channel feature engineering stage, the output is a candidate feature set $z_T = [z_C, z_H] \in R^{m \times d_3}$, $d_3 = 276$, which concatenates machine learned features z_C and handcrafted features z_H together.

3.1.3. Feature analysis

To examine the effectiveness of extracted features, we conduct a correlation analysis between extracted features and the target battery lifetime. The strength of correlation is measured by the Pearson correlation coefficient as defined in Eq. (5).

$$r^j = \frac{\sum_{i=1}^{m}\left(x_i^j - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sqrt{\sum_{i=1}^{m}\left(x_i^j - \bar{x}\right)^2}\sqrt{\sum_{i=1}^{m}\left(y_i - \bar{y}\right)^2}} \tag{5}$$

where r^j is the correlation between the j-th feature and the battery lifetime, x_i^j denotes the j-th feature of the i-th battery sample, and y_i denotes the lifetime of the i-th battery sample.

In each feature channel, we sort features according to their correlations with the battery lifetime. Table 4 lists the correlation coefficient of the most relevant twelve features from each channel. As shown in Table 4, both CNN features and handcrafted features exhibit certain correlation with the target battery lifetime. A few handcrafted features, which are carefully designed based on domain knowledge, appear to be higher correlated with the lifetime than CNN features. In Section IV·B, we further examine the performance of single-channel features and joint two-channel features through computational studies.

3.2. Stage 2: wrapper-based feature selection

In the two-channel feature set z_T, multicollinearity might exist among features. To examine the feature correlation, we arbitrarily choose ten features from the candidate set z_T, evaluate the pair-wise Pearson correlations, and demonstrate results in Fig. 7. It is observable that some features are highly linearly correlated. Besides these ten features, we also find a lot of linearly correlated features in the candidate set. To produce a more compact input space by shrinking features with collinearities, a wrapper feature selection process [34] is applied. In this stage, we input the two-channel features z_T into the wrapper-based feature selection, and finally output a compressed set of features $z_S \in$

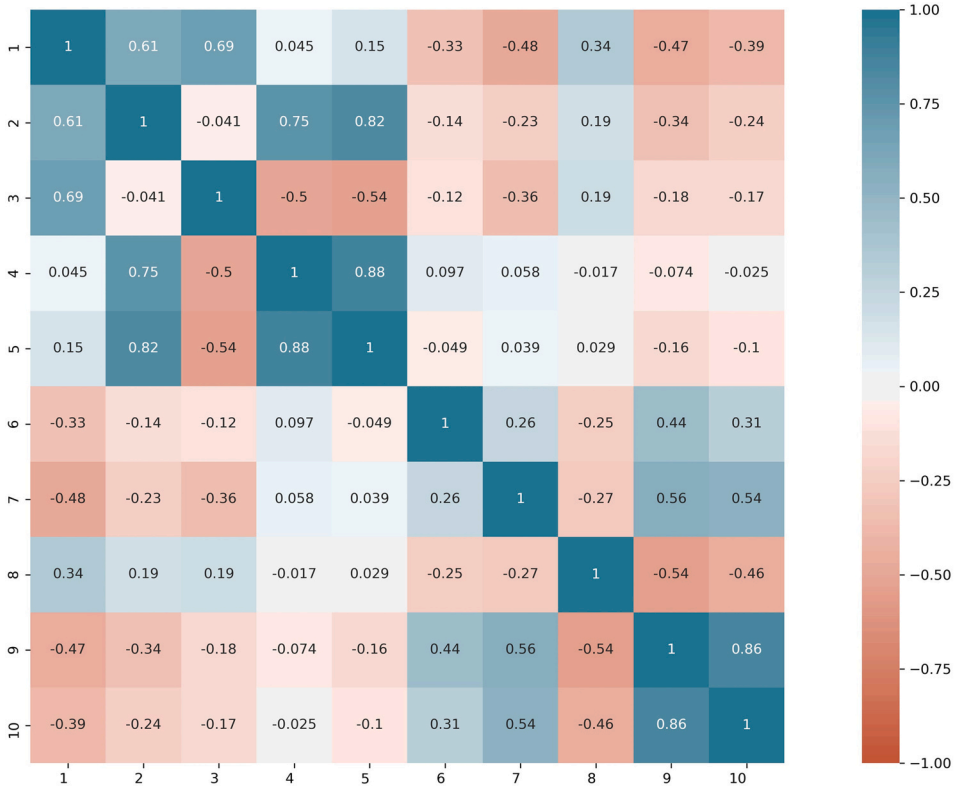

Fig. 7. Correlation plot of ten features from the candidate feature set.

$R^{m\times d_4}$, $d_4 = 132$. In z_S, 121 features come from the CNN feature channel, and the remaining 11 ones belong to the handcrafted feature channel.

3.3. Stage 3: lifetime prediction based on regression models

In this stage, we utilize different data-driven models to perform early-stage battery lifetime predictions. Two different prediction methods as formulated in Eqs. (6) and (7) are considered.

$$formulation\ 1: \hat{y}_i = F_{\Theta^j}^{(j)}\left(z_{S,i}\right),\quad i \in [1,m], j \in [1,5] \tag{6}$$

$$formulation\ 2: \hat{y}_i = F_{\Theta^j}^{(j)}\left(z_{S,i}, r_i\right),\quad i \in [1,m], j \in [1,5] \tag{7}$$

where $\hat{y}_i$ is the predicted lifetime of the i-th cell, $F_{\Theta^j}^{(j)}(\cdot)$ denotes the mapping of the j-th prediction model in five candidate data-driven models {GPR, elastic net, SVM, random forest (RF), and GBRT}, Θ^j denotes the parameter of the j-th model, and $z_{S,\,i}$ denotes the selected two-channel features of the i-th cell sample.

The difference between *formulation 1* and *2* is that an additional categorical input $r_i \in \{\mathrm{r}^1, \mathrm{r}^2, \ldots, \mathrm{r}^{72}\}$ is employed in *formulation 2*. In the studied Dataset A, cell samples underwent 72 different multi-step charging profiles. To integrate the operation condition, which has a large impact on battery degradation patterns, into our framework, we transform the multi-step charge profiles into 72 categorical inputs. The $r_i \in \{r^1, r^2, \ldots, r^{72}\}$ denotes the operation condition of the i-th sample. Performances of these two methods are examined in Section IV·B through computational experiments. For convenience, formulas of the following regression models all follow *formulation 1*.

(1) GPR

Because of the flexible, nonparametric and probabilistic characteristics [35], as well as working well on small datasets, the GPR model has been extensively applied in battery prognostic studies. Moreover, latter comparative experiments in Section IV·D indicate that the GPR model provides the best results for early-stage battery lifetime predictions. Thus, it is selected as the regressor in our framework.

The GPR model does not estimate any parametric solutions for Eq. (6). In fact, it assumes that for each input z, $F(z)$ is a sample from the Gaussian process (GP). The GPR calculates the joint probability distribution of all $F(z)$, which are jointly Gaussian. The mean function $m(z)$ and covariance function $k(z, z')$ characterize the distribution of $F(z)$. Here, the mean function $m(z)$ is set as 0 and the covariance function is adopted as the mostly used squared exponential kernel as depicted in Eq. (8).

$$k_{ij}\left(z_i, z_j\right) = \sigma_f^2 exp\left(\frac{-\left\|z_i, z_j\right\|^2}{2\sigma_l^2}\right) \tag{8}$$

where σ_f and σ_l are hyperparameters of the GPR model. They are optimized via maximizing the marginal log-likelihood function of Y [36]. By marginalizing the joint prior distribution over the training output, we can derive the posterior distribution of the testing output. The mean of the posterior is the predicted lifetime.

(2) Elastic net

The elastic net is a regularized linear regression model [1] which integrates both L1- and L2-norm penalties. The problem of learning an elastic net can be formulated as Eq. (9):

$$\beta = argmin_\beta\left\{\|Y - z_S\beta\|_2^2 + \lambda\left(\frac{1-\alpha}{2}\|\beta\|_2^2 + \alpha\|\beta\|_1\right)\right\} \tag{9}$$

where β is an $m \times 1$ vector of model coefficients, λ is a regularization parameter, and $\alpha \in [0,1]$ is a hyperparameter trading off the L1- and L2-norm penalties.

(3) SVM

The SVM model maps a task in low-dimensional space into a linear task in high-dimensional space via a nonlinear transformation $\psi(\cdot)$ as expressed in Eq. (10).

$$y = w \cdot \psi(z_S) + b \tag{10}$$

where w is the weight vector and b is the bias term. The SVM can be formulated as an optimization problem described in Eq. (11):

$$\begin{aligned} &min \quad J(w,b,\xi) = \frac{1}{2}w^{\mathrm{T}}w + C\sum_{i}^{n}\left(\xi_i^1 + \xi_i^2\right) \\ &\text{s.t.} \begin{cases} y_i - (w\psi(z_i) + b) \le \varepsilon + \xi_i^1 \\ (w\psi(z_i) + b) - y_i \le \varepsilon + \xi_i^2 \\ \xi_i^1, \xi_i^2 \ge 0 \end{cases} \end{aligned} \tag{11}$$

where C is a regularization parameter, ξ_i^1 and ξ_i^2 are slack variables as well as $\varepsilon > 0$ is an error allowed by the model. By solving the Lagrange dual problem of Eq. (8), the optimal w and b can be deduced through the duality [37]. Here, the radial basis function is adopted as the kernel function for SVM.

(4) RF

The RF is an ensemble algorithm which aggregates outputs of multiple decision trees [38]. The idea of the RF is to generate different decision trees, each tree is trained with a random bootstrap sampled subset of training samples. In each split of tree nodes, a random feature subset is utilized. The final output of the battery lifetime is the aggregation of predictions from these various trees. Such randomness improves the tree diversity and renders the RF model learn more patterns of the data.

(5) GBRT

The GBRT is also an ensemble algorithm which aggregates many 'weak' trees to a 'strong' tree with better robustness. The idea of the GBRT is to iteratively generate multiple decision trees [25]. In each iteration, the new tree is fitted to the residual of the previous tree. This process is essentially a functional gradient descent, the model is iteratively updated by minimizing the loss function (adopted as the squared error function here). The final output of the battery lifetime is a weighted sum of predictions from multiple trees.

The pseudo code of the framework is presented in **Algorithm 1**.

Algorithm 1: Framework of the early-stage battery lifetime prediction

Stage 1: Two-channel feature engineering
Channel 1: CNN-based automatic feature extraction
Input: V, I, and T in first 100 cycles
$\boldsymbol{x}_i = InputTensorConstruction(V_i, I_i, T_i) \in R^{100\times100\times3}$
$X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_m] \in R^{m\times100\times100\times3}$
Input for CNN: $\hat{X} = ZeroPadding(X) \in R^{m\times224\times224\times3}$
$z_C = \varphi(\hat{X}) \in R^{m\times d_1}, d_1 = 256$
Output: latent features z_C
End Channel 1
Channel 2: Domain knowledge based handcrafted features
Input: V, I, Q, T, and internal resistance (Ω) in first 100 cycles
$z_H = CraftedFeatures(V, I, Q, T, \Omega) \in R^{m\times d_2}$, $d_2 = 20$
Output: z_H
End Channel 2
$z_T = [z_C, z_H] \in R^{m\times d_3}$, $d_3 = 276$
Output: Two-channel features z_T
End Stage 1

Stage 2: Feature selection
$z_S = WrapperFeatureSelection(z_T) \in R^{m\times d_4}$, $d_4 = 132$

(continued on next column)

(continued)

Stage 3: Battery lifetime prediction based on regression models
Input: Selected features z_S, battery lifetime Y
$z_S^{train}, z_S^{test}, Y_{train}, Y_{test} = TrainTestSpliting(z_S, Y)$
or $z_S^{train}, z_S^{test}, r^{train}, r^{test}, Y_{train}, Y_{test} = TrainTestSpliting(z_S, r, Y)$
Model training:
for j = 1 to 5
$\Theta^{j^*} = \underset{\Theta^j}{argmin}\ J\left(F_{\Theta^j}^{(j)}\left(z_S^{train}\right), Y_{train}\right), \quad j \in [1,5]$
or $\Theta^{j^*} = \underset{\Theta^j}{argmin}\ J\left(F_{\Theta^j}^{(j)}\left(z_S^{train}, r^{train}\right), Y_{train}\right), \quad j \in [1,5]$
end for
Output: The fine-trained models
Model prediction:
for j = 1 to 5
$EvaluateMetrics(F_{\Theta}{}^{*(j)}(z_S^{test}), Y_{test})$
or $EvaluateMetrics(F_{\Theta}{}^{*(j)}(z_S^{test}, r^{test}), Y_{test})$
end for
Output: The best prediction model and metrics
End Stage 3

4. Computational experiments

To justify the value of the proposed framework for early-stage battery lifetime predictions, five sets of computational experiments are conducted. First, the effectiveness of each stage in the framework is verified separately. In Section IV·B, the performance of features extracted from various channels (CNN or handcrafted) are discussed. In Section IV·C, the effect of the wrapper feature selection method is evaluated by comparing with widely used methods. In Section IV·D, the performance of various data-driven models towards our task is evaluated. Next, in Section F, the robustness of the proposed framework towards different battery materials and operation conditions is examined. Finally, in Section G, the superiority of the proposed framework is examined by comparing with state-of-the-art benchmarks.

4.1. Experimental setup

In this study, Dataset A is primarily investigated to evaluate the method performance. Datasets B and C are utilized to further verify the method robustness.

In Dataset A, 123 cell samples excluding one shortest-lived cell are studied. Particularly, given the difference between long-life and short-life batteries, a stratified random sampling method is adopted to partition the data. The 123 batteries are randomly split into 70% train set (86 training samples) and 30% test set (37 testing samples), while the ratio of long-life batteries to short-life batteries is identical in the train and test set.

The performance of the framework is evaluated using three metrics expressed in Eqs. (12)–(14).

- RMSE (root of mean squared error):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{12}$$

- MAPE (mean absolute percentage error):

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \tag{13}$$

- R^2 (coefficient of determination):

Table 5
Formulation 1: lifetime prediction results of using different features.

	Feature type	RMSE (cycles)	MAPE (%)	R^2	Number of features
Before wrapper feature selection	CNN features	186	13.1	0.77	256
	Hand-crafted features	145	11.6	0.86	20
	Combined two-channel features	141	10.2	0.87	276
After wrapper feature selection	Selected CNN features	158	11.4	0.83	121
	Selected hand-crafted features	133	8.7	0.88	11
	Selected two-channel features	112	8.2	0.92	132

Table 6
Formulation 2: lifetime prediction results of using different features.

	Feature type	RMSE (cycles)	MAPE (%)	R^2	Number of features
Before wrapper feature selection	CNN features	224	13.8	0.66	256
	Hand-crafted features	187	12.3	0.77	20
	Combined two-channel features	175	11.2	0.80	276
After wrapper feature selection	Selected CNN features	202	12.1	0.73	121
	Selected hand-crafted features	185	11.8	0.78	11
	Selected two-channel features	151	9.0	0.85	132

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{14}$$

where n is the number of batteries, and y_i and y_i are the observed and the predicted lifetime of battery i, respectively. Specifically, lower RMSE and MAPE mean smaller predictive errors, higher R^2 represents better predictive performance.

4.2. Stage 1: analysis of extracted features

In this section, the performance of the first stage in the proposed framework, the two-channel feature engineering, is examined. Comparative experiments are conducted to evaluate the effectiveness of features extracted from various channels (CNN or handcrafted). Several types of feature combinations are considered: (a) unselected CNN features; (b) unselected handcrafted features; (c) combined two-channel features without feature selection; (d) selected CNN features by the wrapper method; (e) selected handcrafted features; (f) selected two-channel features. For a fair comparison, these various feature combinations are all input into a GPR model for the battery lifetime prediction.

Results of *formulation 1* and *2* proposed in Section III·C are summarized in Tables 5 and 6. In comparison, *formulation 2* integrating the operation conditions as inputs does not generate a positive impact on the lifetime prediction than *formulation 1*. Prediction errors of *formulation 2* are larger while the R^2 are smaller. These underperformed prediction results may be attributed to the limited number of battery samples. In Dataset A, there are 123 cell samples cycled under 72 different operation conditions. When the operation condition is added as an input, for each category of operation condition r^j, $j \in [1,72]$, there are only 1– 3 battery training samples. However, the performance of the data-driven model is largely influenced by the quantity of training samples. Therefore, in *formulation 2*, the number of training samples for each category is far from enough to train a good prediction model. This method needs to be further examined when more battery degradation samples in each category are available. Subsequent computational studies are all based on *formulation 1*.

Fig. 8 demonstrates the results of using (a)/(b)/(c)/(f) type of features. From Table 5 and Fig. 8, we have the following findings:

(1) Before the wrapper feature selection, the unselected CNN-channel features (RMSE, MAPE and R^2 are 186 cycles, 13.1%, and 0.77, respectively) underperforms the unselected handcrafted features (RMSE, MAPE and R^2 are 145 cycles, 11.6%, and 0.86, respectively). This finding still holds after the wrapper feature selection. Selected handcrafted features produce a smaller RMSE (133 cycles) and MAPE (8.7%), and a larger R^2 (0.88) than selected CNN features, whose RMSE, MAPE and R^2 are 158 cycles, 11.4%, and 0.83, respectively. It is known that CNN works well for the image classification and similar classification tasks. However, for the battery lifetime prediction problem, which has a more accurate requirement on the regression output, features automatically learned from CNN may underperform those finely crafted features, since the domain knowledge plays a dominant role in the latter.
(2) Combined two-channel features outperform single-channel features. This conclusion holds both before and after the feature selection. Before the feature selection, by using two-channel features, we get lower RMSE (141 cycles) and MAPE (10.2%), and a higher R^2 (0.87) than using single-channel features. After the feature selection, by adopting two-channel features, we again obtain lower RMSE (112 cycles) and MAPE (8.2%), and a larger R^2 (0.92) than using single-channel features. These results indicate that CNN features and handcrafted features have certain positive interactions and a join of them can improve the battery lifetime prediction.

4.3. Stage 2: comparison of feature selection methods

As shown in Table 5 and Fig. 8(c) and (d), after the wrapper feature selection, the performance of single-channel features and combined two-channel features are significantly improved than those of unselected features. This verifies the effectiveness of the wrapper feature selection towards our task.

To further examine the effect of the wrapper feature selection method, six widely applied dimension reduction techniques, the principal component analysis (PCA), non-negative matrix factorization (NMF), autoencoder, GBRT, and RF, are considered as comparative benchmarks. First three methods are unsupervised dimension reduction methods. For a fair comparison, the unselected two-channel features are input into these methods and then the compressed features are all fed into a GPR model for predictions. While the GBRT and RF are supervised embedded feature selection methods, which automatically select features during the model training process. They can directly output the target lifetime and do not require additional prediction model. Results of the wrapper method and considered benchmarks are summarized in Table 7.

As shown in Table 7, the wrapper feature selection method outperforms benchmarking dimension reduction methods with the lowest errors and the highest R^2. Specifically, the RMSE, MAPE and R^2 of the unsupervised autoencoder are 141 cycles, 10.2%, and 0.87, respectively. Metrics of the supervised GBRT method are 173 cycles, 11.1%, and 0.8, respectively. This indicates that the wrapper method is more appropriate for our task and is thus adopted for subsequent analyses.

4.4. Stage 3: comparison of prediction models

To examine the effect of various models for the early-stage battery lifetime prediction task, both the unselected and selected two-channel features are fed into six models, the GPR, elastic net, SVM, RF, GBRT,

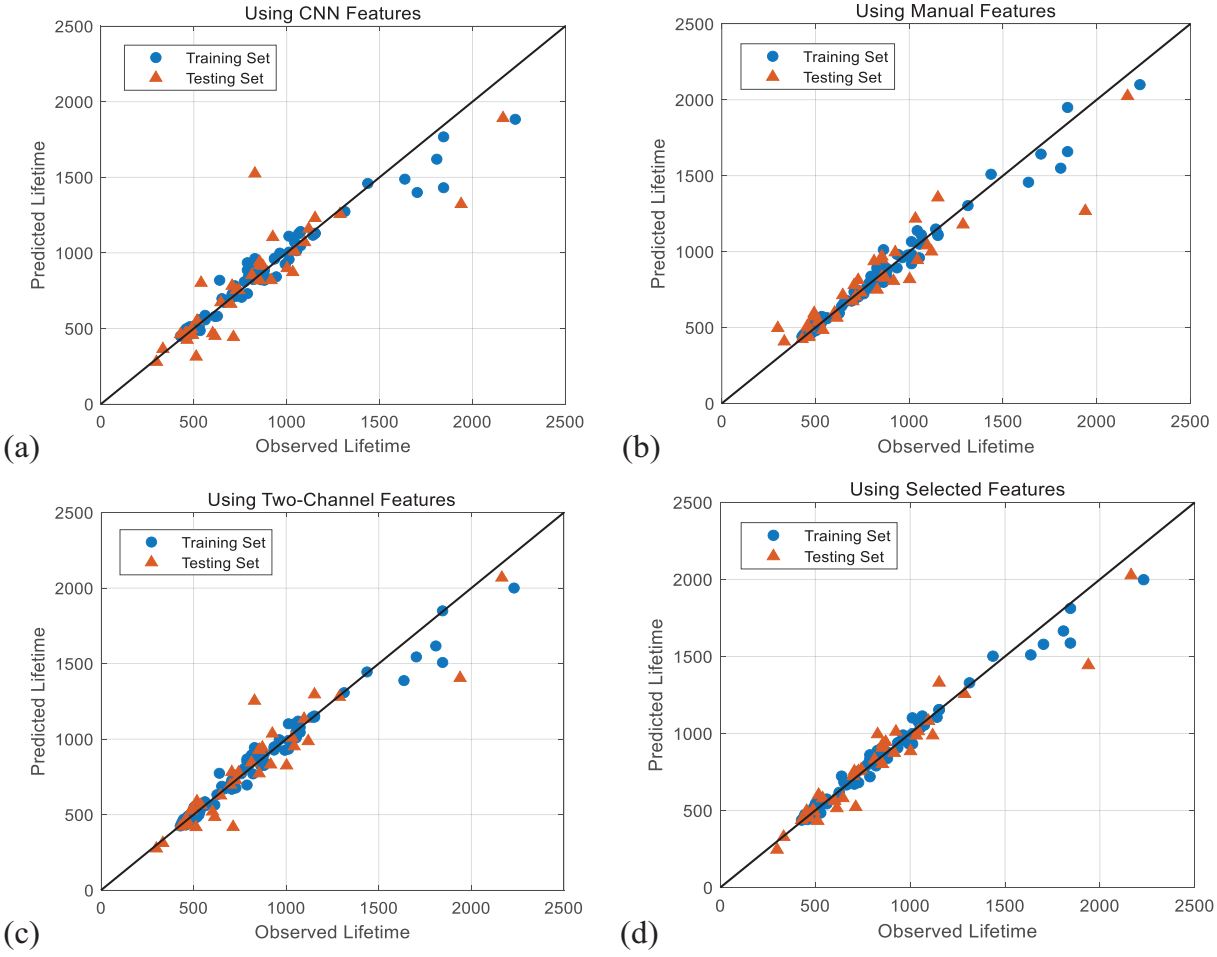

Fig. 8. Observed lifetime v.s. predicted lifetime: (a) using unselected CNN features; (b) using unselected hand-crafted features; (c) using two-channel features without feature selection; (d) using selected two-channel features.

Table 7
Results of different dimensionality reduction methods.

Feature selection method	RMSE (cycles)	MAPE (%)	R^2
Wrapper method	112	8.2	0.92
PCA	159	13.1	0.83
NMF	171	13.3	0.80
Autoencoder	141	10.2	0.87
GBRT	173	11.1	0.80
RF	181	12.9	0.78

Table 8
Lifetime prediction results of different machine learning models.

Model	Selected two-channel features			Unselected two-channel features		
	RMSE (cycles)	MAPE (%)	R^2	RMSE (cycles)	MAPE (%)	R^2
GPR	112	8.2	0.92	141	10.2	0.87
Elastic net	114	8.5	0.91	132	9.4	0.88
SVM	118	9.1	0.91	139	10.8	0.88
RF	168	12.0	0.80	181	12.9	0.78
GBRT	146	11.0	0.84	173	11.1	0.80
CNN	233	19.3	0.65	269	20.1	0.62

and CNN, for the comparison. The same data split is adopted for each model. Results of various prediction models are summarized in Table 8.

When feeding selected two-channel features, among these six models, the GPR model provides the best results, with the lowest RMSE (112 cycles) and MAPE (8.2%), and the largest R^2 (0.92). Yet, the performance difference among the GPR, elastic net, and SVM model is not significant, i.e., the maximum differences of RMSE, MAPE, and R^2 are not more than 6 cycles, 0.9%, and 0.01, respectively. In contrast, the RF, GBRT, and CNN model, which are relatively complex than the previous three simple models, generate significantly worse results. Particularly, the CNN presents the worst performance, with the RMSE 233 cycles, MAPE 19.3%, and R^2 only 0.65. The reason may be that the amount of training samples is quite limited (no more than 90) and is insufficient to train a high-quality CNN containing a large number of parameters. Therefore, for our target task with a limited sample size, simple models with less parameters, such as the GPR, may have better effects.

When feeding unselected two-channel features, results of all models are worse than those of using selected features. This again proves the validity of the wrapper feature selection on our task. Instead of the GPR, the elastic net now becomes the best-performed model. It could be that the elastic net is a regularized model and automatically scale down unimportant features, thus can better handle unselected high-dimensional features. Yet, the performance of the elastic net, GPR, and SVM are still comparable. The RF, GBRT and CNN still underperform the previous three models.

Based on the above discussions, the GPR model offering the best performance is adopted for our subsequent analyses.

Table 9
Lifetime prediction results of ten testing cells in Dataset A.

Cell No.	True lifetime	Predicted lifetime	AE (cycles)	APE (%)
1	334	328	5	1.6%
2	493	494	1	0.2%
3	428	438	9	2.2%
4	742	766	23	3.1%
5	812	831	18	2.3%
6	916	874	42	4.6%
7	1043	1015	29	2.7%
8	1097	1081	15	1.4%
9	1287	1256	31	2.4%
10	2165	2026	138	6.4%

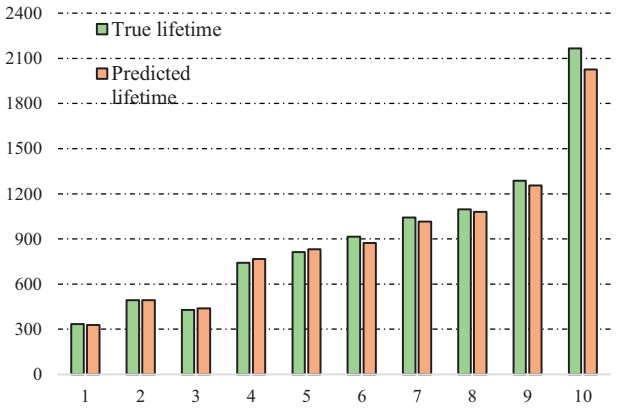

Fig. 9. Lifetime prediction results of ten testing cells in Dataset A.

4.5. Analysis of representative cells

To demonstrate the effect of the proposed method more clearly, predictions on ten representative cells from the test set are analyzed. Two metrics, the absolute error (AE) and absolute percentage error (APE) between the true and predicted battery lifetime, are considered. Results are summarized in Table 9 and Fig. 9.

As shown in Table 9 and Fig. 9, the proposed method can effectively predict the lifetime of most testing cells. The error metrics AE and APE can be as low as only 1 cycle and 0.2%, respectively, for cell No. 2. In general, predictions are more accurate on short-lived cells (such as cell No. 1, 2, 3, 4 and 5) than those on long-lived cells. For instance, for cell No. 10 with the longest lifetime of 2165 cycles, the AE and APE are 138 cycles and 6.4%, respectively, which are much higher than metrics of cell No. 2.

There are two reasons for this result. First, in Dataset A, the number of long-lived cells is much less than that of short-lived cells. Such unbalanced sample distribution leads to relatively worse predictions on long-lived cells. Therefore, more long-lived training samples are needed in the future for better predictions. Second, the proposed early-stage lifetime prediction method utilizes information from only the first 100 cycles. Therefore, it is more difficult to capture the long-term degradation for long-lived cells using just a small proportion of data along the battery whole life.

4.6. Verification of the prediction robustness

To further verify the robustness of the proposed framework, the well-trained model from Dataset A is tested on Dataset B and C. Specifications of four tested cells are listed in Table 10. It should be noted that the temperature signal is not recorded in these two datasets; therefore, temperature-related features are not included in the verification. Results are demonstrated in Table 10 and Fig. 10.

As shown in Table 10 and Fig. 10, the proposed framework can effectively predict the lifetime of different types of LIBs. For instance, for the LiCoO2 cell B2, the AE and APE between the true lifetime and predicted lifetime are 19 cycles and 3.6%, respectively. For the LFP cell C2 with higher current rates and lower cutoff voltages, the AE and APE are 40 cycles and 8.6%, respectively. This indicates that the proposed framework can be generalized to different battery materials and operation conditions.

4.7. Comparison with other published methods

In this section, we compare the proposed method with a set of benchmarking methods to verify its superiority from two aspects, the prediction accuracy and prediction horizon.

To validate the prediction accuracy, the proposed method is compared with three benchmarking methods from most recent publications [1,25,39]. They all target at the early-stage battery lifetime prediction based on Dataset A. Yet, features and prediction models considered in [1,25,39] are different from our method.

1) Benchmark 1

In [1], a set of features were manually crafted based on the first-100-cycle degradation data, and were input into an elastic net for battery lifetime predictions. A combination of six features presented the best prediction results in [1], and thus is selected as Benchmark 1.

2) Benchmark 2

In [25], eighteen handcrafted features were presented based on the first-250-cycle degradation data, and were input into a GBRT model for battery lifetime prediction. We reproduce these features using the first-100-cycle degradation data and feed them into a GBRT model as our Benchmark 2.

3) Benchmark 3

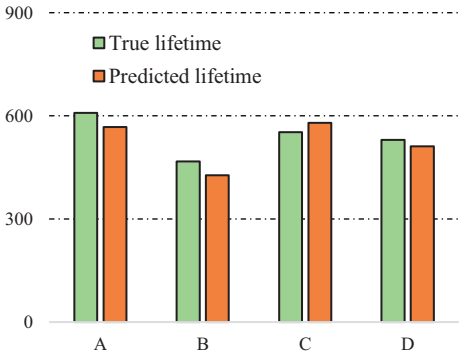

Fig. 10. Lifetime prediction results of four testing cells in Dataset B and C.

Table 10
Results of four testing cells in Dataset B and C.

	Specifications					Performance		
Cell No.	Material	Charge/discharge rate	Nominal capacity	Upper/lower cutoff voltage	True lifetime	Predicted lifetime	AE (cycles)	APE (%)
B1	LiCoO2	0.5/0.5C	1.1 Ah	4.2/2.7 V	552	580	28	5.0%
B2					530	511	19	3.6%
C1	LFP	1/1C	1.2 Ah	3.6/2 V	609	557	52	8.5%
C2					467	427	40	8.6%

Table 11
Comparison with state-of-the-art benchmarks in terms of prediction accuracy.

	RMSE (cycles)	MAPE (%)	R^2
Proposed method	112	8.2	0.92
Benchmark 1	138	10.7	0.87
Benchmark 2	147	13.6	0.85
Benchmark 3	162	14.4	0.83

In [39], four handcrafted features based on the first-100-cycle degradation data were selected and input into a general regression neural network for battery lifetime predictions. We adopt it as the Benchmark 3.

Comparison results are shown in Table 11 and Fig. 11. The same training and testing data split with previous sections is adopted. We can observe that the proposed method is superior to these benchmark methods, with the lowest RMSE and MAPE, as well as the highest R^2. Compared with Benchmark 1, in which Dataset A was first provided, we can reduce the RMSE and MAPE by 18.2% and 23.4%, respectively, and improve the R^2 by 5.7%. Our superior performance than the benchmarks may be attributed to the proposed much larger feature set which covers more potential effective information for battery lifetime prediction, as well as the GPR prediction model which is more applicable to our task.

Another advantage of our proposed method is the early prediction horizon, which means that only the first-100-cylce data is required for an accurate battery lifetime prediction and later-stage degradation data are not needed. However, in many mainstream RUL prediction methods, a large proportion of the battery capacity degradation curve is required for the model training, and the battery lifetime is predicted by extrapolating the model to the end of life. Generally, it is challenging for them to obtain accurate predictions when only the first-few-cycle data are available for model training and parameter updating. To verify the superiority of our method in the early prediction horizon, we compare it with two popular RUL prediction methods.

1) Ensemble model + particle filter: In [7], an ensemble model fusing the exponential model and polynomial model was proposed to track the battery capacity degradation trend. The particle filter technique was adopted to update model parameters and perform RUL prediction.

2) Coulombic efficiency (CE) model + particle filter: In [4], a semi-empirical CE-based model incorporating the particle filter technique was proposed to model the battery capacity degradation and perform the RUL prediction.

Computational results are summarized in Table 12. For two benchmarking methods, the whole life degradation data of training samples are used for the model training and parameter initialization. The first-100-cycle data of test samples are used for the model parameter updating. As shown in Table 12, we can see the proposed method outperforms two benchmarking methods in terms of much lower errors and higher R^2. This experiment validates the superiority of our method in the early prediction horizon.

Table 12
Comparison with benchmarks in terms of prediction horizon.

Method	RMSE (cycles)	MAPE (%)	R^2
Proposed method	112	8.2	0.92
Ensemble model + particle filter	304	23.3	0.59
CE model + particle filter	238	18.6	0.64

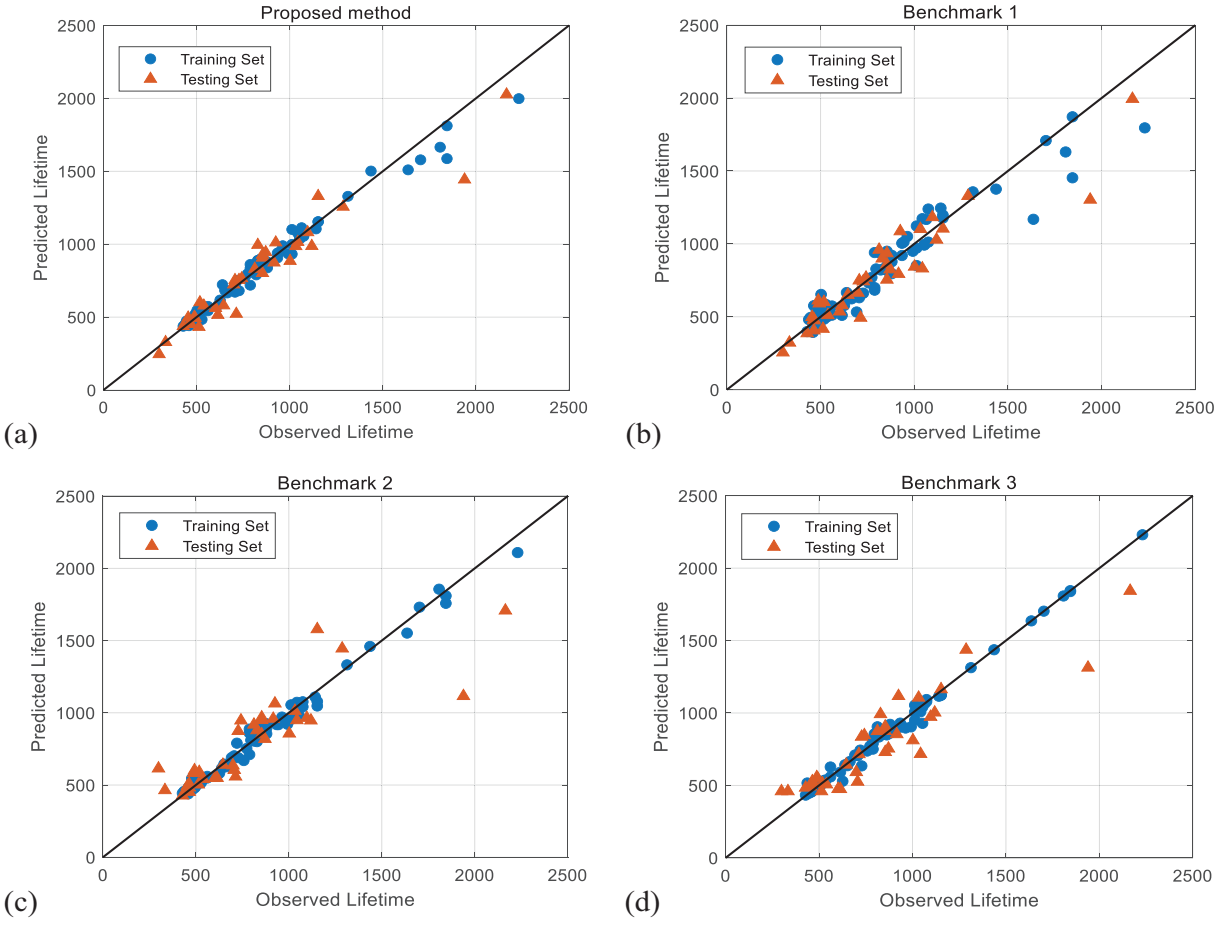

Fig. 11. Observed lifetime v.s. predicted lifetime: (a) the proposed method; (b) Benchmark 1; (c) Benchmark 2; (d) Benchmark.

5. Conclusions

This paper developed a novel data-driven framework for early-stage LIB lifetime predictions using only battery data of first 100 cycles. This promising study could help with accelerating the LIB performance evaluation during its development and optimization process, which is yet a challenging task for most existing LIB lifetime prediction methods, due to the limited information provided by early cycles. The proposed data-driven framework had three stages: 1) a two-channel data feature engineering strategy was proposed for creating a set of candidate features incorporating CNN-based and handcrafted features; 2) a wrapper search-based feature selection method was adopted to select meaningful features from the two-channel feature set; 3) a GPR model was employed to produce early-stage battery lifetime prediction via selected features. Computational experiments were conducted on LIB datasets of different battery materials and operation conditions for the method verification. The effectiveness of each stage of the proposed framework was evaluated via an ablation study. Main research findings based on computational experiments are summarized as follows:

1) CNN features and handcrafted features could complement each other. The appropriate joining of them could benefit the battery lifetime prediction.
2) The wrapper search-based feature selection method and the GPR model were effective enough on selecting valuable input features and developing the prediction model in the considered early-stage battery lifetime prediction task. This was validated by comparing their superior performance with famous benchmarks considered in the feature selection and prediction model development tasks.
3) This framework demonstrated the generalization ability to different battery materials and operation conditions.
4) This method offered both the higher prediction accuracy and earlier lifetime prediction than considered benchmarks published in the literature.

In the future, we would like to improve our method from two directions:

1) Further uplift the accuracy and generalization ability: We plan to collect more battery degradation samples via conducting more experiments in our laboratory as well as collaborate with the industrial partner to obtain data from practical applications. We plan to improve the prediction accuracy and generalization ability of our framework with more battery degradation samples and more operating conditions available.
2) Adapt the proposed framework into more challenging application scenarios: We plan to explore more challenging applications that the proposed framework could be potentially adapted into, such as the online battery lifetime prediction with only partial data accessible.

CRediT authorship contribution statement

Zicheng Fei: Methodology, Software, Validation, Formal Analysis, Investigation, Writing-Original Draft

Zijun Zhang: Conceptualization, Methodology, Writing – Review&Editing, Supervision, Project administration

Fangfang Yang: Data Curation, Formal analysis

Kwok-Leung Tsui: Resources, Writing-Review&Editing, Supervision, Funding acquisition

Lishuai Li: Writing-Review&Editing

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.

Acknowledgments

This work was supported in part by the Hong Kong Research Grants Council General Research Fund Project with No. 11204419 and 11215418, and was supported in part by the National Natural Science Foundation of China Youth Scientist Fund Project with No. 52007160.

References

[1] K.A. Severson, P.M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M.H. Chen, M. Aykol, P.K. Herring, D. Fraggedakis, Data-driven prediction of battery cycle life before capacity degradation, Nat. Energy 4 (5) (2019) 383–391.
[2] N. Wassiliadis, J. Adermann, A. Frericks, M. Pak, C. Reiter, B. Lohmann, M. Lienkamp, Revisiting the dual extended Kalman filter for battery state-of-charge and state-of-health estimation: a use-case life cycle analysis, J. Energy Storage 19 (2018) 73–87.
[3] X. Lai, Y. Huang, X. Han, H. Gu, Y. Zheng, A novel method for state of energy estimation of lithium-ion batteries using particle filter and extended Kalman filter, J. Energy Storage 43 (2021), 103269.
[4] F. Yang, X. Song, G. Dong, K.-L. Tsui, A coulombic efficiency-based model for prognostics and health estimation of lithium-ion batteries, Energy 171 (2019) 1173–1182.
[5] Y. Ma, Y. Chen, X. Zhou, H. Chen, Remaining useful life prediction of lithium-ion battery based on Gauss–Hermite particle filter, IEEE Trans. Control Syst. Technol. 27 (4) (2019) 1788–1795.
[6] M.V. Micea, L. Ungurean, G.N. Cârstoiu, V. Groza, Online state-of-health assessment for battery management systems, IEEE Trans. Instrum. Meas. 60 (6) (2011) 1997–2006.
[7] Y. Xing, E. W. M. Ma, K.-L. Tsui, and M. Pecht, "An ensemble model for predicting the remaining useful performance of lithium-ion batteries," Microelectron. Reliab., vol. 53, no. 6, pp. 811–820, 2013/06/01/, 2013.
[8] D. Yang, X. Zhang, R. Pan, Y. Wang, Z. Chen, A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve, J. Power Sources 384 (2018) 387–395.
[9] J. Liu, Z. Chen, Remaining useful life prediction of lithium-ion batteries based on health indicator and Gaussian process regression model, IEEE Access 7 (2019) 39474–39484.
[10] Y. Li, D.-I. Stroe, Y. Cheng, H. Sheng, X. Sui, R. Teodorescu, On the feature selection for battery state of health estimation based on charging–discharging profiles, J. Energy Storage 33 (2021).
[11] H. Feng, D. Song, A health indicator extraction based on surface temperature for lithium-ion batteries remaining useful life prediction, J. Energy Storage 34 (2021), 102118, 2021/02/01/.
[12] Y. Li, M. Abdel-Monem, R. Gopalakrishnan, M. Berecibar, E. Nanini-Maury, N. Omar, P. van den Bossche, J. Van Mierlo, A quick on-line state of health estimation method for Li-ion battery with incremental capacity curves processed by Gaussian filter, J. Power Sources 373 (2018) 40–53, 2018/01/01/.
[13] Z. Wei, H. Ruan, Y. Li, J. Li, C. Zhang, H. He, Multistage state of health estimation of lithium-ion battery with high tolerance to heavily partial charging, IEEE Trans. Power Electron. 37 (6) (2022) 7432–7442.
[14] X. Bian, Z. Wei, J. He, F. Yan, L. Liu, A novel model-based voltage construction method for robust state-of-health estimation of lithium-ion batteries, IEEE Trans. Ind. Electron. 68 (12) (2021) 12173–12184.
[15] S. Zhang, B. Zhai, X. Guo, K. Wang, N. Peng, and X. Zhang, "Synchronous estimation of state of health and remaining useful lifetime for lithium-ion battery using the incremental capacity and artificial neural networks," J. Energy Storage, vol. 26, pp. 100951, 2019/12/01/, 2019.
[16] G. Ma, Y. Zhang, C. Cheng, B. Zhou, P. Hu, Y. Yuan, Remaining useful life prediction of lithium-ion batteries based on false nearest neighbors and a hybrid neural network, Appl. Energy 253 (2019), 113626.
[17] Y. Zhang, Q. Tang, Y. Zhang, J. Wang, U. Stimming, A.A. Lee, Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using machine learning, Nat. Commun. 11 (1) (Apr 6, 2020) 1706.
[18] E. Locorotondo, V. Cultrera, L. Pugi, L. Berzi, M. Pierini, and G. Lutzemberger, "Development of a battery real-time state of health diagnosis based on fast impedance measurements," *Journal of Energy Storage,* vol. 38, 2021.
[19] X. Hu, J. Jiang, D. Cao, B. Egardt, Battery health prognosis for electric vehicles using sample entropy and sparse Bayesian predictive modeling, IEEE Trans. Ind. Electron. 63 (4) (2015) 2645–2656.
[20] D. Roman, S. Saxena, V. Robu, M. Pecht, D. Flynn, Machine learning pipeline for battery state-of-health estimation, Nat. Mach. Intell. 3 (5) (2021) 447–456.
[21] N. Yang, Z. Song, H. Hofmann, J. Sun, Robust state of health estimation of lithium-ion batteries using convolutional neural network and random forest, J. Energy Storage 48 (2022), 103857, 2022/04/01/.
[22] Y. Fan, F. Xiao, C. Li, G. Yang, X. Tang, A novel deep learning framework for state of health estimation of lithium-ion battery, J. Energy Storage 32 (2020), 101741, 2020/12/01/.
[23] Z. Fei, F. Yang, K.-L. Tsui, L. Li, Z. Zhang, Early prediction of battery lifetime via a machine learning based framework, Energy 225 (2021).
[24] F. Xu, F. Yang, Z. Fei, Z. Huang, K.-L. Tsui, Life prediction of lithium-ion batteries based on stacked denoising autoencoders, Reliab. Eng. Syst. Saf. 208 (2021).

[25] F. Yang, D. Wang, F. Xu, Z. Huang, K.-L. Tsui, Lifespan prediction of lithium-ion batteries based on various extracted features and gradient boosting regression tree model, J. Power Sources 476 (2020).
[26] L. Ren, J. Dong, X. Wang, Z. Meng, L. Zhao, J. Deen, A data-driven auto-CNN-LSTM prediction model for lithium-ion battery remaining useful life, IEEE Trans. Ind. Inform. 17 (5) (2021) 3478–3487, https://doi.org/10.1109/TII.2020.3008223.
[27] L. Ren, L. Zhao, S. Hong, S. Zhao, H. Wang, L. Zhang, Remaining useful life prediction for lithium-ion battery: a deep learning approach, IEEE Access 6 (2018) 50587–50598.
[28] X. Hu, L. Xu, X. Lin, M. Pecht, Battery lifetime prognostics, Joule 4 (2) (2020) 310–346.
[29] X. Li, Q. Ding, J.-Q. Sun, Remaining useful life estimation in prognostics using deep convolution neural networks, Reliab. Eng. Syst. Saf. 172 (2018) 1–11.
[30] X. Song, F. Yang, D. Wang, K. Tsui, Combined CNN-LSTM network for state-of-charge estimation of lithium-ion batteries, IEEE Access 7 (2019) 88894–88902.
[31] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and Transferring Mid-Level Image Representations Using Convolutional Neural Networks." pp. 1717-1724.
[32] O. Badmos, A. Kopp, T. Bernthaler, G. Schneider, Image-based defect detection in lithium-ion battery electrode using convolutional neural networks, J. Intell. Manuf. 31 (4) (2020) 885–897.
[33] M. Broussely, S. Herreyre, P. Biensan, P. Kasztejna, K. Nechev, R.J. Staniewicz, Aging mechanism in Li ion cells and calendar life predictions, J. Power Sources 97-98 (2001) 13–21, 2001/07/01/.
[34] X. Hu, Y. Che, X. Lin, S. Onori, Battery health prediction using fusion-based feature selection and machine learning, IEEE Trans. Transport. Electrification 7 (2) (2021) 382–398, https://doi.org/10.1109/TTE.2020.3017090.
[35] H. Sheng, X. Liu, L. Bai, H. Dong, Y. Cheng, Small sample state of health estimation based on weighted Gaussian process regression, J. Energy Storage 41 (2021), 102816.
[36] L. Fan, P. Wang, Z. Cheng, A remaining capacity estimation approach of lithium-ion batteries based on partial charging curve and health feature fusion, J. Energy Storage 43 (2021), 103115.
[37] D. Yang, Y. Wang, R. Pan, R. Chen, Z. Chen, State-of-health estimation for the lithium-ion battery based on support vector regression, Appl. Energy 227 (2018) 273–283, 2018/10/01/.
[38] Y. Li, C. Zou, M. Berecibar, E. Nanini-Maury, J.C.W. Chan, P. van den Bossche, J. Van Mierlo, N. Omar, Random forest regression for online capacity estimation of lithium-ion batteries, Appl. Energy 232 (2018) 197–210.
[39] Y. Zhang, Z. Peng, Y. Guan, L. Wu, Prognostics of battery cycle life in the early-cycle stage based on hybrid model, Energy 221 (2021).