Energy 225 (2021) 120205

Contents lists available at ScienceDirect

Energy

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

Early prediction of battery lifetime via a machine learning based framework

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

[a] School of Data Science, City University of Hong Kong, Hong Kong
[b] Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, United States

ARTICLE INFO

Article history:
Received 2 January 2021
Received in revised form
15 February 2021
Accepted 23 February 2021
Available online 27 February 2021

Keywords:
Lithium-ion battery
Battery lifetime prediction
Feature extraction
Feature selection
Machine learning

ABSTRACT

Accurately predicting the lifetime of lithium-ion batteries in early cycles is crucial for ensuring the safety and reliability, and accelerating the battery development cycle. However, most of existing studies presented poor prediction results for early prediction, due to the nonlinear battery capacity fade with negligible variation in early cycles. In this paper, to achieve an accurate early-cycle prediction of battery lifetime, a comprehensive machine learning (ML) based framework containing three modules, the feature extraction, feature selection, and machine learning based prediction, is proposed. First, by analysing the evolution pattern of various informative parameters, forty-two features are manually crafted based on the first-100-cycle charge-discharge raw data. Second, to manage feature irrelevancy and redundancy, four typical feature selection methods are adopted to generate an optimal lower-dimensional feature subset. Finally, the selected features are fed into six representative ML models to effectively predict the battery lifetime. Numerical experiments and paired *t*-test are conducted to statistically evaluate the performance of the proposed framework. Results show that the wrapper-based feature selection method outperforms other methods, and significantly improves the prediction performance of subsequent ML models. Both before and after wrapper feature selection, the elastic net, Gaussian process regression, and support vector machine present better performance than other complex ML prediction models. The support vector machine model combined with wrapper feature selection statistically presents the best result for battery lifetime prediction, with a root-mean-square-error of 115 cycles, and a R^2 of 0.90. Finally, when compared with an existing work, the root-mean-square-error is substantially decreased from 173 to 115 cycles, by using the proposed framework.

1. Introduction

Lithium-ion batteries exhibit low-cost, long-lifetime, and high energy-density characteristics [1], and have thus been widely applied as power sources in many scenarios, such as in smartphones, laptops and electric vehicles [2]. In addition, lithium-ion batteries play an important role in optimising the operation cost of energy storage systems in smart grids and microgrids [3,4]. However, as a result of their internal electrochemical reactions and external environmental conditions, the performance of lithium-ion batteries gradually degenerates over time and with use, which increases the economic cost of the replacement of energy storage systems, such as in smart grids, and may even result in major accidents, such as explosions and consequent bodily injuries. Therefore, to optimise the operation and ensure the safety and reliability of energy storage systems, accurate battery lifetime prediction is of great importance. Moreover, accurate lifetime prediction in early cycles would significantly enhance battery development, production, and optimisation. For example, in an emerging application of high-throughput battery optimisation such as multistep fast-charging, early prediction of battery lifetime could help reduce the number of testing cycles required for each cell, and thereby accelerate the battery optimisation process [5].

A variety of methods have been proposed for battery lifetime prediction in recent years, which is also studied as remaining useful life (RUL) prediction in much literature. Studies in this area can generally be grouped into two categories, model-based methods and machine learning (ML) based methods. Model-based methods

* Corresponding author.
E-mail addresses: zicheng.fei@my.cityu.edu.hk (Z. Fei), fangfyang2-c@my.cityu.edu.hk (F. Yang), kltsui@vt.edu (K.-L. Tsui), lishuai.li@cityu.edu.hk (L. Li), zijzhang@cityu.edu.hk (Z. Zhang).

https://doi.org/10.1016/j.energy.2021.120205

typically build mathematical models to capture the battery degradation dynamics, and then apply various adaptive filter techniques, such as Kalman filter [6] and particle filter [7], to constantly update the model parameters for online lifetime prediction. Some existing studies proposed empirical models to assume simple mathematical relationships between the battery capacity and ageing cycles, without any consideration of interior electrochemical mechanisms of batteries. The most commonly used empirical models are linear [8], polynomial [9], exponential [10], logarithmic [11], and hybrid [12] models. In contrast, semi-empirical models incorporate electrochemical principles into the battery capacity degradation model, and thus provide a good trade-off between battery ageing mechanisms and empirical models. The primary ageing mechanisms of lithium-ion batteries include solid-electrolyte interphase (SEI) formation and growth, lithium plating, loss of active material, and increased impedance [13,14]. Tian et al. [15] established three semi-empirical models to depict three ageing modes of lithium-ion batteries, i.e., state of health (SOH) decline, ohmic internal resistance (IR) growth and polarization resistance growth, which were quantified by an open circuit voltage model. Broussely et al. [16] proposed a square-root-of-time model to describe battery capacity fade and they found that the SEI film thickness is proportional to the square root of time. Yang et al. [7] devised a coulombic efficiency (CE)-based model to capture the battery capacity fade based on the relationship between CE and battery degradation rate. These methods could then perform online prediction of battery lifetime after incorporated with adaptive filter techniques. Although model-based methods have presented good predictive effects, there are still some limitations. First, the lifetime prediction accuracy is largely dependent on the underlying degradation models. Second, model-based methods usually achieve higher prediction accuracy when a larger amount of data along the degradation trajectory is updated for prediction. However, it is challenging for them to accurately predict battery lifetime at the early stages of degradation because the capacity degradation is typically nonlinear and exhibits negligible variation in early battery cycles [5].

ML methods have shown considerable promise on battery prognostics studies. They are mechanism-free and do not require an explicit mathematical model to describe the complex nonlinear ageing dynamics of lithium-ion batteries. They consider the battery as a black-box system and map a set of extracted battery features to the battery lifetime through various ML models. Severson et al. [5] extracted features from discharge voltage and capacity curves, and then applied the elastic net model to predict battery lifetime in early cycles. Zhang et al. [17] obtained features from observations of the discharge capacity, terminal voltage, discharge current, and internal resistance, and then employed a neural network (NN) to predict battery lifetime. Xu et al. [18] designed features based on discharge temperature and voltage curves, and used a stacked denoising autoencoder to predict battery lifetime. Yang et al. [19] extracted features from the voltage, capacity and temperature profiles, and applied a gradient boosting regression tree (GBRT) model to predict battery lifetime. Aforementioned studies regarded the battery lifetime or the RUL value as a direct output of ML models. However, most ML-related studies first estimate the battery health indicators, such as the SOH, capacity, etc., and then predict the RUL by further calculations, such as by extrapolating the degradation model to a failure threshold. Li et al. [20] extracted features from partial incremental capacity curves, and applied a Gaussian process regression (GPR) model to estimate the battery SOH. Shu et al. [21] identified features from partial charging voltage curves, and utilised a least squares support vector machine (SVM) model to estimate the battery SOH. Yang et al. [22] extracted four features from constant-current charging curves, and used an improved GPR model to estimate the battery SOH. Li et at [23] used the battery charging time, discharging time, and discharge capacity as features, and applied a gated recurrent unit-recurrent neural network to estimate the battery SOH. Wu et al. [24] sampled the battery terminal voltages during the charging process as features, and applied a three-layer feed forward NN to predict the current battery cycle number. Ma et al. [25] used the battery capacity in a specific window as features, and designed a hybrid neural network that combined a convolutional neural network and long short-term memory to estimate the capacity in the next window. In these studies, the RULs were then predicted as the estimated SOH/cycle number/capacity reached the failure threshold.

These existing methods have demonstrated satisfactory performance in the battery lifetime prediction; however, they usually perform battery lifetime predictions after accumulating 25% long-term historical degradation data along the failure trajectory [26], which is not beneficial to the rapid battery development and validation. Accurate battery lifetime prediction in early cycles can effectively speed up the development and optimisation of lithium-ion batteries; yet, it is a challenging task due to the typically nonlinear battery ageing dynamics with significantly less degradation in early cycles. In this paper, based on using a dataset released by a group from Massachusetts Institute of Technology (MIT dataset) [5], we focus on studying the early-cycle battery lifetime prediction with only the first-100-cycle degradation data, at which point most cells have not shown any signs of capacity degradation yet.

Moreover, existing studies were mostly performed with extremely small training sets ($\leq$10 samples in most cases), as it is rather time-consuming to acquire a large number of battery degradation samples [19]. And most of them make lifetime predictions by inputting only one or a very few features into ML models. However, as we know, the predictive performance of ML models is influenced primarily by the quantity of training samples and the quality of features. Therefore, it is necessary to explore and exploit more potentially effective features for a more accurate early prediction of battery lifetime, especially when more training samples are available (124 battery samples in MIT dataset). In this paper, a total of 42 features are proposed, which are manually crafted by analysing the battery ageing dynamics based on various parameters, such as the voltage, capacity, temperature, IR, etc. Yet, it is notable that as the feature space enlarges, some features may be redundant or irrelevant and therefore lead to unwanted model overfitting. To avoid this issue, a feature selection procedure is necessary to reduce the feature dimension before predictions are made. Therefore, in this paper, four mainstream feature selection methods are investigated and analysed comparatively to find an optimal lower-dimensional feature subset. For ML prediction models, the diverse intrinsic principles render them suitable for handling different kinds of prediction problems. For example, the SVM is applicable to small-sample-sized problems, while a large number of samples are required to properly train a NN [27]. Hence, an investigation and comparison of various ML algorithms for their ability to deal with the battery lifetime prediction problem is warranted. In this paper, six typical ML models are studied and discussed comparatively to demonstrate the best method for the battery lifetime prediction.

Based on former discussions, in this paper, a comprehensive ML-based framework involving three stages, the feature extraction, feature selection, and ML-based prediction, is proposed for an early prediction of battery lifetime. The MIT dataset containing 124 battery degradation samples with lifetimes ranging from 150 to 2300 cycles is investigated. In the proposed framework, first, a total of 42 features are manually extracted based on battery degradation data from the first 100 cycles, and are grouped into charge-related, discharge-related, capacity-related, temperature-related, and IR-related features to reflect the battery ageing dynamics from different perspectives. Second, to manage feature irrelevancy and redundancy, four typical feature selection methods are investigated, namely filter, wrapper, embedded, and principal component analysis (PCA) methods, to generate an optimal lower-dimensional feature subset. Third, by feeding the selected feature subset, six representative ML algorithms involving the elastic net, GPR, SVM, random forest (RF), GBRT, and NN, are investigated for effectively predicting the battery lifetime. Various comparative experiments are conducted on the MIT dataset to demonstrate the effectiveness of the proposed framework. To capture the pattern of most cell samples, a stratified random sampling method is used to split the dataset, and this is repeated 20 times to reduce the randomness of experimental results. Results show that the wrapper-based feature selection method outperforms other feature selection methods, and would enhance the predictive ability of subsequent ML prediction models. Moreover, the SVM model integrated with the wrapper-based feature selection is the most effective method for battery lifetime prediction, with a root-of-mean-square-error of 115 cycles, a mean-absolute-percentage-error of 8.0%, and a R^2 of 0.90. And this result is significantly better than the prediction result of using the MIT method.

The paper is organised as follows: In Section 2, the experimental data are introduced, and the feature extraction process is elaborated. In Section 3, four types of feature selection methods and six ML models are introduced and discussed. And then, a comprehensive ML-based framework for battery lifetime prediction is presented. In Section 4, the experimental results of battery lifetime prediction are demonstrated. In Section 5, the paper is concluded.

2. Experimental data and feature extraction

2.1. Data description

The MIT dataset, which is currently the largest publicly available dataset for long-term battery degradation studies, is used in this paper. It is composed of 124 commercial lithium iron phosphate/graphite A123 APR18650M1A cells with a nominal capacity of 1.1 Ah. In a temperature chamber at 30 °C, all cells were charged with a multi-step fast charging policy and then followed with a constant-current discharging. As illustrated in Fig. 1, cells were first charged with a current C1 until the state-of-charge (SOC) reached S1 and next charged with a current C2 until the SOC reached S2, which is 80% for all cells. Then, cells were charged from 80% to 100% SOC with 1 C-rate constant current-constant voltage (CC-CV) charging to the cut-off voltage, 3.6 V. These cells were subsequently discharged with a constant current of 4 C-rate to 2.0 V. All of signals (e.g., voltage, capacity, current) within a cycle or on a life-cycle basis are measured and recorded continuously during cycling. The battery lifetime is defined as the cycle number at which point the battery capacity drops to 80% of its nominal capacity. Lifetimes of the 124 cell samples range from 150 cycles to 2300 cycles.

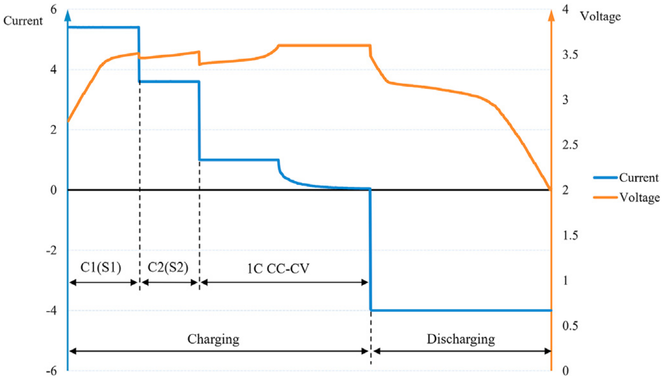

Fig. 1. Schematic diagram of multi-step charging and constant-current discharging.

2.2. Feature extraction

A number of parameters can effectively reflect the ageing dynamics of lithium-ion batteries, such as the voltage, capacity, temperature, IR, SOC, SOH, etc. . In this section, by analysing the evolution pattern of these parameters, 42 features are extracted for battery lifetime prediction. The degradation data from the first 100 cycles are investigated, at which point most cells have yet to exhibit capacity degradation. The 42 features are categorised into five types based on their extraction sources and processes, to reflect the battery ageing dynamic from different perspectives, namely charge-related features, discharge-related features, capacity-related features, temperature-related features and IR-related features, as listed in Table 1. The proposed features are inspired by a range of publications [19,22,28,29] serving various research purposes.

2.2.1. Charge-related features

Fig. 2(a) plots curves of the charging voltage v.s. charging time of a random battery sample from the dataset. These curves are sampled from cycle 10 to 400 with an interval of 50 cycles (In other subfigures of Fig. 2, the same battery sample is taken, and the cycle sampling range and interval are also the same with Fig. 2(a)). In Fig. 2(a), the rising section of the curve corresponds to the CC charging mode, while the flat section represents the CV charging mode. Some patterns reflecting the battery ageing behaviour can be clearly observed. For example, as cycle number increases, the charging curve moves towards top left as annotated in Fig. 2(a), which means the duration of CC mode decreases and the duration of CV mode increases. This is due to the battery polarization phenomenon, which is exacerbated as the battery degrades [31]. Based on this pattern, feature F1 is defined as the time difference of CC mode between cycle 10 and 100. Similarly, F2 is defined as the time difference of CV mode between cycle 10 and 100. On the other hand, F3 and F4 are derived from the charging time, a directly measurable parameter; F3 is the difference in charging time between cycle 10 and 100, while F4 is the average charging time from cycle 10 to 100. Here, the cycle-to-cycle evolution is set as cycle 10 to 100, as was applied in Ref. [5].

2.2.2. Discharge-related features

The discharge-related features are extracted according to the geometrical analysis and summary statistics of the discharge voltage curves and their derivatives, which affords much effective information for battery degradation diagnosis [32]. For example, the incremental capacity (IC) curve, which is calculated by differentiating the change in discharge capacity to the change in voltage within a small voltage step, is a powerful indicator for ageing mechanism identification [33] and online SOH estimation [34]. Features F5, F6 and F7 are extracted from the IC curve. Fig. 2(b) plots the battery IC curves during multiple cycles. Generally, as cycle number increases, the height and voltage of the IC peak decrease simultaneously. Hence, F5 and F6 are defined as the transformation of the height and voltage of the IC peak between cycle 10 and 100, respectively. In addition, F7 is defined as the difference in the covering area of the IC peak between cycle 10 and 100, which is calculated as the variation of the integral of the IC peak region.

Fig. 2(c) shows the $Q(V)$ curves during multiple cycles, where $Q_i(V)$ denotes the discharge capacity curve as a function of the discharge voltage at cycle i. As the cycle number increases, the

Table 1
Features for early prediction of battery lifetime.

Feature Type	No.	Feature Description
Charge-related features	F1	Time duration of CC charging mode, difference between cycle 10 and 100
	F2	Time duration of CV charging mode, difference between cycle 10 and 100
	F3	Charging time, difference between cycle 10 and 100
	F4	Average charging time from cycle 10 to 100
Discharge-related features	F5	Height of IC peak, difference between cycle 10 and 100
	F6	Voltage of IC peak, difference between cycle 10 and 100
	F7	Covering area of IC peak, difference between cycle 10 and 100
	F8	Time duration of discharging process, difference between cycle 10 and 100
	F9	Variance of $\Delta T_{100\text{-}10}(V)$
	F10	Minimum of $\Delta T_{100\text{-}10}(V)$
	F11	Mean of $\Delta T_{100\text{-}10}(V)$
	F12	Maximum of $\Delta T_{100\text{-}10}(V)$
	F13	Skewness of $\Delta T_{100\text{-}10}(V)$
	F14	Kurtosis of $\Delta T_{100\text{-}10}(V)$
	F15	Variance of $\Delta Q_{100\text{-}10}(V)$
	F16	Minimum of $\Delta Q_{100\text{-}10}(V)$
	F17	Mean of $\Delta Q_{100\text{-}10}(V)$
	F18	Maximum of $\Delta Q_{100\text{-}10}(V)$
	F19	Skewness of $\Delta Q_{100\text{-}10}(V)$
	F20	Kurtosis of $\Delta Q_{100\text{-}10}(V)$
Capacity-related features	F21	p1 of linear model, fit to the capacity fade curve from cycle 80 to 100
	F22	p2 of linear model, fit to the capacity fade curve from cycle 80 to 100
	F23	p3 of square-root-of-time model, fit to the capacity fade curve from cycle 80 to 100
	F24	p4 of square-root-of-time model, fit to the capacity fade curve from cycle 80 to 100
	F25	p5 of CE model, fit to the capacity fade curve from cycle 80 to 100
	F26	p6 of CE model, fit to the capacity fade curve from cycle 80 to 100
	F27	p7 of CE model, fit to the capacity fade curve from cycle 80 to 100
	F28	Discharge capacity at cycle 100
	F29	Difference between maximum discharge capacity and the discharge capacity at cycle 100
Temperature-related features	F30	Temperature peak value during charging process, difference between cycle 2 and 100
	F31	Time of reaching temperature peak during charging process, difference between cycle 2 and 100
	F32	Temperature peak value during discharging process, difference between cycle 2 and 100
	F33	Time of reaching temperature peak during discharging process, difference between cycle 2 and 100
	F34	Integral of temperature over time, difference between cycle 2 and 100
	F35	Minimum temperature, difference between cycle 2 and 100
	F36	Average temperature, difference between cycle 2 and 100
	F37	Maximum temperature, difference between cycle 2 and 100
	F38	Mean of minimum temperature from cycle 2 to 100
	F39	Mean of average temperature from cycle 2 to 100
	F40	Mean of maximum temperature from cycle 2 to 100
IR-related features	F41	Internal resistance, difference between cycle 2 and 100
	F42	Average internal resistance from cycle 2 to 100

Note: CC = constant current; CV = constant voltage; IC = incremental capacity; CE = coulombic efficiency; IR = internal resistance; $\Delta T_{100\text{-}10}(V)$ = discharge time as a function of discharge voltage, difference between cycle 10 and 100; $\Delta Q_{100\text{-}10}(V)$ = discharge capacity as a function of discharge voltage, difference between cycle 10 and 100.

discharge capacity and the covering area of the voltage curve tends to decline. In physical terms, the integral of the region between $Q_i(V)$ and $Q_j(V)$ curves depicts the discharge energy dissipation between cycle i and j. Therefore, the cycle-to-cycle evolution $\Delta Q_{i-j}(V)$, denoted as $Q_i(V) - Q_j(V)$, is informative for battery lifetime prediction. Here, features F15–F20 are defined as the summary statistics, comprising the variance, minimum, mean, maximum, skewness, and kurtosis of $\Delta Q_{100-10}(V)$, denoted as $Q_{100}(V) - Q_{10}(V)$. For the computational formulas of these summary statistics, please refer to Ref. [5].

Similarly, F8–F14 are extracted from the cycle-to-cycle evolution of the $T(V)$ curve, which is the discharge time as a function of discharge voltage. Feature F8 is defined as the difference in the discharge time between cycle 10 and 100. And F9–F14 are defined as the summary statistics comprising the variance, minimum, mean, maximum, skewness and kurtosis of $\Delta T_{100-10}(V)$. Although the discharge time is linearly correlated with the discharge capacity, features F8–F14 may provide some new information for battery lifetime prediction based on features F15–F20, because different kinds of summary statistics are used here to perform nonlinear transformations on the original data.

For discharge-related features, the cycle-to-cycle evolution is also set to be for cycle 10 to 100, as was used in Ref. [5].

2.2.3. Capacity-related features

Capacity evolution is an intuitive indicator of the battery degradation behaviour. Fig. 3 displays the capacity fade of the 124 cell samples. In this section, three common models are considered, namely the linear model, the square-root-of-time model [16], and the CE model [7], all of which have been widely used for modelling the degradation dynamics of lithium-ion batteries. Here, these three models are successively fitted to the capacity degradation curves, and then the corresponding fitting parameters of three models are defined as the capacity-related features. Details of the three models and corresponding fitting parameters are described as below.

- *Linear model*: the simplest degradation model, which is given by the following equation (Eq. (1)):

$$C_l = p_1 \cdot l + p_2 \tag{1}$$

- *Square-root-of-time model*: a typical semi-empirical model, which models the degradation of a lithium-ion battery by considering the loss of its lithium inventory. This is caused by

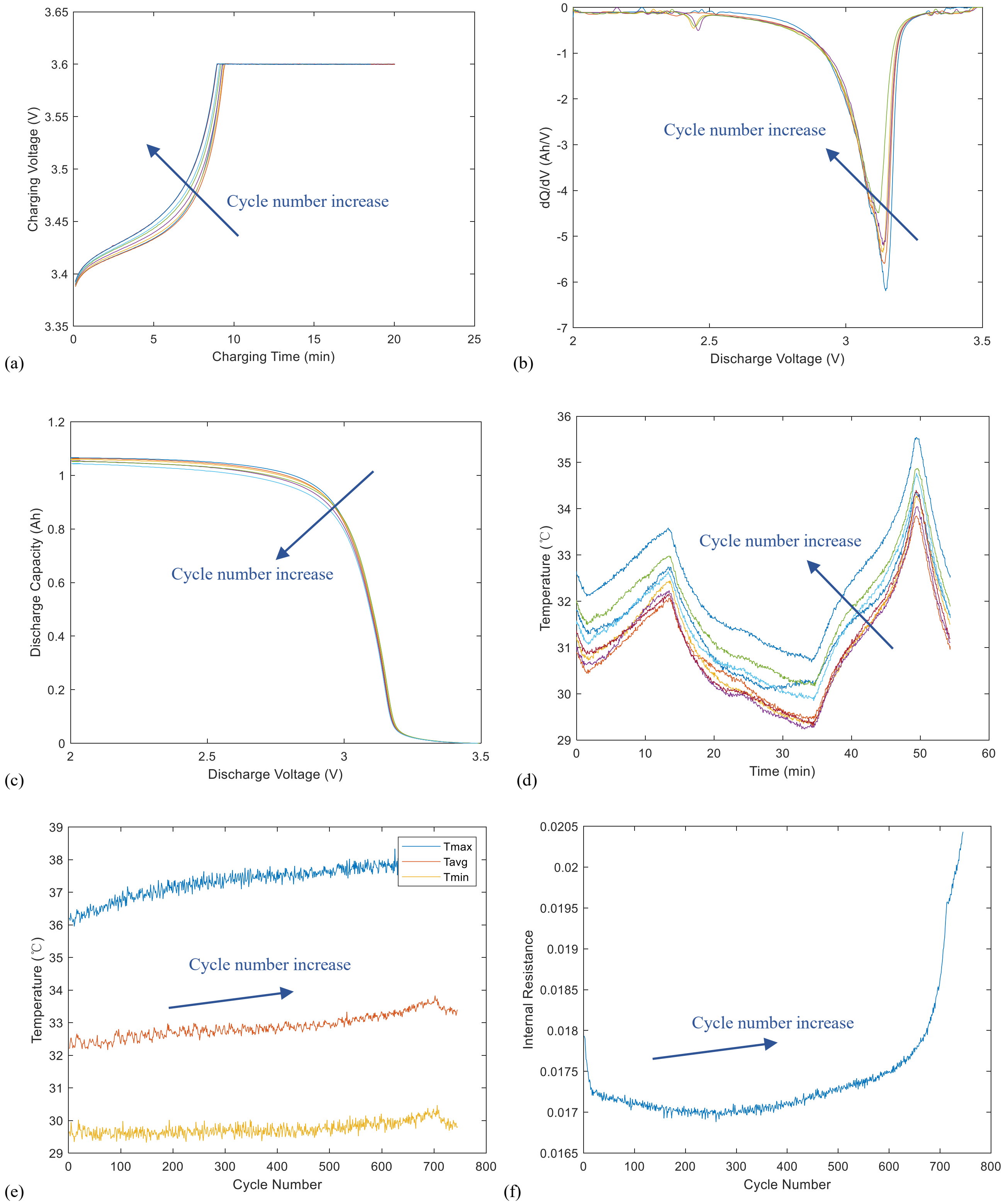

Fig. 2. Curves of a random battery sample: (a) Charging curves during multiple cycles; (b) Incremental capacity (IC) curves during multiple cycles; (c) Discharge capacity vs. discharge voltage ($Q(V)$) curves during multiple cycles; (d) Temperature vs. time within a cycle; (e) Temperature vs. cycle number in the life-cycle dimension (where Tmax, Tavg and Tmin = the maximum, average and minimum surface temperatures, respectively); (f) Internal resistance vs. cycle number in the life-cycle dimension.

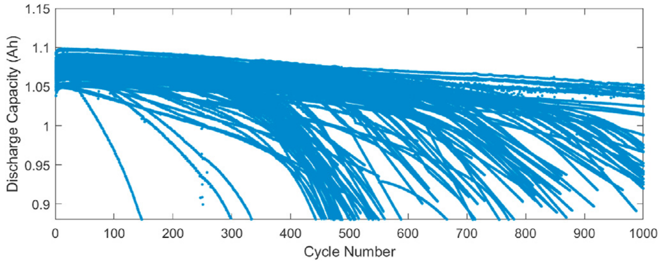

Fig. 3. Discharge capacity curves of 124 cell samples in the MIT dataset.

the thickening of the SEI film, as the film thickness is proportional to the square root of time, as given by Eq. (2), as follows:

$$C_l = p_3 \cdot \sqrt{l} + p_4 \tag{2}$$

- *CE model*: a semi-empirical model that models the battery capacity degradation by capturing the relationship between CE and the battery degradation rate, given by the following equation (Eq. (3)):

$$C_l = p_5 \cdot p_6^l + p_7 \tag{3}$$

In Eqs. (1)–(3), C_l is the discharge capacity at cycle l, and p_1-p_7 are seven fitting parameters of the three ageing models fitted to the capacity degradation curves. Parameters p_1-p_7 are defined as features F21–F27, respectively. For example, feature F21 is the slope "p_1" and F22 is the intercept "p_2" of the linear fitting to the capacity fade curves from cycle 80 to cycle 100. Here, the cycle range is set as cycle 80–100, because we find that as the starting point of the cycle range approaches cycle 100, capacity-related features better capture the nonlinearity of the capacity fade curve and thereby become more predictive for the battery lifetime.

F28 is directly defined as the discharge capacity at cycle 100, and F29 is calculated as the difference between the maximum discharge capacity and the capacity at cycle 100.

2.2.4. *Temperature-related features*

Temperature is another significant indicator that reflects the battery ageing dynamic. Fig. 2(d) displays the temperature profile during multiple cycles. Two peaks can be observed in Fig. 2(d): the left peak is for the charging process while the right peak is for the discharging process. The battery temperature rises significantly during the fast-charging and discharging processes. Moreover, as the cycle number increases, both temperature peaks increase in height, and the time required to reach the peak decreases, which results from the reduction in the electrolyte conductivity and the lithium ion diffusivity [35]. Based on this pattern, F30 and F32 are defined as the difference in the temperature peak value between cycle 2 and 100 during the charging and discharging processes, respectively. Similarly, F31 and F33 are defined as the difference in the time taken to reach the temperature peak between cycle 2 and 100, during the charging and discharging processes, respectively. F34 is defined as the difference in the temperature integral over time between cycle 2 and 100. F35–F37 are defined as the difference in the minimum, average, and maximum surface temperatures between cycle 2 and 100, respectively.

Fig. 2(e) displays the variation of the maximum, average and minimum surface temperatures in the life-cycle dimension. As the battery degrades, these three measurements of surface temperatures generally increase as the cycle number increases. Therefore, F38–F40 are defined as the means of the minimum, average, and maximum surface temperatures from cycle 2 to 100, respectively. Here, the cycle-to-cycle evolution is set as cycle 2 to 100, which was also used in Ref. [5].

2.2.5. *Internal resistance-related features*

Battery degradation is closely related to the increase in IR. Fig. 2(f) displays the evolution of the IR in the life-cycle dimension. A declining trend can be observed in early cycles and then a rising trend in later cycles. Based on this pattern, F41 is defined as the difference in the IR between cycle 2 and 100, while F42 is defined as the average IR from cycle 2 to 100. Here, the cycle-to-cycle evolution is set as being for cycle 2 to 100, for the same reason as given in Section 2.2.4.

3. Machine learning-based framework for battery lifetime prediction

In this section, a comprehensive ML-based framework is presented for the early-cycle lifetime prediction of lithium-ion batteries. Three main modules constitute the structure of the framework, namely feature extraction, feature selection, and ML-based prediction. In Section 2.2, feature extraction is discussed. In Sections 3.1 and 3.2, feature selection and ML-based prediction are elaborated.

3.1. *Feature selection*

In Section 2.2, a full set of 42 features are generated. However, the problem we target is a small-sample-sized application, with an experimental dataset of only 124 battery samples in total. Excessive features may lead to the overfitting of prediction models. Moreover, some of the extracted features may be redundant or irrelevant and would thus not improve the model performance, thereby resulting in a waste of memory and computational costs. Given this situation, it is necessary to conduct feature selection on the full feature set to generate an optimal feature subset.

Concerning various selection strategies, feature selection methods can generally be categorised into three types, namely filter, wrapper, and embedded methods [36]. In this section, a representative method in in each category is investigated to generate a lower-dimensional feature subset to feed into prediction models. The flowchart of each feature selection method category is shown in Fig. 4. In addition, PCA, as a classical dimension reduction technique, has been widely used in battery research [37–39] to reduce the dimension of feature space, so we include PCA too.

3.1.1. *Filter method*

Filter feature selection methods (Fig. 4(a)) rely on ranking techniques to assess feature importance. Here, the Pearson correlation coefficient is investigated as a possible ranking criterion, as it is the most commonly used ranking criterion in filter methods [40]. It measures the linear correlation between the feature x and the target y, defined as below (Eq (4):

$$r_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{4}$$

where x_i and y_i denote the i-th sample of feature x and the target y, and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean, and analogously for $\bar{y}$.

There are two steps in the filter feature selection process. First,

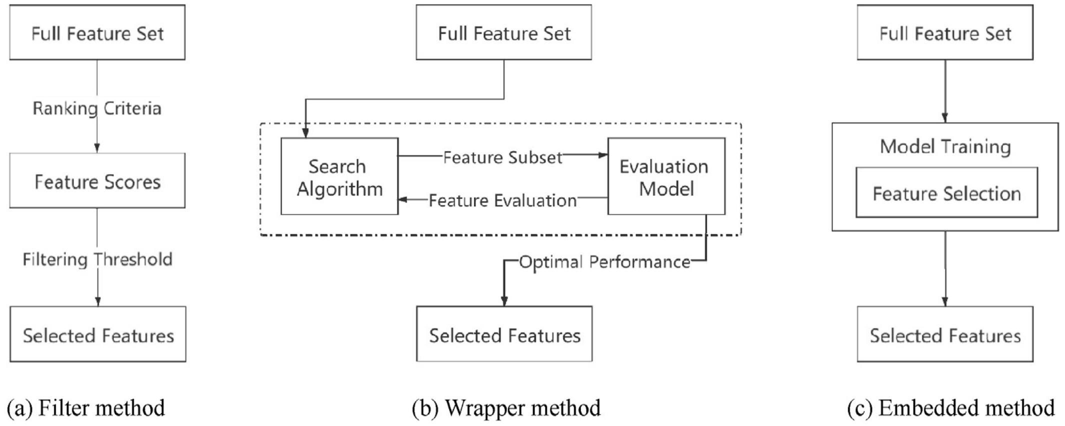

Fig. 4. Flowcharts of three categories of feature selection methods.

the Pearson correlation coefficient between each feature and the target (battery lifetime) is calculated, and a score is assigned to each feature. Second, a threshold is set to filter out the lowly-ranked features. Here, we set the threshold as 20, meaning that the top 20 features based on the Pearson-correlation ranking are selected and then fed into the ML-based lifetime prediction models.

3.1.2. *Wrapper method*

The wrapper method (Fig. 4(b)) considers the feature selection process as a search problem, and finds an optimal feature subset that produces the best predictive performance in a predefined ML prediction model.

For the searching process, we adopt a typical heuristic search algorithm, the genetic algorithm (GA), which searches for different feature subsets by finding the optimal solution of optimisation problems. Compared with some widely used deterministic search algorithms, such as the sequential backward selection algorithm, the GA has the advantage of finding the global optimum by introducing randomness into the searching process [41]. For the predefined ML prediction model, we adopt the GBRT model, because the ML model used in wrapper methods should be capable of providing the feature importance. The details of the GA can be found in Ref. [41], and the principle of the GBRT model will be explained in Section 3.2.5.

There are two steps in the wrapper feature selection process. First, the GA algorithm is used to search through the feature space and generate a possible subset of features. Then, the quality of selected features is evaluated based on the GBRT model, which here acts as a black box. These two steps are repeatedly performed until the best predictive performance is achieved. By using the GA and the GBRT, 12 features are automatically searched out and then fed into ML-based lifetime prediction models.

3.1.3. *Embedded method*

The embedded feature selection method is a trade-off between the filter and wrapper methods, as it performs feature selection as part of the model training process, as shown in Fig. 4(c). Regularisation models, such as the lasso and the elastic net, are the most widely used embedded methods [36]. Here, we adopt the elastic net as an exemplar embedded method. It introduces both L1 and L2 norm penalties to shrink the coefficients of certain features to zero, thereby realising feature selection. A higher coefficient in the regularised linear model indicates a more important feature. The detail of the elastic net is further discussed in Section 3.2.1.

3.1.4. *Principal component analysis*

PCA is a well-known unsupervised dimension reduction algorithm that is also included within the scope of feature selection in some literature [37,38]. PCA uses a projection onto a unique orthogonal basis to map the original feature space onto a lower-dimensional principal component set, where most of the variability of the original data is efficiently preserved. Features that contribute little to the variance are removed during this process. The variability is measured by the cumulative explainable variance, which limits the number of principal components to be retained. Here, we set the cumulative explainable variance of PCA as 99%, and then derive a six-dimensional feature subset to feed into ML-based lifetime prediction models.

Unlike the abovementioned three methods that simply pick out certain features from the full set without modifying them, the projection transformation performed in PCA alters the information within features, such that the generated features are not easily interpretable.

3.2. *Machine learning-based prediction*

By inputting the lower-dimensional feature subset derived from various feature selection methods, six representative ML models, including the elastic net, GPR, SVM, RF, GBRT, and NN are investigated for their application to battery lifetime prediction. Given a training set $\{(x_i, y_i)\}_{i=1}^{n}$, n is the number of training samples. The input of these ML models is$X = \{x_i\}_{i=1}^{n}$, an $n \times p$ matrix of features after feature selection, while the output of the ML models is y, a n-dimensional vector of battery lifetimes.

3.2.1. *Elastic net*

The elastic net is a commonly used regularised linear regression model [5] that incorporates both L1-norm and L2-norm regularisation, which are also known as lasso and ridge regularisation, respectively. The elastic net takes the following form (Eq. (5)):

$$\theta = \mathrm{argmin}_{\theta}\left\{\|\widehat{y} - X\theta\|_2^2 + \lambda\left(\frac{1-\alpha}{2}\|\theta\|_2^2 + \alpha\|\theta\|_1\right)\right\} \tag{5}$$

where the argmin function aims to find the value of θ that minimises the argument. The first term inside the square bracket is a form of ordinary least squares, $\widehat{y}$ is an n-dimensional vector of observed battery lifetimes, X is an $n \times p$ matrix of features, and θ is an $n \times 1$ vector of model coefficients.

The second term is the regularisation term, containing two non-negative hyperparameters of the elastic net: λ and α (as shown in

Table 2), where λ is the regularisation parameter and α (ranging from 0 to 1) determines the relative importance of the L1 and L2 norm penalties. The case of $\alpha = 1$ corresponds to the lasso regression, in which the coefficients of certain features are drawn to zero and therefore can simultaneously perform feature selection and prediction. The case of $\alpha = 0$ corresponds to the ridge regression, which shrinks weights of features closer to but not exactly equal to zero. For a value of αbetween 0 and 1, the elastic net combines both selection and shrinkage.

3.2.2. *Gaussian process regression*

The GPR has been widely used in battery prognostic research, because it is flexible, nonparametric and probabilistic [42]. It is a kernel-based ML method that performs prediction tasks by incorporating prior knowledge and obtaining a hypothesis of posterior probability via a Bayesian framework. For the input x, the prior probability distribution over function $f(x)$ follows a Gaussian distribution. Then, the property of $f(x)$ can be fully described by its mean function $m(x)$, and its covariance function $k(x, x')$. The key idea is that, rather than assuming a parametric form of the function $f(x, \theta)$ and estimating the parameters θ, the GPR calculates the joint probability distribution over all of the admissible $f(x)$ sampled from the Gaussian process. Here, the mean function $m(x)$ is usually set as 0 [42]. For the covariance function $k(x,x')$, we adopt the commonly used squared exponential kernel, as shown in Equation (6), as follows:

$$k_{ij}(x_i, x_j) = \sigma_f^2 \exp\left(\frac{-\|x_i, x_j\|^2}{2\sigma_l^2}\right) \tag{6}$$

where, σ_f, the signal standard deviation, and σ_l, the characteristic length scale, are two hyperparameters to be tuned in the GPR (see Table 2).

The prior distribution of the training samples can be expressed as Equation (7), as follows:

$$y \sim N\left(0, K(x_i, x_j) + \sigma_n^2 I_n\right) \tag{7}$$

where y is a vector of observed battery lifetimes, x denotes the input features, $K(x_i, x_j) = (k_{ij})_{n\times n}$ is a n-dimensional symmetric positive definite matrix, and $\sigma_n^2 I_n$ is the noise covariance matrix.

Then, the two hyperparameters are determined using a maximum likelihood method [22]. For the testing samples, based on the joint prior distribution with training samples, the posterior distribution of y can be obtained by using Bayesian theory. The mean value of the posterior y is the predicted battery lifetime.

Table 2
Hyperparameters used for the six machine learning models.

Elastic net	$\alpha \in \{0, 0.1, \ldots, 0.9, 1\}$
	$\lambda \in (10^{-3}, 10^{-1})$
Gaussian process regression	$\sigma_f \in (10^{-4}, 10^{1})$
	$\sigma_l \in (10^{-3}, 10^{3})$
Support vector machine	$C \in (10^{-5}, 10^{5})$
	$\sigma \in (10^{-2}, 10^{2})$
Random forest	$\#Trees \in (1, 1800)$
	$MaxNumSplits \in (2^0, 2^6)$
	$MinLeafSize \in (2^0, 2^5)$
Gradient boosting regression tree	$\#Trees \in (1, 1800)$
	$MaxNumSplits \in (2^0, 2^6)$
	$Learningrate \in (0.01, 0.5)$
Neural network	$\#Hiddenlayers \in \{1, 2, 3\}$
	$\#Neuronsineachlayer \in (2^0, 2^7)$
	$\mathrm{max}Epochs = 300$
	$\mathrm{min}BatchSize = 20$
	$Learningrate \in (0.01, 0.1)$
	$AdamPara. = Default$

3.2.3. *Support vector machine*

The SVM is a classical nonparametric ML algorithm based on kernels. Although mainly used in classification problems, the SVM is capable of solving regression problems, also known as support vector regression (SVR) [43]. It performs nonlinear prediction task by transforming the problem in low-dimensional space into a linear problem in high-dimensional feature space through a nonlinear mapping $\phi(\cdot)$, as formulated in Eq. (8):

$$y = w\phi(x) + b \tag{8}$$

where y is the predicted battery lifetime, and w, x, and b denote the coefficients, input features and intercept, respectively.

The aim of SVR is to find an ε-insensitive error function such that the maximum deviation of y in the training data is lower than a predefined threshold ε, and simultaneously maintain the largest possible smoothness of the function. This is formulated as a convex optimisation problem, as shown in Eq. (9):

$$\min J(w, b, \xi) = \frac{1}{2} w^{\mathrm{T}} w + C \sum_{i}^{n} \left(\xi_i^1 + \xi_i^2\right)$$

$$\text{s.t.} \begin{cases} y_i - (w\phi(x_i) + b) \le \varepsilon + \xi_i^1 \\ (w\phi(x_i) + b) - y_i \le \varepsilon + \xi_i^2 \\ \xi_i^1, \xi_i^2 \ge 0 \end{cases} \tag{9}$$

where C is a regularisation parameter, ξ_i^1 and ξ_i^2 are two slack variables.

By introducing Lagrange multipliers, the Lagrange dual problem of (9) can be solved. Then, the optimal solution of w and b for the primal problem (9) can be deduced according to the duality [44]. Finally, the SVR expression of the nonlinear prediction can be formulated as Eq. (10):

$$y = w\phi(x) + b = \sum_{i=1}^{n} K(x_i, x_j)(\alpha_i - \beta_i) + b \tag{10}$$

where α_i and β_i are Lagrange multipliers, and $K(x_i, x_j) = \phi(x_i)\phi(x_j)$ is the kernel function.

Here, we adopt the most popular radial basis function (RBF) [45] as the kernel for the SVM, such that the hyperparameters to be tuned comprise the kernel scale σ and the box constraint C, as listed in Table 2.

3.2.4. *Random forest*

The RF is a typical ensemble model that aggregates the predictions from multiple trees [46]. The training process of RF for battery lifetime prediction is to construct t different decision trees. For the growth of trees, a random feature subset is adopted in each split of the tree nodes. In addition, each tree grows on a randomly sampled subset of training samples through bootstrap sampling, which randomly selects m samples with replacement from the training set with n observations. By introducing such randomness, the RF is able to increase the diversity of trees and thus capture more patterns in the data. Finally, the predicted battery lifetime is obtained from the RF by averaging the predictions of t trees in the forest.

Three hyperparameters need to be tuned in the RF, namely the number of trees in the forest (t), the maximum number of trees split, and the minimum tree leaf size (see Table 2). The number of features selected at each tree split is set as one-third (the default option in Matlab) of the total feature number.

3.2.5. Gradient boosting regression tree

The GBRT is a tree-based ensemble model that incorporates a statistical technique called boosting [19]. The GBRT aggregates many ‘weak’ trees to comprise a single ‘strong’ tree with better stability than a single complex tree. During the training process of the GBRT, a new simple tree is constructed in each step S to compensate for the prediction residuals of previous S-1 simple trees, thereby minimising the loss function. Here, the commonly used least squares loss function is adopted [47]. Additionally, the prediction result of each tree is shrunken by a factor α (also known as the learning rate) to help prevent the model from overfitting. This tree-building process, which is fundamentally a form of functional gradient descent, is iterated until there are t trees in the GBRT ensemble. Finally, predictions from t base trees are aggregated together by a weighted sum to form the lifetime output of the GBRT.

For the GBRT, three hyperparameters need to be tuned: the number of trees t, the maximum number of trees split, and the learning rate α (see Table 2).

3.2.6. Neural network

NNs have been popular in a wide variety of applications, including in battery prognostic research [27]. The battery lifetime prediction in this work is a small-sample-sized application, so here we investigate an NN with traditional architecture: a feedforward neural network (FFNN) with a shallow structure of no more than three hidden layers. The FFNN contains three kinds of layers: an ‘input layer’ of input feature X, one or more ‘hidden layers’ that realise the interactions and nonlinear transformation of inputs, and at last an ‘output layer’ that merges the results of hidden layers into y, a final output of the network. In the hidden layer, the commonly used *tanh* activation function is used at all of the nodes to realise the nonlinear transformation [26].

For the NN, the hyperparameters to be optimised are the number of hidden layers and the number of neurons in each layer (see Table 2).

3.3. Machine learning-based framework

Based on the above discussion, a comprehensive ML-based framework for early lifetime prediction of lithium-ion batteries is presented, as shown in Fig. 5. Feature extraction, feature selection, and ML-based prediction constitute the main architecture of the framework. First, in Section 2.2, a set of 42 features are extracted from the raw degradation data and then categorised into five groups to reflect the battery ageing dynamic from different angles. Second, in Section 3.1, to deal with feature redundancy and irrelevancy, the full feature set is inputted into four feature selection methods to generate a lower-dimensional feature subset. Third, in Section 3.2, by feeding the selected features, six representative ML algorithms for battery lifetime prediction are investigated. The feature extraction, ML model development, hyperparameter optimisation, and model evaluation are performed on the MIT dataset using MATLAB tools.

In this work, 123 cell samples from the MIT dataset are studied, excluding the shortest-lived sample whose pattern does not match that of the other cells. Notably, due to the presence of long-lived cells and short-lived cells in the original dataset, the data are subject to stratified random sampling, i.e., the data are randomly split, with 70% used as the training set (86 samples) and 30% used as the testing set (37 samples), with equal ratios of long-lived cells to short-lived cells are maintained in each split. To prevent overfitting, the hyperparameter optimisation is conducted via a 5-fold cross-validation on the training set, and the Bayesian search algorithm [48] is used to search for good hyperparameter combinations. Furthermore, to reduce the randomness of our experiments, we repeat the stratified random sampling 20 times, and average the results of the 20 data splits.

The performance of prediction models is evaluated according to the following three metrics (Eqs. 11–13):

- RMSE (root-mean-square error):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{11}$$

- MAPE (mean absolute percentage error):

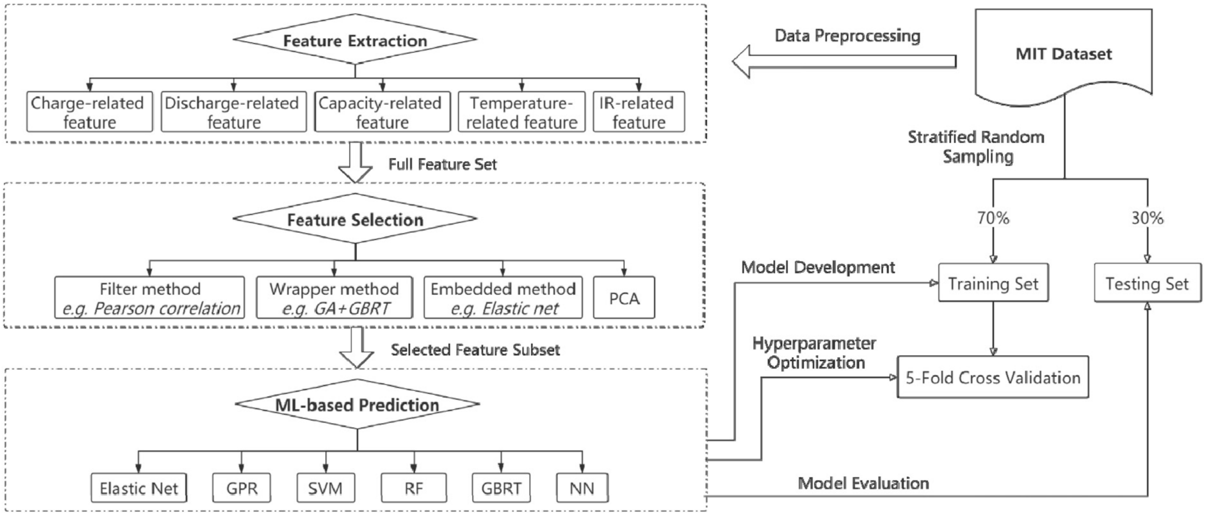

Fig. 5. Architecture of the ML-based framework for battery lifetime prediction.

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|y_i - \hat{y}_i\right|}{y_i} \times 100\% \tag{12}$$

- R^2 (coefficient of determination):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2} \tag{13}$$

where n denotes the number of cell samples, and y_i and $\hat{y}_i$ denote the observed lifetime and the predicted lifetime of cell i, respectively.

The RMSE describes the variations in data errors, and is more sensitive to large errors, while the MAPE depicts the error in terms of percentages and measures the relative error. Thus, smaller values of RMSE and MAPE indicate that a model has better predictive performance. In contrast, a larger value of R^2, which is a percentage-based metric, represents better performance.

4. Results and discussions

4.1. Results - feature selection

In this section, four feature selection methods summarised in Section 3.1 are conducted to select an optimal lower-dimensional feature subset from the full set of 42 features. For the filter method (Pearson correlation), wrapper method (GA + GBRT), and the PCA method, the feature dimension is reduced to 20, 12, and 6, respectively. For the embedded method (the elastic net), since the experiment is repeated 20 times and it adaptively selects different number of features each time, its output dimension is not listed here. To evaluate the effectiveness of these four feature selection methods, the derived lower-dimensional feature subsets are further fed into a GPR model to predict the battery lifetime, and to evaluate the performance of selected features. Here the GPR model is used as the base evaluation model because it presents good predictive power and stability among the aforementioned six ML models, and more importantly, it does not perform embedded feature selection so that the prediction results can be determined only by the selected features without the distraction of ML model. Notably, the embedded method (elastic net) performs feature selection and model prediction simultaneously, so it does not require an additional GPR model for prediction.

Table 3 tabulates the average prediction results on the testing set, by applying full features and four types of feature selection methods. The RMSEs for the full-feature model, the filter method, the wrapper method, the embedded method, and the PCA are 137, 147, 119, 133, and 157 cycles, respectively. The MAPE of each method falls within the range of 8.9%–12.1%, while the R^2 ranges from 0.82 to 0.89. Overall, the wrapper method presents the best performance, with the lowest RMSE and MAPE, and the highest R^2.

To further evaluate the performance before and after feature selection statistically, a paired t-test is conducted to pairwise-compare the unbalanced prediction result of 20 splits. The paired t-test [49] is a statistical technique used to determine whether there is a statistically significant difference between two sets of paired observations. The mean difference between the full-feature model and other selection models and its standard error are listed in Table 3. For all of these methods, the standard errors of RMSE, MAPE, and R^2 are less than 6 cycles, 0.4%, and 0.01, respectively, indicating the stability of the prediction results from 20 splits. Besides, to have a more intuitive view on the comparison results, Table 3 also lists the performance improvement percentage (PIP), calculated as $\pm\left|\frac{P_f - P_0}{P_0}\right| \times 100\%$, where P_f and P_0 denote the predictive performance after and before feature selection, respectively. Here, positive PIPs signify that a method has improved performance than the full-feature method (lower RMSE and MAPE, and higher R^2). A bold PIP indicates that the performance difference is statistically significant at the 5% level (the p-value of the paired t-test is less than the confidence level 0.05).

As shown in Table 3, compared with the full-feature method, the wrapper and embedded methods both improve the predictive performance, and outperform the filter and PCA methods. The wrapper method has positive PIPs of 13.1%, 13.8% and 4.1% for the three metrics, which are all statistically significant at the 5% level. The performance enhancement of the embedded method is not as large as that of wrapper method: it has positive PIPs of 2.7%, 3.8% and 1.1% for the three metrics, which are not statistically significant. These two methods present better performance because they are dependent on the ML model and therefore can produce a superior feature subset for use by the prediction model. Furthermore, they can well handle feature redundancy and irrelevancy. In contrast, the filter and PCA methods both have negative effects on the predictive performance of the full-feature method. These two methods provide worse results because they are model-free methods and therefore lack guidance from ML models during the feature selection process. What's more, they cannot identify and remove highly correlated features, and may even omit some efficient features for prediction.

Among these four methods, the wrapper method (GA + GBRT) statistically presents the best selection results for battery lifetime prediction. Table 4 shows the 12-dimensional feature subset

Table 3
Results of using full-feature model (without feature selection) and four feature selection methods.

Feature selection method (feature dimension)	Average results of 20 splits			Results of paired t-test			Performance improvement percentage (PIP) (%)		
				Mean difference (Standard error)					
	RMSE (cycles)	MAPE (%)	R^2	RMSE (cycles)	MAPE (%)	R^2	RMSE	MAPE	R^2
Full-feature (42)	137	10.3	0.86	/	/	/	/	/	/
Filter (20)	147	11.7	0.84	10 (5)	1.4 (0.4)	−0.02 (0.01)	−7.5%	**−13.5%**	−2.5%
Wrapper (12)	119	8.9	0.89	−18 (6)	−1.4 (0.3)	0.03 (0.01)	**13.1%**	**13.8%**	**4.1%**
Embedded (/)	133	10.0	0.87	−4 (5)	−0.3 (0.3)	0.01 (0.01)	2.7%	3.8%	1.1%
PCA (6)	157	12.1	0.82	20 (5)	1.8 (0.4)	−0.04 (0.01)	**−14.5%**	**−16.9%**	**−4.9%**

Note: The full-feature model is the benchmark for paired t-test. $\mathrm{PIP} = \pm\left|\frac{P_f - P_0}{P_0}\right| \times 100\%$, where P_f and P_0 denote the predictive performance after and before feature selection, respectively. Positive PIPs indicate better performance than the full-feature model. A bold font indicates the performance difference is statistically significant at the 5% level.

Table 4
12-dimensional feature subset after wrapper feature selection.

Type	No.	Feature	Pearson correlation with lifetime
Charge-related features	F1	Time duration of CC charging mode, difference between cycle 10 and 100	0.28
Discharge-related features	F7	Covering area of IC peak, difference between cycle 10 and 100	0.51
	F8	Time duration of discharging process, difference between cycle 10 and 100	0.73
	F9	Variance of $\Delta T_{100\text{-}10}(V)$	0.93
	F17	Mean of $\Delta Q_{100\text{-}10}(V)$	0.91
Capacity-related features	F22	p2 of linear model, fit to the capacity fade curve from cycle 80 to 100	0.23
	F24	p4 of square-root-of-time model, fit to the capacity fade curve from cycle 80 to 100	0.36
Temperature-related features	F34	Integral of temperature over time, difference between cycle 2 and 100	0.39
	F35	Minimum temperature, difference between cycle 2 and 100	0.34
	F38	Mean of minimum temperature from cycle 2 to 100	0.38
	F39	Mean of average temperature from cycle 2 to 100	0.15
IR-related features	F41	Internal resistance, difference between cycle 2 and 100	0.25

derived from this method, which covers features from each of the five categories. To further demonstrate the predictive power of these selected features, their Pearson correlation coefficients with the battery lifetime are listed in Table 4. The most predictive features are F9 (variance of $\Delta T_{100\text{-}10}(V)$) and F17 (mean of $\Delta Q_{100\text{-}10}(V)$), which have correlation coefficients with the battery lifetime of 0.93 and 0.91, respectively.

4.2. Results — machine learning based prediction

In this section, the 12-dimensional feature subset selected from the wrapper method is fed into six ML models for battery lifetime prediction, namely the elastic net, GPR, SVM, RF, GBRT, and NN models.

Table 5 demonstrates the prediction results for all of the ML models. Among these, the NN has the worst performance, as indicated by its having the highest RMSE and MAPE, and the lowest R^2 (206 cycles, 17.3%, and 0.67, respectively). This result is mainly due to the fact that the battery lifetime prediction made here involves a small-sample-sized application with only 86 training samples, whereas the NN needs a large number of samples to effectively train its network to obtain better performance. By excluding the NN, the RMSEs of the other five models are within the range of 115–151 cycles, and the MAPEs range from 8.0% to 11.7% and the R^2 ranges from 0.83 to 0.90. Overall, the SVM presents the best predictive performance, as it has the lowest RMSE and MAPE, and the largest R^2.

To statistically compare the performance within different ML models, a paired *t*-test is conducted and results are summarised in Table 5. Here, the SVM, which experimentally provides the best performance, is chosen as the benchmark method for paired *t*-test. The PIPs of each ML model compared with the benchmark SVM are also listed in Table 5. As shown in Table 5, compared with the SVM, the PIPs of the other five ML models are all negative, indicating their worse performance than the SVM. Among these five models, the performance reduction of the elastic net and the GPR are relatively light. For instance, the PIP of RMSE is only −4.8% for the elastic net and −3.5% for the GPR, which are not statistically significant. In contrast, the performance reduction of the RF, the GBRT, and the NN are large and statistically significant for all metrics. For example, the PIP of RMSE is −31.3% for the RF, −17.4% for the GBRT, and −79.5% for the NN.

Based on the above results, we statistically conclude that the SVM model using features selected by the wrapper method affords the best predictive performance for battery lifetime prediction.

In addition, to further explore the effect of feature selection on ML prediction models, we also conduct another paired *t*-test to compare the model performance before and after feature selection. Table 6 demonstrates the quantitative comparison results while Fig. 6 visually presents the RMSE results. It can be clearly seen from Fig. 6 that the RMSEs of all six ML models decrease after feature selection, indicating that the wrapper-based feature selection strengthens the predictive ability of each ML model. Specifically, for the elastic net, the GPR, the SVM and the NN, the performance improvements for all three metrics are statistically significant. For example, the PIPs of RMSE are 9.6%, 13.1%, 13.5%, and 13.4% for these four models, respectively. In contrast, for the RF and the GBRT, little difference is observed after feature selection.

What's more, both before and after feature selection, the elastic net, the GPR, and the SVM outperform the other three models, as shown in Fig. 6. This may be due to these being relatively simple compared with the other ML models, and thus more generally applicable to our small-sample-sized problem. Moreover, when all 42 features are input, the elastic net, the GPR, and the SVM produce similarly good predictive results, but after feature selection, the SVM becomes the best performing model, which indicates that the

Table 5
Prediction results of six ML models using 12 features derived from the wrapper method.

ML models	Average results of 20 splits			Results of paired *t*-test			Performance improvement percentage (PIP) (%)		
				Mean difference (Standard error)					
	RMSE (cycles)	MAPE (%)	R^2	RMSE (cycles)	MAPE (%)	R^2	RMSE	MAPE	R^2
Elastic net	121	9.1	0.89	6 (3)	1.1 (0.2)	−0.01 (0.01)	−4.8%	**−13.4%**	−1.1%
GPR	119	8.9	0.89	4 (3)	0.9 (0.1)	−0.01 (0.01)	−3.5%	**−10.8%**	−1.1%
SVM	115	8.0	0.90	/	/	/	/	/	/
RF	151	11.7	0.83	36 (6)	3.7 (0.4)	−0.07 (0.01)	**−31.3%**	**−45.8%**	**−7.8%**
GBRT	135	11.0	0.86	20 (5)	3.0 (0.3)	−0.04 (0.01)	**−17.4%**	**−36.8%**	**−4.4%**
NN	206	17.3	0.67	91 (11)	9.3 (0.8)	−0.23 (0.03)	**−79.5%**	**−115.2%**	**−26.0%**

Note: The SVM model is the benchmark for paired *t*-test. PIP $= \pm\left|\frac{P-P_s}{P_s}\right| \times 100\%$, where P_s and P denote the performance of the SVM model and other models, respectively. Negative PIPs indicate worse performance than the SVM model. A bold font indicates the performance difference is statistically significant at the 5% level.

Table 6
Comparison of six ML models before and after wrapper feature selection.

ML models	Before wrapper feature selection			After wrapper feature selection			Results of paired *t*-test			Performance improvement percentage (PIP) (%)		
							Mean difference (Standard error)					
	RMSE (cycles)	MAPE (%)	R^2	RMSE (cycles)	MAPE (%)	R^2	RMSE (cycles)	MAPE (%)	R^2	RMSE	MAPE	R^2
Elastic net	133	10.0	0.87	121	9.1	0.89	−12 (2)	−0.9 (0.2)	0.02 (0.00)	**9.6%**	**8.3%**	**2.6%**
GPR	137	10.3	0.86	119	8.9	0.89	−18 (6)	−1.4 (0.3)	0.03 (0.01)	**13.1%**	**13.8%**	**3.7%**
SVM	133	10.2	0.87	115	8.0	0.90	−18 (4)	−2.2 (0.3)	0.03 (0.01)	**13.5%**	**21.0%**	**3.9%**
RF	153	12.1	0.82	151	11.7	0.83	−2 (4)	−0.4 (0.2)	0.01 (0.01)	1.6%	2.7%	0.7%
GBRT	145	11.2	0.84	135	11.0	0.86	−10 (5)	−0.2 (0.3)	0.02 (0.01)	**7.2%**	1.8%	2.2%
NN	238	19.4	0.57	206	17.3	0.67	−32 (13)	−2.1 (1.2)	0.10 (0.04)	**13.4%**	**10.7%**	**17.7%**

Note: Results before wrapper feature selection is the benchmark for paired *t*-test. Positive PIPs indicate better performance after wrapper feature selection. A bold font indicates the performance difference is statistically significant at the 5% level.

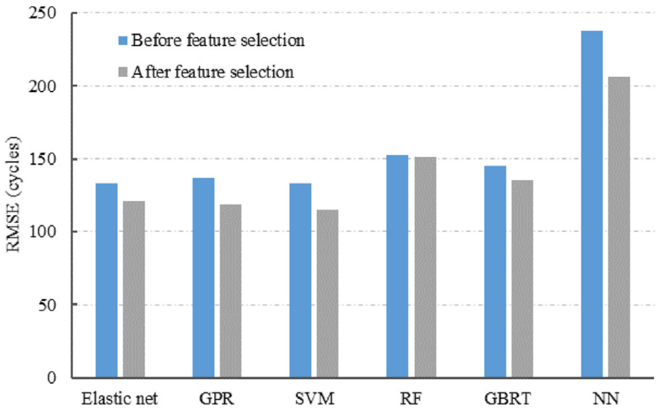

Fig. 6. RMSEs of six ML models before and after wrapper feature selection.

SVM is more sensitive to the feature size. In addition, for the elastic net, the RF, and the GBRT, which are essentially embedded feature selection methods, better results are achieved after feature selection, but their performance enhancements are not as large as those generated by the other three models.

4.3. *Results — further comparison*

In this section, the performance of our proposed methods is compared with the original MIT work [5], in which the MIT dataset was first provided. The MIT work was based on three testing batches, and the dataset was divided into three subsets: a training set (41), a primary testing set (43) and a secondary testing set (40). To the best of our knowledge, there are some problems with this kind of data split. First, the training set (41) is too small to properly train the ML models, and may easily lead to overfitting. Second, as most training samples are short-lived cell samples while the long-lived samples are mostly assigned to the testing set, the resulting imbalanced sample distribution will fail to capture the dynamics of long-lived cells in the testing set. To address these problems, the 20-times stratified random sampling method, as detailed in Section 3.3, is adopted to split the dataset, and evaluate the prediction performance. Specifically, in the MIT work, a 6-feature model was fed into the elastic net prediction model, and the RMSE on the secondary testing set was 173 cycles. By using our proposed new data split, the RMSE is substantially reduced, to 127 cycles, which means that our data split generates more accurate prediction results.

The performance of the 6-feature model (presented to be the best in the MIT work) and our 12-feature model (derived from the wrapper-based feature selection method) are compared via a paired *t*-test. Table 7 offers the comparison results on six ML models, while Fig. 7 visually presents the RMSE results. For all of the ML models except NN, prediction based on the 12-feature model is much better than that based on the 6-feature model. For example, the PIP of RMSE is 5.1% for the elastic net, 2.4% for the GPR, 1.3% for the SVM, 15.4% for the RF, and 9.7% for the GBRT. The performance of the RF improves the most, with positive PIPs on the three metrics, which are all statistically significant. Overall, the 12 features derived from the wrapper-based method are more predictive than the six features presented in the MIT work. In addition, as seen in Fig. 7, for both 6-feature and 12-feature models, the elastic net, the GPR, and the SVM produce better prediction results than the other three models.

Finally, the best prediction result of the proposed method (wrapper + SVM) and that of the MIT work are compared. By using the proposed method, the RMSE is substantially decreased, from 173 to 115 cycles, and the MAPE is reduced from 8.6% to 8.0%. Therefore, the performance improvements compared with the MIT's results are substantial.

Table 7
Comparison of six ML models using MIT's 6-feature model and the proposed 12-feature model.

ML models	MIT's 6-feature model			Proposed 12-feature model			Results of paired *t*-test			Performance improvement percentage (PIP) (%)		
							Mean difference (Standard error)					
	RMSE (cycles)	MAPE (%)	R^2	RMSE (cycles)	MAPE (%)	R^2	RMSE (cycles)	MAPE (%)	R^2	RMSE	MAPE	R^2
Elastic net	127	9.5	0.88	121	9.1	0.89	−6 (5)	−0.4 (0.2)	0.01 (0.01)	5.1%	3.8%	1.4%
GPR	122	9.3	0.88	119	8.9	0.89	−3 (5)	−0.4 (0.3)	0.01 (0.01)	2.4%	3.9%	0.6%
SVM	117	9.0	0.89	115	8.0	0.90	−2 (6)	−1.0 (0.4)	0.01 (0.02)	1.3%	**11.1%**	1.2%
RF	178	12.6	0.77	151	11.7	0.83	−27 (6)	−0.9 (0.2)	0.06 (0.01)	**15.4%**	**7.2%**	**8.4%**
GBRT	149	11.2	0.83	135	11.0	0.86	−15 (8)	−0.1 (0.4)	0.03 (0.02)	**9.7%**	1.5%	3.7%
NN	181	17.1	0.73	206	17.3	0.67	26 (15)	0.2 (1.3)	−0.07 (0.04)	**−14.1%**	−1.1%	**−9.2%**

Note: MIT's 6-feature model is the benchmark for paired *t*-test. Positive PIPs indicate that the proposed 12-feature model presents better performance than MIT's 6-feature model. A bold font indicates the performance difference is statistically significant at the 5% level.

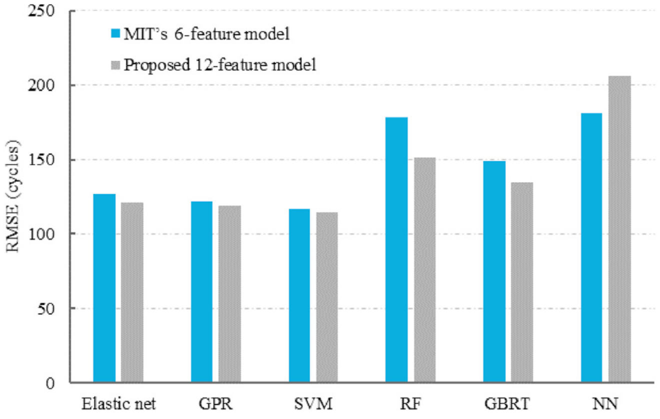

Fig. 7. RMSEs of six ML models using MIT's 6-feature model and the proposed 12-feature model.

5. Conclusion and future work

Accurate battery lifetime prediction in early cycles could effectively speed up the development and optimisation of lithium-ion batteries. In this paper, a comprehensive ML-based framework was proposed for the early prediction of battery lifetime, which consisted of three modules, namely feature extraction, feature selection, and ML-based prediction. The MIT dataset, which is the largest publicly available dataset, containing 124 battery degradation samples, was fully investigated. In the proposed framework, first, the battery ageing dynamics were investigated and discussed from various perspectives, such as the evolution of various informative parameters including the voltage, capacity, temperature, and IR during charging and discharging processes. Based on these ageing patterns, a set of 42 features were manually designed using the battery degradation data from only the first 100 cycles, at which point most cells had not shown any signs of capacity degradation yet. The 42 features were grouped into five categories, i.e., charge-related features, discharge-related features, capacity-related features, temperature-related features, and IR-related features. Second, to eliminate irrelevant and redundant features from the full set of 42 features, four typical feature selection methods (filter, wrapper, embedded, and PCA methods) were studied comparatively to select an optimal lower-dimensional feature subset. Third, the selected features were subsequently fed into six representative ML models (the elastic net, GPR, SVM, RF, GBRT, and NN) to comparatively investigate the performance of various ML models in the battery lifetime prediction application.

A series of computational experiments were conducted to demonstrate the effectiveness the proposed framework for early predicting the battery lifetime. To capture the pattern of most cell samples, a stratified random sampling method was presented to split the dataset, and was repeated 20 times to reduce the randomness of experimental results. Regarding feature selection methods, results showed that the wrapper-based feature selection method outperformed other methods, and significantly improved the predictive performance of subsequent ML prediction models. Regarding ML prediction models, results indicated that both before and after wrapper feature selection, the elastic net, the GPR, and the SVM provided better results than the RF, the GBRT, and the NN, because the former three models were relatively simple and therefore generally applicable to our small-sample-sized lifetime prediction problem. The NN had the worst performance for battery lifetime prediction, as it required a large number of samples to sufficiently train the network. By using a statistical technique, paired *t*-test, it was statistically concluded that the SVM model integrated with the wrapper-based feature selection was the most effective method for battery lifetime prediction, with an RMSE of 115 cycles, a MAPE of 8.0%, and an R^2 of 0.90. Finally, the performance of the proposed framework was compared with that of the MIT method. Results showed that the proposed 12-feature model, which was derived from the wrapper-based feature selection, was superior to the 6-feature model presented in the MIT work. By using the proposed framework, the RMSE was substantially decreased from 173 to 115 cycles, and the MAPE was reduced from 8.6% to 8.0%.

In this paper, the battery lifetime prediction framework is proposed based on the MIT dataset, in which the battery samples have a nominal capacity of 1.1 Ah and are cycled with varied-charging and constant-discharging policies. Although in real applications, types of lithium-ion batteries and the operation conditions could differ, for example, batteries with larger nominal capacity, or cycled under different charging and discharging profiles, this framework can be adapted into other applications. Since in the first stage of the proposed framework, the features are extracted by analysing the changing pattern of various informative parameters like the voltage, capacity, temperature, and IR during charge-discharge process, the analysis of these parameters can be generalized to different battery types and operation conditions, and features for battery lifetime prediction can be extracted in a similar way just by changing the profiles of these informative parameters. The discussion of various feature selection methods and ML prediction models could also be generalized to other battery applications. Due to the limitation of lab conditions and experiment time, the proposed framework is currently validated on the MIT dataset, in the future, more computational experiments will need to be conducted on other battery types under different operation conditions.

In the future, we will try to improve this work from three aspects. First, we will explore more potential efficient features for battery lifetime prediction. And instead of manual feature extraction used in this work, we will further explore deep learning based automatic feature extraction method. Second, we will explore that whether we can achieve an accurate lifetime prediction in even earlier cycles, e.g., the first 50 cycles. Third, we will explore the performance of the proposed framework in other battery applications, such as battery sorting/grading and pack design applications, which are battery lifetime classification problems.

Author statement

Zicheng Fei: Investigation, Data curation, Methodology, Software, Validation, Writing- Original draft, Writing- Reviewing and Editing; Fangfang Yang: Conceptualization, Data curation, Supervision, Writing- Reviewing and Editing; Kwok-Leung Tsui: Project administration, Supervision, Writing- Reviewing and Editing. Lishuai Li: Writing- Reviewing and Editing; Zijun Zhang: Writing-Reviewing and 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.

Acknowledgment

This research was supported by the General Research Fund (Project No. CityU 11204419).

References

[1] Schmuch R, Wagner R, Hörpel G, Placke T, Winter M. Performance and cost of materials for lithium-based rechargeable automotive batteries. Nature Energy 2018;3(4):267–78.

[2] You G-w, Park S, Oh D. Real-time state-of-health estimation for electric vehicle batteries: a data-driven approach. Appl Energy 2016;176:92–103.

[3] Wang S, Guo D, Han X, Lu L, Sun K, Li W, Sauer DU, Ouyang M. Impact of battery degradation models on energy management of a grid-connected DC microgrid. Energy 2020;207:118228.

[4] Homan B, ten Kortenaar MV, Hurink JL, Smit GJM. A realistic model for battery state of charge prediction in energy management simulation tools. Energy 2019;171:205–17.

[5] Severson KA, Attia PM, Jin N, Perkins N, Jiang B, Yang Z, Chen MH, Aykol M, Herring PK, Fraggedakis D. Data-driven prediction of battery cycle life before capacity degradation. Nature Energy 2019;4(5):383–91.

[6] Xu P, Li J, Sun C, Yang G, Sun F. Adaptive state-of-charge estimation for lithium-ion batteries by considering capacity degradation. Electronics 2021;10(2):122.

[7] Yang F, Song X, Dong G, Tsui K-L. A coulombic efficiency-based model for prognostics and health estimation of lithium-ion batteries. Energy 2019;171: 1173–82.

[8] Burgess WL. Valve Regulated Lead Acid battery float service life estimation using a Kalman filter. J Power Sources 2009;191(1):16–21.

[9] Micea MV, Ungurean L, Carstoiu GN, Groza V. Online state-of-health assessment for battery management systems. IEEE Transactions on Instrumentation and Measurement 2011;60(6):1997–2006.

[10] He W, Williard N, Osterman M, Pecht M. Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method. J Power Sources 2011;196(23):10314–21.

[11] Yang F, Wang D, Xing Y, Tsui K-L. Prognostics of Li (NiMnCo) O 2-based lithium-ion batteries using a novel battery degradation model. Microelectron Reliab 2017;70:70–8.

[12] Hu C, Ye H, Jain G, Schmidt C. Remaining useful life assessment of lithium-ion batteries in implantable medical devices. J Power Sources 2018;375:118–30.

[13] Yang F, Wang D, Zhao Y, Tsui K-L, Bae SJ. A study of the relationship between coulombic efficiency and capacity degradation of commercial lithium-ion batteries. Energy 2018;145:486–95.

[14] Liu S, Winter M, Lewerenz M, Becker J, Sauer DU, Ma Z, Jiang J. Analysis of cyclic aging performance of commercial Li4Ti5O12-based batteries at room temperature. Energy 2019;173:1041–53.

[15] Tian J, Xu R, Wang Y, Chen Z. Capacity attenuation mechanism modeling and health assessment of lithium-ion batteries. Energy 2021;221:119682.

[16] Broussely M, Herreyre S, Biensan P, Kasztejna P, Nechev K, Staniewicz RJ. Aging mechanism in Li ion cells and calendar life predictions. J Power Sources 2001;97–98:13–21.

[17] 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.

[18] Xu F, Yang F, Fei Z, Huang Z, Tsui K-L. Life prediction of lithium-ion batteries based on stacked denoising autoencoders. Reliab Eng Syst Saf 2021;208: 107396.

[19] Yang F, Wang D, Xu F, Huang Z, Tsui K-L. Lifespan prediction of lithium-ion batteries based on various extracted features and gradient boosting regression tree model. J Power Sources 2020;476.

[20] Li X, Yuan C, Li X, Wang Z. State of health estimation for Li-Ion battery using incremental capacity analysis and Gaussian process regression. Energy 2020;190:116467.

[21] Shu X, Li G, Shen J, Lei Z, Chen Z, Liu Y. A uniform estimation framework for state of health of lithium-ion batteries considering feature extraction and parameters optimization. Energy 2020;204:117957.

[22] 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. J Power Sources 2018;384:387–95.

[23] Li S, Ju C, Li J, Fang R, Tao Z, Li B, Zhang T. State-of-Charge estimation of lithium-ion batteries in the battery degradation process based on recurrent neural network. Energies 2021;14(2):306.

[24] Wu J, Zhang C, Chen Z. An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks. Appl Energy 2016;173:134–40.

[25] Ma G, Zhang Y, Cheng C, Zhou B, Hu P, Yuan Y. Remaining useful life prediction of lithium-ion batteries based on false nearest neighbors and a hybrid neural network. Appl Energy 2019;253:113626.

[26] Zhang Y, Xiong R, He H, Pecht MG. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. IEEE Trans Veh Technol 2018;67(7):5695–705.

[27] Li Y, Liu K, Foley AM, Zülke A, Berecibar M, Nanini-Maury E, Van Mierlo J, Hoster HE. Data-driven health estimation and lifetime prediction of lithium-ion batteries: a review. Renew Sustain Energy Rev 2019;113:109254.

[28] Ren L, Zhao L, Hong S, Zhao S, Wang H, Zhang L. Remaining useful life prediction for lithium-ion battery: a deep learning approach. IEEE Access 2018;6: 50587–98.

[29] Hu C, Jain G, Zhang P, Schmidt C, Gomadam P, Gorka T. Data-driven method based on particle swarm optimization and k-nearest neighbor regression for estimating capacity of lithium-ion battery. Appl Energy 2014;129:49–55.

[31] Zhang C, Jiang J, Gao Y, Zhang W, Liu Q, Hu X. Charging optimization in lithium-ion batteries based on temperature rise and charge time. Appl Energy 2017;194:569–77.

[32] Birkl CR, Roberts MR, McTurk E, Bruce PG, Howey DA. Degradation diagnostics for lithium ion cells. J Power Sources 2017;341:373–86.

[33] Dubarry M, Liaw BY, Chen M-S, Chyan S-S, Han K-C, Sie W-T, Wu S-H. Identifying battery aging mechanisms in large format Li ion cells. J Power Sources 2011;196(7):3420–5.

[34] 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. J Power Sources 2018;373:40–53.

[35] El Mejdoubi A, Oukaour A, Chaoui H, Gualous H, Sabor J, Slamani Y. State-of-charge and state-of-health lithium-ion batteries' diagnosis according to surface temperature variation. IEEE Trans Ind Electron 2015;63(4):2391–402.

[36] Li J, Cheng K, Wang S, Morstatter F, Trevino RP, Tang J, Liu H. Feature selection: a data perspective. ACM Comput Surv 2017;50(6). Article 94.

[37] Hossain Lipu M, Hannan M, Hussain A, Saad MHM. Optimal BP neural network algorithm for state of charge estimation of lithium-ion battery using PSO with PCA feature selection. J Renew Sustain Energy 2017;9(6):064102.

[38] Lipu MH, Hannan M, Hussain A. Feature selection and optimal neural network algorithm for the state of charge estimation of lithium-ion battery for electric vehicle application. Int J Renew Energy Resour 2017;7(4):1700–8.

[39] Sheng H, Xiao J, Wang P. Lithium iron phosphate battery electric vehicle state-of-charge estimation based on evolutionary Gaussian mixture regression. IEEE Trans Ind Electron 2016;64(1):544–51.

[40] Chandrashekar G, Sahin F. A survey on feature selection methods. Comput Electr Eng 2014;40(1):16–28.

[41] Xue B, Zhang M, Browne WN, Yao X. A survey on evolutionary computation approaches to feature selection. IEEE Trans Evol Comput 2016;20(4):606–26.

[42] Richardson RR, Osborne MA, Howey DA. Gaussian process regression for forecasting battery state of health. J Power Sources 2017;357:209–19.

[43] Patil MA, Tagade P, Hariharan KS, Kolake SM, Song T, Yeo T, Doo S. A novel multistage Support Vector Machine based approach for Li ion battery remaining useful life estimation. Appl Energy 2015;159:285–97.

[44] Yang D, Wang Y, Pan R, Chen R, Chen Z. State-of-health estimation for the lithium-ion battery based on support vector regression. Appl Energy 2018;227:273–83.

[45] Wei J, Dong G, Chen Z. Remaining useful life prediction and state of health diagnosis for lithium-ion batteries using particle filter and support vector regression. IEEE Trans Ind Electron 2018;65(7):5634–43.

[46] Li Y, Zou C, Berecibar M, Nanini-Maury E, Chan JC-W, van den Bossche P, Van Mierlo J, Omar N. Random forest regression for online capacity estimation of lithium-ion batteries. Appl Energy 2018;232:197–210.

[47] Zheng Z, Peng J, Deng K, Gao K, Li H, Chen B, Yang Y, Huang Z. A Novel method for lithium-ion battery remaining useful life prediction using time window and gradient boosting decision trees, 2019 10th International Conference on power Electronics and ECCE Asia (ICPE 2019-ECCE Asia). IEEE; 2019. p. 3297–302.

[48] Klein A, Falkner S, Bartels S, Hennig P, Hutter F. Fast Bayesian optimization of machine learning hyperparameters on large datasets. In: Aarti S, Jerry Z, editors. Proceedings of the 20th international conference on artificial intelligence and statistics, PMLR, proceedings of machine learning research; 2017. p. 528–36.

[49] Kim TK. T test as a parametric statistic. Korean journal of anesthesiology 2015;68(6):540.