



State of health estimation of lithium-ion battery with automatic feature extraction and self-attention learning mechanism



Yiyue Jiang^{a,b}, Yuan Chen^c, Fangfang Yang^{a,b}, Weiwen Peng^{a,b,*}

^a School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen, 518107, China

^b Guangdong Provincial Key Laboratory of Fire Science and Intelligent Emergency Technology, Guangzhou, 510006, China

^c Science and Technology on Reliability Physics and Application Technology of Electronic Component Laboratory, Guangzhou, 510610, China

HIGHLIGHTS

- An automatic health feature extraction method for LIBs without prior knowledge is proposed.
- Convolutional autoencoder model is used to extract features automatically.
- Self-attention mechanism is incorporated to obtain accurate SOH estimation results.
- The performance is compared with the manual feature-based methods and other data-driven methods.

ARTICLE INFO

Keywords:

Lithium-ion battery
State of health estimation
Automatic feature extraction
Convolutional autoencoder
Self-attention mechanism

ABSTRACT

Accurate state of health (SOH) estimation is significantly important to ensure the safe and reliable operation of lithium-ion battery. Most existing data-driven estimation methods are based on feature engineering and rely heavily on expert experience and manual operation. However, manually extracting qualified health features requires rich prior knowledge, and these highly-designed features for one specific application may not generalize well to other situations. In this work, an automatic feature extraction method combining convolutional autoencoder and self-attention mechanism is proposed for battery SOH estimation. With preprocessed data fed into the convolutional autoencoder, efficient features characterizing battery health are automatically extracted without human intervention. A self-mechanism module is then further employed to map these high-dimensional abstract health features into battery SOH. Finally, experimental study of battery aging is implemented to demonstrate the proposed method, and comparisons of the proposed method with existing data-driven approaches and the manual feature-based methods have also been presented. With the help of the convolutional autoencoder and self-attention module, the proposed method replaces the conventional manual feature engineering with automatic feature extraction, and reaches 0.0048 average test root-mean-squared error (RMSE) and 0.46% mean-absolute-percentage error (MAPE) on our dataset and 3.69% on the NASA public dataset.

1. Introduction

Lithium-ion batteries (LIBs) are well known energy storage equipment with many virtues such as high energy-density and high power-density [1,2]. They play an important role in many applications such as transportation, aerospace industry and portable electronics [3,4]. Energy storage capacity degrades and the power delivery capability deteriorates by battery calendar time and charge/discharge cycles [5]. When battery health degrades to a certain extent, accidents such as battery leakage, insulation damage and partial short circuit problems

may give rise to several safety hazards [6,7]. Therefore, efficient lithium-ion battery state of health (SOH) estimation is necessary to remind users the time to replace and maintenance [8,46].

There are many challenges for SOH estimation, for example, battery capacity can only be indirectly inferred from the measurable variables such as voltage and current. Besides, the degradation of LIB capacity is a complicated and nonlinear process, varying with different battery type and working conditions [9–11]. Many methods have been proposed for lithium-ion battery SOH estimation, which can be broadly categorized into model-based methods and data-driven methods [12,13].

* Corresponding author. School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen, 518107, China.
E-mail address: pengww3@mail.sysu.edu.cn (W. Peng).

Model-based methods are methods aiming at estimating battery SOH by establishing physical and chemical degradation models and measuring related parameters. And these methods can be further categorized into electrical equivalent circuit model [14,15], electrochemical model [16, 17] and empirical model [18–20]. Model-based methods are interpretable, easy to understand, and can estimate battery degradation from the root causes of aging. However, there are still several unsolved problems among these methods. For instance, the establishment of the models relies on abounding prior knowledge about LIBs. Moreover, establishing models puts forward a variety of assumptions, which easily causes deviation from the real-world applications. In addition, inaccurate parameter estimation further hinders the performance.

In contrast, data-driven methods can provide accurate degradation approximation with minimal demands of lithium-ion battery physical and chemical mechanism knowledge. They estimate SOH by directly utilizing historical cycle data that can be easily collected through sensors [21]. There are generally four steps in a data-driven battery SOH estimation method: data collection, feature engineering, model training and SOH estimation. Zhu et al. [22] extracted six statistical features from voltage curve of relaxation process after fully charging. Then they used XGBoost, ElasticNet and support vector regression models to estimate battery capacity on three datasets containing 130 *LiNiCoAl* and *LiNiCoMn* commercial batteries. Their best estimation result achieved 0.011 RMSE. Wang et al. [23] developed a modified Gaussian process regression model for battery SOH estimation. Three features were extracted from charging voltage, discharging voltage and discharging temperature. Their results reached 0.0241 RMSE on the NASA public dataset. Data-driven methods are easy for implementation and can perform efficiently on SOH estimation tasks. More importantly, they can achieve equivalent or even more precise results compared with model-based methods.

However, there is still room for improvement among data-driven methods, not least in the case of the need for prior knowledge when extracting efficient health features that can best represent battery health state. Roman et al. [24] developed a machine learning (ML) capacity estimation pipeline which extracted 30 features from constant current-constant voltage (CC-CV) charging stage. Most relevant features were further selected through a modified cross validation method, and these features were finally fed into four ML algorithms to estimate battery capacity. Their approach was validated on multiple public datasets with varied charging protocols, and the best result of 0.0045 RMSE were achieved. Huang et al. [25] proposed an SOH evaluation method based on the experimental data collected from NCA battery cell and LFP battery module using the framework of probability density function which can be applied under low sampling frequency. They introduced two new health indicators of regional voltage and regional frequency. The SOH has a strong linear relationship with the extracted regional frequency feature, the fitting R^2 can reach more than 0.99. Ma et al. [26] extracted three features from differential thermal voltammetry curve, and recursive neural network models were constructed to predict remaining useful life and estimate SOH. The approach was validated on NASA database and SOH estimation RMSE maintained less than 1%. Ma et al. [27] extracted six features from charging-discharging process for NASA dataset and MIT dataset, some of them were directly measured while others needed calculation. Then they used transfer learning and deep belief network-long short-term memory hybrid network to perform SOH estimation. Their best results were obtained on the MIT dataset with 0.99% MAPE. It is not doubtful that these existing data-driven methods are feasible in SOH estimation problems, but their drawbacks are also very obvious, that is, some of these manual feature engineering processes require prior knowledge of battery aging mechanisms and extensive mathematical calculations. Besides, it is still necessary to find suitable data-driven models to process the health features extracted from battery cycle data. Therefore, how to use much less prior knowledge and work load to easily while effectively extract health features which can be handled by an appropriate SOH estimation

model remains an open challenge.

One applicable solution is to utilize the strong learning and nonlinear modeling ability of ML algorithms to automatically extract features from raw battery data. Not too many works focused on this aspect, and existing methods often integrate feature extraction and SOH estimation module into a supervised model which is fed with raw battery data as input and trained with SOH values as labels. For instance, Gong et al. [28] utilized an encoder-decoder model to estimate battery SOH. Convolutional neural network and ultra-lightweight subspace attention mechanism was combined in the encoder for feature extraction, and there was no further manual feature extraction. Only raw battery current, voltage and temperature data were needed to estimate battery SOH. However, when extracting features using supervised learning algorithm, the model will learn to map the input raw data into the SOH labels. By the intervention of SOH labels, these features do not naturally represent the original interior proper of raw cycle data, but have a specific tendency which is artificially added to express SOH labels. An unsupervised learning algorithm can be a good solution to improve this problem with building an automatic feature extractor. No label is needed during unsupervised learning training. Therefore, features which can independently represent the raw cycle data can be obtained through this algorithm.

Autoencoder (AE), one of the representative unsupervised learning algorithms, is a good choice for feature extraction. It has a merit of extracting lower-dimensional features by reconstructing input on the output layer. AEs have been already used as feature extractors in some other fields. For example, Liu et al. [29] used stacked autoencoder to extract features from frequency domain signals to diagnose gearbox faults. In the field of short-term water demand forecast for urban water supply, Chen et al. [30] proposed a framework using one dimensional convolution to extract features automatically based on historical data and improve the forecasting results. And when addressing the problem of accurate hydraulic piston pump fault diagnosis, Tang et al. [31] introduced normalized convolutional neural network framework for feature extraction. The motivation of these automatic feature extraction works can be summarized as to solve the research gap that manual features may not mine the data completely, and enhance the model adaptability towards raw data to improve model behavior.

However, in battery health estimation works, AEs were hardly used as feature extractors. Xu et al. [32] utilized stacked denoising autoencoder to predict battery life cycles. They manually engineered features from temperature, internal resistance and discharge capacity-voltage curves, SDAE model was used to predict battery life. Wei et al. [33] manually extracted six features from battery voltage, current and temperature data. SAE was utilized to address the problem of redundancy and deficiency of the features, then Gaussian mixture regression is used to predict battery life. AEs in these works were used on other purposes, their advantages on automatically extracting features haven't been developed to the best extent.

There are still some further problems when introducing AE to extract features automatically, for example, the features an AE extracts are abstract and relatively high-dimensional representations, bringing some trouble for the subsequent processing. Therefore, how to use these abstract health features to estimate battery SOH becomes a new challenge. Some easily-structured models may not have the ability to unearth the relationship between these abstract health features and battery SOH. Attention mechanism models have a strong ability to find out which part of the input is important to the task, they can augment effective information and suppress redundant one. Widespread applications in other fields such as natural language processing [34,35] and computer vision [36,37] also demonstrate the powerful ability of attention mechanisms. Self-attention (SA) suits for SOH estimation task among various types of attention mechanism in this paper. It calculates response at a position as a weighted sum of the features at all positions with only a small computational cost [36]. For instance, Vaswani et al. [38] have demonstrated that machine translation models could achieve excellent

results by solely using a self-attention model, proving the effectiveness of this technique. SA mechanism is suited to further learn the internal rules of the abstract health features extracted by AE by exploring the interaction among the elements and estimate battery SOH in our work.

As existing data-driven approaches are mostly implemented under a framework with manual feature extraction step, which is time-consuming, labor-intensive and low-generalized, in this paper, we propose an autoencoder based automatic feature extraction approach to replace the conventional manual feature engineering in lithium-ion battery state of health estimation tasks. Self-attention mechanism, a powerful ML model, is further used to process the abstract health features extracted by the autoencoder to estimate battery state of health. The main contributions of this work are summarized as follows:

- 1) A novel data-driven SOH estimation approach of LIBs is proposed. This approach can extract features automatically from raw battery cycle data, thus saving time and labor from the troublesome process

of manual feature engineering. The experimental results on our dataset show that the proposed method achieved average 0.0043 RMSE and 0.38% MAPE, demonstrating a good performance of our proposed model on LIB SOH estimation task.

- 2) Convolutional autoencoder (CAE) is introduced as the feature extractor. As an unsupervised learning algorithm, the CAE can automatically extract features directly from the raw cycle data without the intervention of labels. Our CAE model provides a convenient and efficient way to extract health features from raw battery data.
- 3) With the employment of SA module, features can be properly processed and precise SOH estimation results can be obtained. Features extracted by CAE can be abstract and relatively high-dimensional, the SA module can guarantee the learning ability of estimation model and provide accurate estimation results.

The rest of this paper is organized as follows: Section 2 will propose

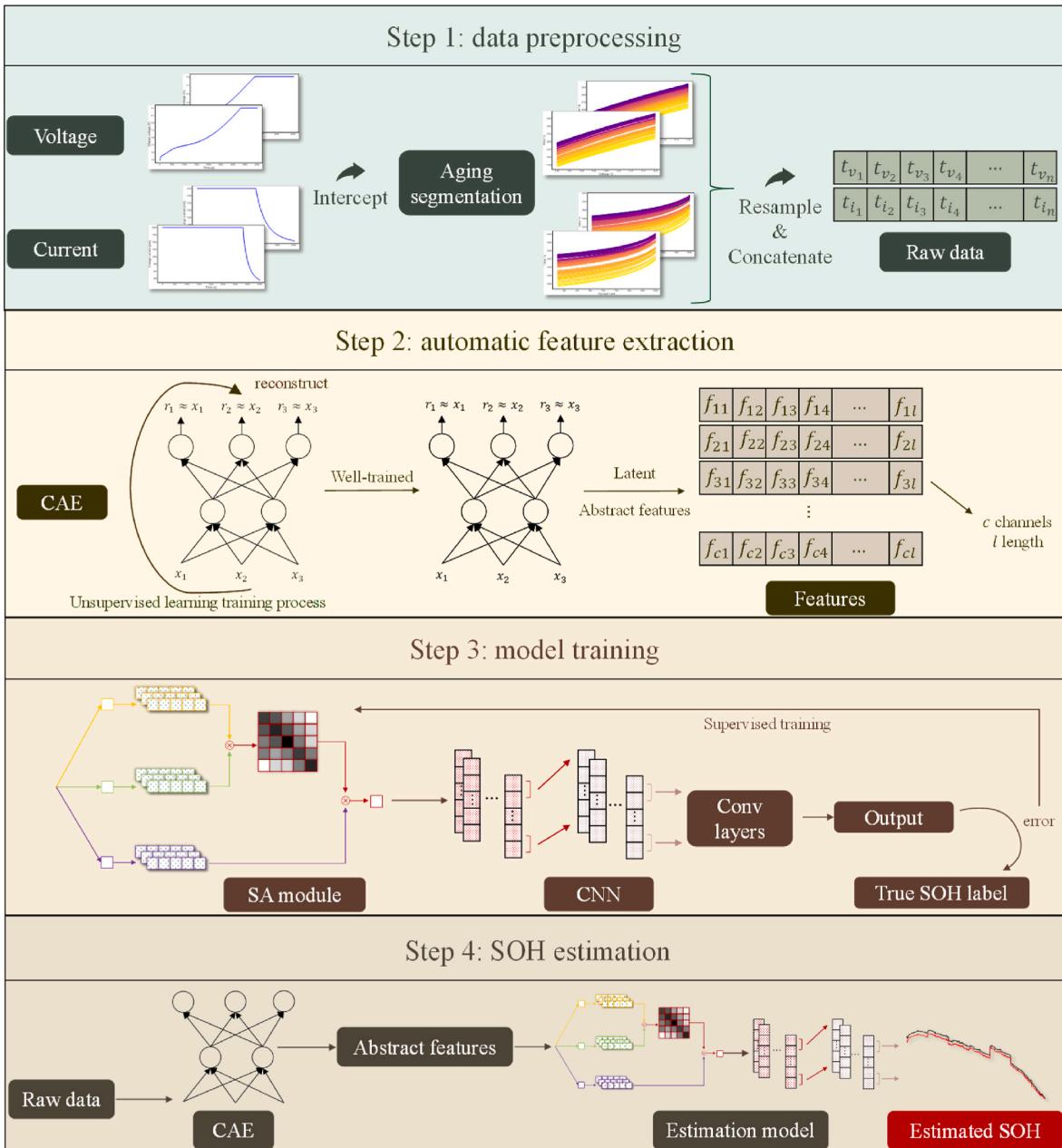


Fig. 1. The framework of our proposed method.

our overall framework and workflow and introduce the preliminaries of our methodology. Section 3 will discuss the experimental details, including the dataset introduction, implementation details, evaluation metrics, and experimental results. Finally, we give our conclusions in Section 5.

2. Methodology

In this section, we will introduce the overall framework and some basic knowledge of the proposed CAE-SA SOH estimation method.

2.1. Overview

The whole framework of the proposed method is shown in Fig. 1. In step 1, we first preprocess the voltage and current data. Then in step 2, we feed the preprocessed raw data into CAE and train it through unsupervised learning training, and we can get the abstract health features in the hidden layer of the CAE. In step 3, we feed the extracted features into the estimation model, and train it through supervised learning model using the true SOH values as labels to conduct the training process. Finally, in step 4, after the whole model is trained, we can perform SOH estimation on our CAE-SA model. The detailed description of the four steps in the proposed method's framework are as follows:

Step 1: data preprocessing. We utilize raw voltage data from CC charging stage and current data from CV charging stage, and provide a “cutting-laying down” preprocessing method. The preprocessed data are 2-channel sequences which describe the change of charging time on the charging period with the same voltage and current.

Step 2: Automatic feature extraction. CAE is utilized in our method to automatically extract features from the preprocessed data. The model is trained to learn to reconstruct the input data on the output layer through unsupervised learning algorithm, so other values such as SOH are not required during training. The extracted features can be obtained on the hidden layer. Therefore, the feature extraction process needs only to feed the preprocessed raw data into the CAE model and battery health features can be extracted automatically. Note that once CAE training is done, we construct our model by removing the decoder, and freezing the parameters of the encoder, then attaching the estimation model after the encoder, that in the follow-up training the parameters in the encoder will not be upgraded.

Step 3: model training. The follow-up ML model needs to use the features extracted by CAE, and we employ SA module in our model. SA module can learn the internal relationship among data elements, and can handle the abstract health features. Besides, a CNN model is combined with the SA module to be the SOH estimation model. This model is trained to learn to map the abstract health features into the SOH values through supervised learning algorithm, thus, SOH values are required as labels during training.

Step 4: SOH estimation. We can get the well-trained battery SOH estimation CAE-SA model after the above steps. Once the raw battery data are acquired and preprocessed, we input them into the CAE to extract features. Then these features will be fed into the SA estimation model, which will output the estimated SOH.

2.2. Data preprocessing

A “cutting-laying down” data preprocessing method is used for data preprocessing. Specifically, “cutting” means the intercepting whose details have been discussed before, and “laying down” means the data alignment which needs an exchange of the independent and dependent variable on the charging curve. For the CCCV charging data under different SOH of LIBs, the cut-off voltage and cut-off current are the same in different cycles while the charging time varies. The purpose of this preprocessing method is to intercept the data (“cut”) and align them

(“lay-down”). A sketch of this process is shown in Fig. 3. The voltage and current data have a clear trend that follows the variation of the cycles.

Time-varying charging current and voltage data can be easily measured by sensors, and the patterns of which in different cycles can reflect different health state. However, it is unrealistic to utilize the entire charging data as model input due to the constantly changing voltage or current range and corresponding charging time. Therefore, to facilitate subsequent processing as well as reduce data volume, we attempt to intercept the voltage data of CC stage and current data of CV stage into a specific data segment with voltage or current range that is included in each different cycle and can best represent the battery SOH. The intercepted voltage or current data segment can be represented by the range of voltage: $[V_l, V_h]$ at CC stage, and current: $[I_l, I_h]$ at CV stage, where V_l and I_l mean the lower bounds of voltage or current segment, V_h and I_h mean the higher bounds. A sketch of the intercepting process of CCCV charging data from one cycle is shown in Fig. 2, this intercepting step is called “cutting”. After “cutting” step, in each cycle we will obtain one segment of voltage data from CC charging stage and another of current data from CV charging stage with the same voltage or current range. Note that the method can adopt voltage and current data segments from any ranges as long as with the ranges exist in all different cycles. While the determination of the both bounds of voltage range $[V_l, V_h]$ and current range $[I_l, I_h]$ is conducted by experiments in our case, which will be introduced in details in section 3.4.1.

The intercepted voltage and current data are formatted as Fig. 3(a). Apparently, for different cycles their corresponding charging time is different, and this difference has a correlation with the cycle number. Thus, to reflect this difference, variable relationship should be reversed, the varying time should be dependent variable while the voltage or current data with the fixed range $[V_l, V_h]$ and $[I_l, I_h]$ in different cycles should be independent variable. This exchange step of independent and dependent variables is called “laying down”. Fig. 3 shows this step performed on the data from different cycle.

Please note that the time value we utilize is the relative charging time in a single cycle. For instance, when the charging starts in each cycle, the time will be re-marked as 0 to start timing, as shown in Fig. 2; and when CC charging stage ends and CV charging starts, the timing continues without resetting to 0. We have to mention that, in each cycle, the battery is assumed fully discharged before charging. If it isn't fully discharged, the remaining capacity may cause uncertain charging voltage and current. This may affect the data preprocessing and consequently affect the performance. Possible solution to this problem is to change a way of selecting the voltage and current segment.

As shown in Fig. 3(b), to get a set of neatly processed curves, cubic spline interpolation, a spline curve fitting method is utilized to align the intercepted data. It aims to establish several piecewise cubic polynomial relationships, which are called the interpolation function. This method fits a smooth cubic spline curve by calculating the fitting function on each interpolation subinterval, and finally getting a piecewise fitting function on the whole interval. After alignment, data from different cycle will become tidy as shown in Fig. 3(b).

After the “cutting-laying down” preprocess, the voltage and current data are concatenated to be 2-channel sequences, in a shape of

$$d = (\text{cycle}, \text{channel}, \text{samples}), \quad (1)$$

in which the first dimension stands for different cycles, the second dimension stands for the channel for voltage and current data, and the third dimension stands for the sampling points in one cycle while fitting the spline curves. The preprocessed data will be fed into the CAE for feature extraction.

2.3. CAE-SA for lithium-ion battery SOH estimation

An autoencoder is a neural network structure that is trained to reconstruct the input to output. The network structure of an autoencoder

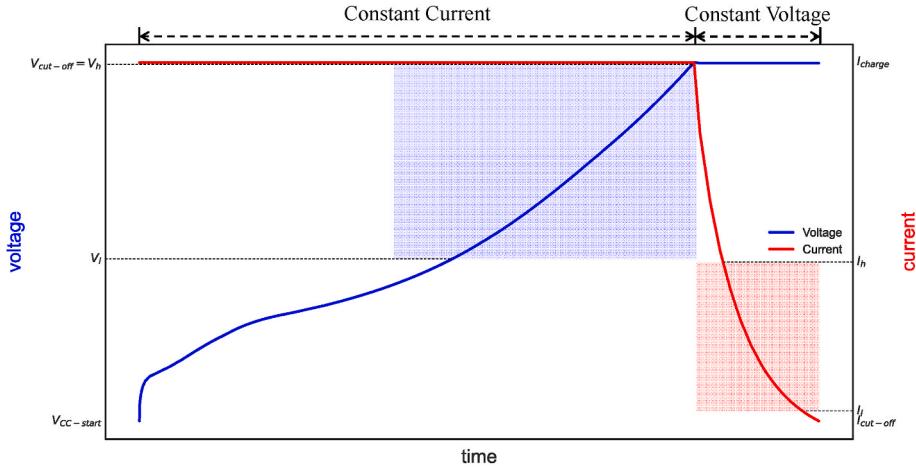


Fig. 2. The “cutting” on the charging voltage and current data in the constant current-constant voltage charging (CCCV) protocol. V_h , V_l , I_h and I_l represent the cutting points of voltage and current, respectively.

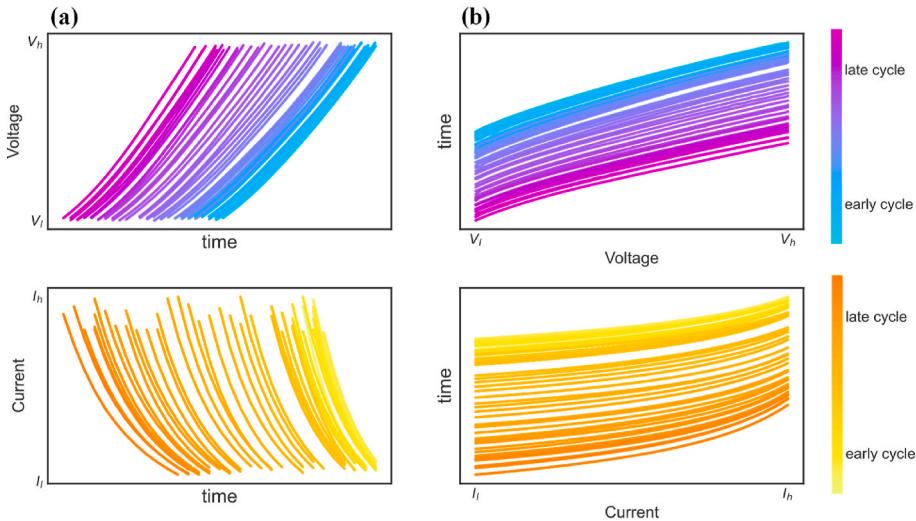


Fig. 3. A sketch of “lay down” in different cycles. (a) the data segments intercepted directly from the raw voltage and current charging data. (b) the data segments after “lay down”.

can be divided into two parts, the encoder and decoder:

$$\begin{cases} h = f(d) \\ \hat{d} = g(h) \end{cases} \quad (2)$$

where d is the preprocessed data with the shape of Eq. (1) input, and $f(*)$ means the encoder function, h is the latent features the encoder extracts; similarly, $g(*)$ means the decoder function and \hat{d} is the output, which is the reconstruction of input d . *Training an autoencoder doesn't require additional labels, which means that we train the whole autoencoder under the target of approximating the model output \hat{d} as close as possible to the target d .* The latent feature h can be regarded as the low-dimensional feature vectors representing the input data. The structure of decoder is always symmetric to that of encoder in a vanilla autoencoder.

A CAE is an autoencoder built with CNN. CNN, composed of multiple convolutional layers, is a kind of special neural network invented to process data that has a specific grid-like topology [39], the inputs and outputs of which are usually multidimensional tensors. To build up a CAE with CNN, we need to replace several fully connected layers in a vanilla AE with convolutional layers. Fully connected layers in a vanilla AE can't properly deal with multi-channel data such as our 2-channel data, whose expression is given in Eq. (1), it can only learn the

non-linear patterns on the last dimension of the input data. But convolutional layers are able to train multiple kernels on each channel of a multi-channel data, and learn the non-linear relationship among different channels. Finally, the convolutional layers will map the input data flexibly into another multi-channel data with different number of channels. These convolutional layers make the network component and structure more complex and endow the AE with stronger ability to extract latent feature representations from our input data not only on the last dimension representing the data points, but also different channels representing the voltage or current data.

Then, as for the SA module, we follow the attention architecture proposed by Zhang et al. [36] and Vaswani et al. [38], adopts the “query-key-value” (Q-K-V) mode. First, the inputs will be mapped to three sets: queries, keys and values, queries and keys share the same dimension d_k , while values have dimension of d_v , and they are packed together into matrices Q , K , and V . We compute the dot products of Q and K , divide it by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values, and then we multiply the weight matrix along with the V matrix to get the output [38]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

We use 3 different 1D-convolutional layers to map the input, which is extracted features by autoencoder, into three representations: query, key and value. The kernel size of these 1D-convolutional layers is 1 for each. The attention score map contains the self-attention weights of each element to another in the input data, in other words, this attention score map will reflex the importance of each element to every other element, and enhance the learning ability of the estimation model to help it perform the final SOH estimation task.

The data input into our SOH estimation machine learning model are 2-channels: ① time VS. voltage in CC charging stage, and ② time VS. current in CV charging stage, both 1 dimensional. We use 1D CNN model for sequential data processing. CAE, the autoencoder consisted of CNN, is used for extracting features. Then the CAE model is trained to reconstruct the input data according to Eq. (2), and finally, we can obtain the features at the output of the hidden layer h . After the automatic feature extraction by our CAE model, we feed the features into SA module to get the weight of each element and obtain the output, which is called the self-attention features. Finally, we input the self-attention features into a CNN to get the lithium-ion battery SOH estimation value. The whole model is trained through back propagation, a classic algorithm to train neural network models for nonlinearity relationship fitting.

3. Experimental

To verify the effectiveness and performance of our proposed method, extensive experiments are conducted to comprehensively show qualitative results and corresponding analyses are provided on our proposed model.

3.1. Dataset description

3.1.1. Our dataset

The dataset used in this work contains the aging data of 6 commercial ICR18650 cylindrical cells, and they are labeled as Cell 1 to 6, with parameters listed in Table 1. This dataset is used for most of the detailed step descriptions.

The cycle aging test was carried out based on six new LIBs at 25 °C. In each cycle, the charge contains a CC-CV charge process: a CC charge at 1 C-rate (2 A) until the voltage reaches 4.2 V, followed by a CV charge until detection of current below 0.04 A. The discharge is then performed in CC mode at 3 C-rate (6 A) until the voltage reaches a cut-off voltage of 2.75 V. A two-stage discharge approach is performed every 50 cycles to measure the battery remaining capacity. The first discharge is performed at 1 C-rate (2 A) until the voltage reaches 2.75 V, then after 10 min pause time the second discharge is performed at 0.5 C-rate (1 A) until the voltage reaches 2.75 V again. The discharge and charge capacity of LIBs can be obtained directly through the battery test system after two-stage discharge and CC-CV charge.

We use the cell data before the end-of-life (EOL) which is defined as the 80% of the nominal capacity, and the capacities before EOL of all cells are illustrated in Fig. 4. The SOH of cells in this paper is defined as

Table 1
Parameters of the 18,650 LIBs.

Parameter	Content
Anode material	Graphite
Cathode material	LiNi _{0.5} Co _{0.2} Mn _{0.3} O ₂
Maximum size	Diameter: 18.6 mm Height: 65.3 mm
Weight	About 43 g
Nominal capacity	2000 mAh
Nominal voltage	3.6 V
Discharge cut-off voltage	2.75 V
Charging upper limit voltage	4.20 ± 0.05 V
Charging cut-off current	0.02 C-rate

[40]:

$$SOH = \frac{C_{\text{pres}}}{C_{\text{nom}}}, \quad (4)$$

where C_{pres} represent present capacity while C_{nom} represents nominal capacity.

To utilize the full information the dataset contains, we make six datasets using these six batteries, where each battery was used as test set and the remaining five were used as a training set, and all our experiments were done on these six datasets.

3.1.2. NASA dataset

In our work, we further utilized another dataset, which is the famous NASA public dataset. In the NASA dataset, 4 Li-ion batteries which are numbered as 5, 6, 7 and 18 were run through charging and discharging cycles. Charging was carried out in a CC mode at 1.5A until the battery voltage reached 4.2V and then continued in a CV mode until the charge current dropped to 20 mA. Discharge was carried out at a CC level of 2A until the battery voltage fell to 2.7V, 2.5V, 2.2V and 2.5V for batteries 5 6 7 and 18 respectively. The experiments were stopped when the batteries reached a 30% fade in rated capacity from 2Ah to 1.4Ah. This dataset will be used for the description for the comparisons with manual features in experimental sections 4.3.

3.2. Implementation details

3.2.1. Automatic feature extraction

We use CAE to extract the abstract health features of the raw battery data, and the encoder structure of our CAE model is shown in Table 2 and the decoder structure is symmetric to the encoder. We use a fully convolutional architecture for our CAE, which means there are only convolutional layers in it. The encoder consists of five convolutional layers and one linear layer. The convolutional layers have kernel shape (5, 5, 3, 5, 5), output channels (8, 32, 64, 64, 20), stride (1, 1, 3, 1, 5), and the linear layer output a 20-channel feature vector with length of 5. Similarly, decoder has one linear layer and five 1D transpose convolutional layers. Despite the simplicity of our CAE, it provides an efficient way of extracting features from the raw battery data without expert experience.

3.2.2. SOH estimation

We feed the automatically extracted features from the CAE model into the SA module, and the network outputs an SOH estimation value. The specific architecture of the network is shown in Table 3.

The layers producing query, key and value are 1D convolutional layers with the channel numbers of (4, 4, 20), and their kernel sizes are all set to 1 to maintain the data length. After that, we obtain the self-attention features. Then these self-attention features are feed into the output network to produce a single value, finally, this value passes through an output function to produce the battery SOH estimation value. To obtain the value in the appropriate interval of battery SOH using the output function, we utilized sigmoid function, denoted in Eq. (5), which originally produces values in the interval [0,1], to customize a function which output values in the interval of [0.7, 1.1], denoted by the output function in Eq. (6).

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

$$\text{output}(x) = \frac{0.4}{1 + e^{-x}} + 0.7 \quad (6)$$

3.3. Evaluation metrics

Since lithium-ion battery SOH estimation is a regression task, we use two common metrics to evaluate the model performance: root-mean-

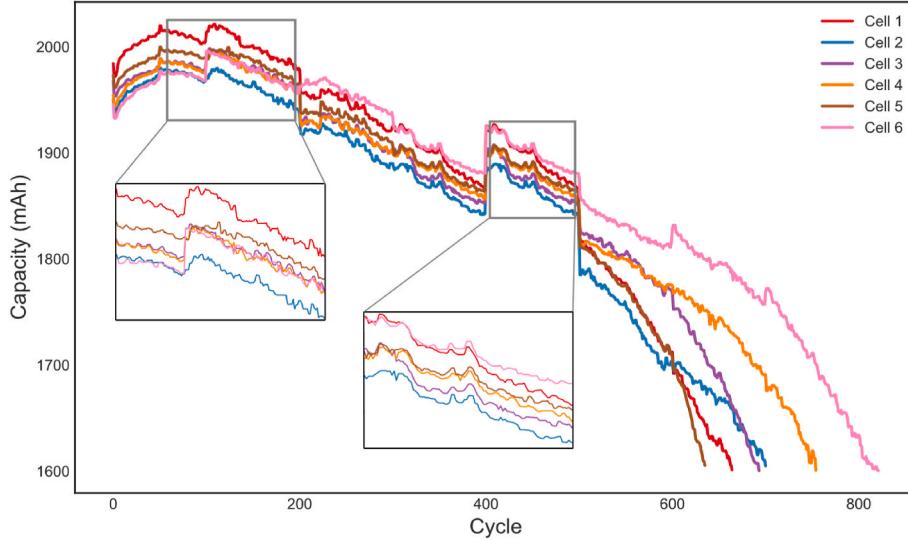


Fig. 4. The capacity curve of the six cells in the battery dataset.

Table 2
Structure of the encoder of our CAE model.

Encoder	
Input	Two-channel input raw current and voltage data
Layer1	1D Conv5-8-1
Layer2	1D Conv5-32-1
Layer3	1D Conv3-64-3
Layer4	1D Conv5-64-1
Layer5	1D Conv5-20-5
Layer6	FC5
Output	Latent abstract health features

Table 3
Structure of the attention-network model for SOH estimation.

Layer	Hyperparameters
Input	Abstract latent features extracted by CAE
query conv	1D Conv1-4
key conv	1D Conv1-4
value conv	1D Conv1-20
CNN layer 1-3	1D Conv3-20 with padding size of 1
CNN layer 4	1D Conv3-1 with padding size of 1
Output	FC1 with customized sigmoid

square error (RMSE), and mean-absolute-percentage error (MAPE). Their definitions are given in Eq. (7):

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (7)$$

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

where y_i represents the target at position i out of total of n , and \hat{y}_i represents the output of the machine learning model also at position i , i.e., the SOH estimation value at position i in our task.

3.4. Preprocessing details

3.4.1. Interception details determined by experiments

Specific voltage range of $[V_1, V_h]$ and current ranges of $[I_l, I_h]$ which

were illustrated in Fig. 2 need to be selected for an efficient battery aging representation, and “laying down” step needs to resample data points in specific length. Thus, the ability of different ranges and resample length to represent SOH needs to be discussed when preprocessing. Some data analysis methods may be useful for data shape determination, but they may require some domain mathematical analysis knowledge. Here we use intuitive experiments without the need of such domain knowledge to find out which part of the data contain the most information.

We make a compare on aging segments with different voltage and current ranges to find out the ranges which can best represent battery aging. We split the voltage into five different segments with 0.3V interval and current into four segments with 1000 mA: ① voltage: [3.7V, 4.0V], [3.75V, 4.05V], [3.8V, 4.1V], [3.85V, 4.15V] and [3.9V, 4.198V]; ② current with the same interval of 1000 mA: [250mA, 1250mA], [500mA, 1500mA], [750mA, 1750mA] and [1000mA, 1980mA]; ③ current with the same end of 250 mA: [250mA, 1250mA], [250mA, 1450mA], [250mA, 1650mA], [250mA, 1850mA] and [250mA, 1980mA] (the full CV charging current). For these nine groups of data, we process them using the same way as what we have described in Section 2.1 step 2, we use the cubic spline interpolation to fit them, and resample data with length of 200. These nine groups of data are then fed into a 3-layer fully connected network model to estimate the SOH value. We create an independent network model for each group, and the hyperparameters of which are consistent and they are listed in Table 4. The RMSE results of these different data are shown in Fig. 5. Each network model was trained and tested 100 times, and the test RMSEs are averaged.

Note that in Table 4, the input neuron number of the network model is the same as the data length. We use data length of 200 when we are choosing the best voltage or current segments thus the input neuron number is 200, but when we choose the best fit length of the data, the length will be different, as well as the input neuron number.

In Fig. 5(a), when start voltage rises from 3.7V to 3.75V the error of SOH estimation is in an increasing trend, but when it continues to rise,

Table 4
The hyperparameters of the 3-layer network model.

Fully connected network	
Input	Segments with different range or fit length
Hidden layer1	FC20
Hidden layer2	FC20
Hidden Layer3	FC1
Output	SOH estimation value

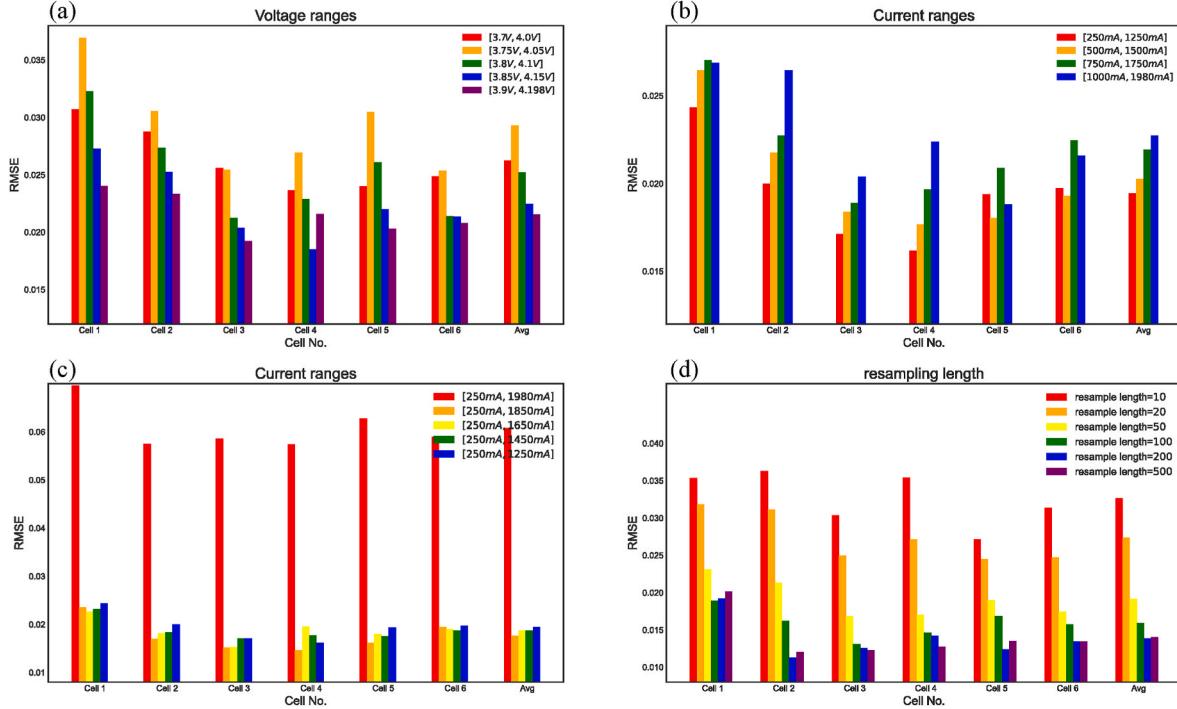


Fig. 5. Estimation RMSE for data selection. Using data of (a) different voltage ranges; (b) current ranges with the same 1000 mA interval length; (c) current ranges with the same end current of 250 mA; (d) resample length.

the errors are generally decreasing, when we have voltage range of [3.9V, 4.198V], the error of almost every battery reaches the minimum, as well as the average error. So, based on the results shown in Fig. 5(a), the voltage range of [3.9V, 4.198V] is selected. And in Fig. 5(b), the estimation errors of each cell is rising in general with the increment of start current when the interval of current is fixed to 1000 mA, and we have the maximum average error for all cells when the current range is [1000mA, 1980mA], while the average error meets the minimum value when the current range is [250mA, 1250mA]. As shown in Fig. 5(c), the current ranges with the same end charging current of 250 mA but different interval current, we can see that the estimation of the full CV stage current of [250mA, 1980mA] is extremely high, and the errors are improved when decreasing the start current from 1980 mA to 1850 mA, and when it further decrease to 1250 mA, the improvement of the errors were very limited. It is speculated that the data points from some cycles may be not fitted properly near the very start current of 1980 mA because of the bad sampling conditions of the original collected data. Combining Fig. 5 (b) and (c), and with the further consideration of the difficulty of data interception and spline fitting, the current range of [250mA, 1250mA] is selected.

After we fit the data with spline curve, we need to resample from the fitted curve. We choose six resample data length to find out which offers the best SOH estimation performance: ①10; ②20; ③50; ④100; ⑤200; ⑥500. The network model in this test has the same structure as the one in the previous test, and the training process is the same as well. In this test, we concatenate whole part of data, i.e., voltage [3.7V, 4.198V] and current [250mA, 1980mA], to feed the network. The test RMSE results are shown in Fig. 5(d).

In Fig. 5(d) we can see that it's not the longer the length or the larger the volume, the better performance or efficiency the data can provide in our SOH estimation task. There is a decreasing trend when resample data length is increasing when $10 < \text{length} < 200$, in general, among each battery, except battery 05. But when the length continues to increase to 500, the estimation error of battery 05, 11, 18 and 22 increases either, as well as the average error. Therefore, we use length of 200 when we resample the data.

According to Fig. 5, this experiment of determining the data ranges and sampling points may have some guiding instructions suggested by the results. Fig. 5(a) (b) indicates that during the CC-CV charging, the ranges near the cut-off voltage or currents may show better correlation with battery aging. In our case, the voltage range of [3.9V, 4.198V], which is the nearest to the cut-off voltage of 4.2V of the CC charging stage in those ranges and the current range of [250mA, 1250mA], which is also the closest to the cut-off current of 40 mA of the CV charging stage show the greatest correlation with battery SOH. Besides the above discussed conclusions, from the perspective of the trend of error shown in Fig. 5(a) (b), some useful information can be further revealed. For example, overall, the error is getting increasingly larger when the ranges are getting farther from the cut-off voltage and cut-off current. However, unlike the current data which reach the poorest average performance in the farthest range of [1000mA, 1980mA], the voltage data reach the poorest performance in the range of [3.75V, 4.05V] rather than the farthest range of [3.7V, 4.0V], showing that there is still some useful information contained in this range. These pieces of information indicate that for battery CCCV charging data, the most informative data for CC charging voltage may be at the beginning and bottom, while the most informative CV charging current may be at the bottom only. Fig. 5 (d) indicates that a proper data resample length is critical for estimation performance, a small number may not contain enough information for accurate SOH estimation while a large number may contain redundant parts and aggravate the computational burden.

Overall, Table 5 shows the voltage and current interception ranges, and the resample length on the curve after our experimental determination.

Table 5
The interception and alignment details.

Voltage segment range	Current segment range	Resample length
[3.9V, 4.198V]	[250mA, 1250mA]	200

3.4.2. Data interception and alignment

Concretely, Fig. 6 (a) and (b) show the CCCV charging stage voltage and current curve at the charging stage of one battery cycle, and the interception schematic of aging segments. The green part on the line respect CC mode, and orange part CV mode. In both figures, the part enclosed by the grey boxes are the aging segments we intercepted, $V_{\text{cut-off}}$ and $I_{\text{cut-off}}$ are cut off voltage and current, the lower bound of the voltage and current is indicated by subscript l and the upper bound by h, the $V_{\text{CC-start}}$ is the beginning charging voltage, and I_{charge} is the constant current at CC stage. Fig. 6 (c) and (d) represent the voltage and current data from different cycles of Cell 1, early cycles are colored by red in (c) and blue in (d) while the late cycles are yellow in (c) and green in (d). We can see the trend that the later the cycle is, the quicker the charging stage will complete. Fig. 6 (e) represents the fitted spline curve of Cell 1,

we can see that the curves of early cycles are at an upper position, which means they take more time for CC and CV charging stage during the same voltage or current range.

4. SOH estimation results and discussions

4.1. Estimation based on different components

We further conduct ablation studies to investigate the effectiveness of different components of our model. Thus, we create three models: ① CAE-CNN, by removing the SA module and use regular CNN to replace it; ② SA, by removing the CAE and directly input the raw data into a CNN with SA module, ③ CNN only, by removing both parts and use only a regular CNN to complete the job, the estimation RMSE and MAPE results

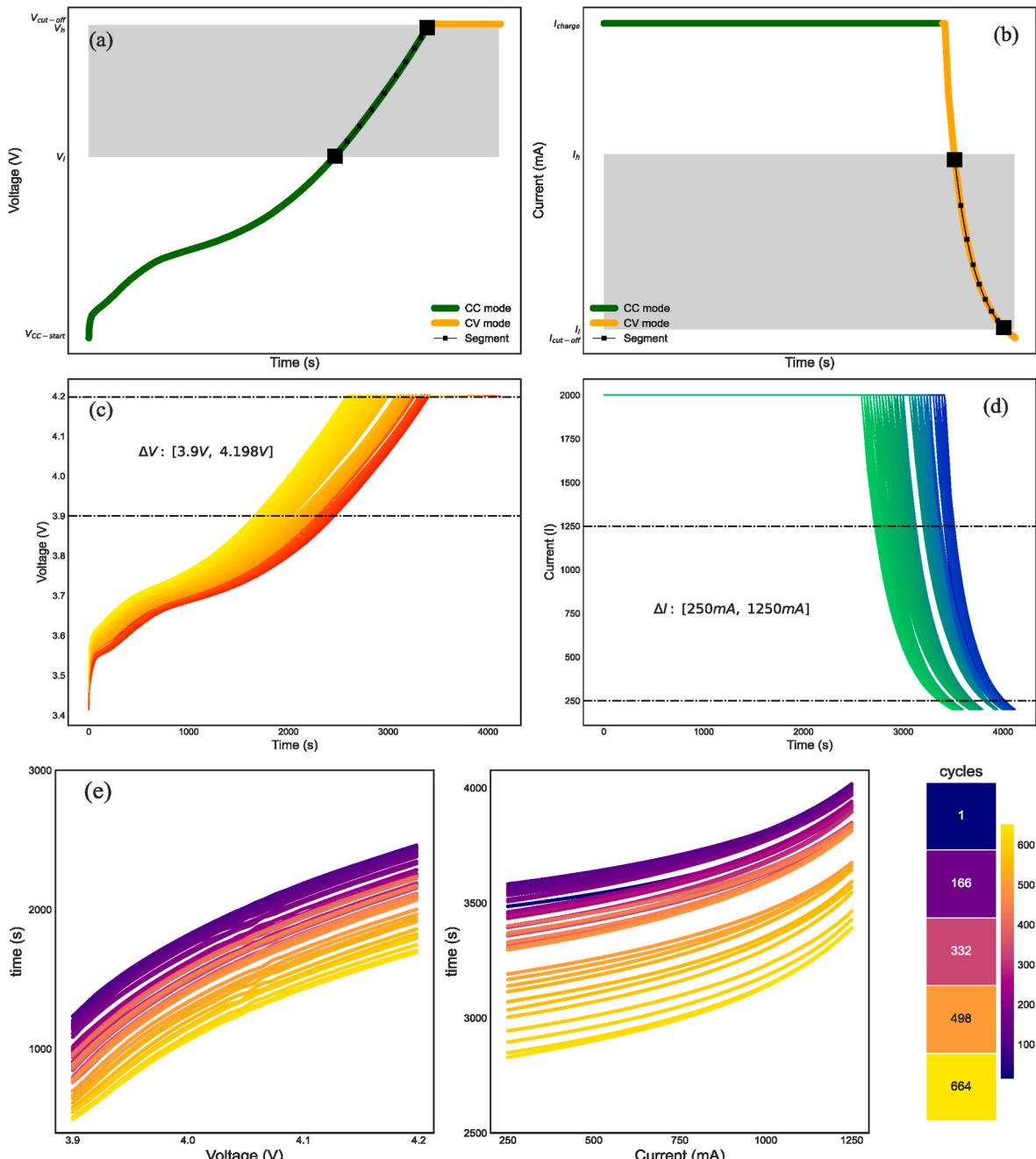


Fig. 6. The intercepted voltage and current. (a) Voltage during charging, (b) Current during charging, (c) charging voltage of all cycle data from Cell 1, (d) charging current segments of Cell 1, (e) intercepted and fitted curves for aging segments.

are presented in [Table 6](#).

In [Table 6](#) we can see that both parts improve the performance in similar extent. With the quantitative measurement using RMSE, when combining two modules, 27.9% of improvement are brought, separately, SA module brings 12.4%, and CAE model 17.8%. Thus, the CAE can automatically extract abstract health features which can help improve the model accuracy, while SA module can find out the relationship among the input features.

The effect of each components may provide some hints for the results shown in [Table 6](#). In our complete model, the preprocessed data is input into the CAE to extract features automatically, and the extracted features are then input into the SA module for further processing. Finally, the subsequent CNN is used for final SOH estimation. Without CAE, the features cannot be extracted, which causes incomplete information mining of the raw data. Without SA, the features extracted by CAE are directly input into a CNN, causes an incomplete model learning of the abstract feature patterns. Combining the both components we can explain the poorest result of the forth column of “CNN” only.

4.2. Comparisons with other approaches

4.2.1. Model structures

In this section, we compare our proposed model with several methods which have been utilized for data-driven SOH estimation tasks in the existing some researches. These models are: ELM [41], DCNN [42], ITDNN [43] and LSTM [44]. They are all data-driven methods without feature engineering, and good at processing raw battery sequential data. Note that the input data and training algorithm for these models are the same as our CAE-SA model. The inputs are the pre-processed raw data which are in shape of Eq. (1), and we also train these models through back propagation algorithm to fit the nonlinearity between input data and battery SOH. The introductions of each models are as follows, whose key hyperparameters are listed in [Table 7](#):

ELM [41]: extreme learning machine, an ELM can simply has one hidden layer. Since first proposed by Huang et al. [45], the ELM has attracted extensive attention for its efficient learning ability.

DCNN [42]: deep convolutional neural network. A deep neural network with convolutional layers can be called as DCNN. With the help of convolutional layers, DCNN is good at processing grid-like data.

ITDNN [43]: input time delay neural network, it has interesting properties such as the ability to learn and handle dynamic information compared to just a regular feedforward neural network. The key hyperparameter in [Table 7](#) means there are 4 time-delay layers with 20 feature channels each.

LSTM [44]: long short-term memory, it is a kind of recurrent neural network algorithm. LSTM is known for its strong ability for processing sequence like data such as time series. The key parameters in [Table 7](#) means there are 4 recurrent layers with 20 hidden feature channels each,

Table 6
The estimation results of the ablation experiment.

		CAE-SA	CAE-CNN	SA	CNN only
RMSE	Cell1	0.0090282	0.0105103	0.0102092	0.0111948
	Cell2	0.0079496	0.0082811	0.0081983	0.0107282
	Cell3	0.0022541	0.0028112	0.0049093	0.0046078
	Cell4	0.0021067	0.0033649	0.0034339	0.0041919
	Cell5	0.0040943	0.0035906	0.003947	0.0044494
	Cell6	0.0035543	0.0045151	0.0045489	0.005049
MAPE	Average	0.0048312	0.0055122	0.0058744	0.0067035
	Cell1	0.0091837	0.0107115	0.0101696	0.0111542
	Cell2	0.0078737	0.0083617	0.0081345	0.010947
	Cell3	0.0018857	0.0023882	0.0046034	0.0043085
	Cell4	0.0017217	0.0032307	0.0031825	0.0038252
	Cell5	0.0039053	0.0033849	0.0036266	0.0040377
	Cell6	0.0033806	0.0044904	0.0043664	0.0048573
Average	Average	0.0046584	0.0054279	0.0056805	0.0065216

Table 7

Key hyperparameters of the compared models.

	ELM	DCNN	ITDNN	LSTM
Input	Data after preprocessing			
Layers	FC 400	3 * 1D Conv3-16 1D Conv3-1	4*Time-delay-20	LSTM 4-20-1
	FC 1000	Average Pooling FC 20		FC 100
		FC 10		FC 10
Output	SOH value			

and 1 output channel. Results.

[Fig. 7](#) shows the estimation results by our proposed CAE-SA model on cell 2 in our dataset and B0006 in NASA dataset. In addition, we show the specific test RMSE and MAPE of every method on each cell of our dataset in [Table 8](#). Note that we run CAE-SA model 10 times and [Fig. 7](#) shows the best results, while [Table 8](#) shows the average error values.

For the results on cells in our dataset which are listed in [Table 8](#), averagely, our CAE-SA model outperforms ELM by 26.2%, DCNN 22.6%, ITDNN 52.1% and LSTM 9.7% in terms of RMSE. ITDNN, which has a similar aim of processing grid-like data as CNN, may be not as flexible as CNN in practical applications. ELM and DCNN have close average error values, but ELM has simpler structure. LSTM reaches very good results which are very closed to our CAE-SA model, but LSTM also has a shortcut of high computational complexity and time cost. Despite good average error, our model has the poorest performance on the second dataset, and is also beaten by ELM and DCNN on the fifth dataset, it is partly because the complex structure of our model leads to overfitting.

4.2.2. Computational complexity

We demonstrate the computational complexity of above methods by presenting the average training time and testing time in [Table 9](#), with the same hardware.

The training time is averaged on each cell and every training epoch, and testing time on each cell. The training time in [Table 9](#) of the first four models are acceptable, ELM has the shortest training time of 304 ms/cell/epoch because of a simple structure, and DCNN can be trained faster than the ITDNN. Although our proposed CAE-SA model has relatively slower training speed and higher computational complexity, we have high accuracy which can match it. However, LSTM has a poorer performance but is either 3.19 times slower than our CAE-SA.

Combining the results in [Tables 8 and 9](#), we conclude that the proposed CAE-SA model has an effective and accurate estimation performance using the raw battery voltage and current data without extracting features manually.

4.3. Comparisons with manual features

To obtain comprehensive results, experiments are further conducted to compare the performance of automatic features and manual features. The automatic features are extracted by the proposed CAE from the “cutting-laying down” preprocessing and subsequently processed by SA module, while manual features are extracted based on domain knowledge, there are four typical while well-performing manual features which were extracted and utilized in many existing methods, they are listed in [Table 10](#) as well as the literatures which utilized them.

We are about to provide comparisons from the aspects of both the feature patterns and SOH estimation performance.

4.3.1. Feature extraction

To provide more intuitive illustrations of the patterns of automatic features extracted by CAE, t-distribution stochastic neighbor embedding (t-SNE) is utilized. The t-SNE is a method to embed high dimensional data into low-dimensional ones, making it a convenient and commonly used algorithm for the visualization of the characteristics and patterns of

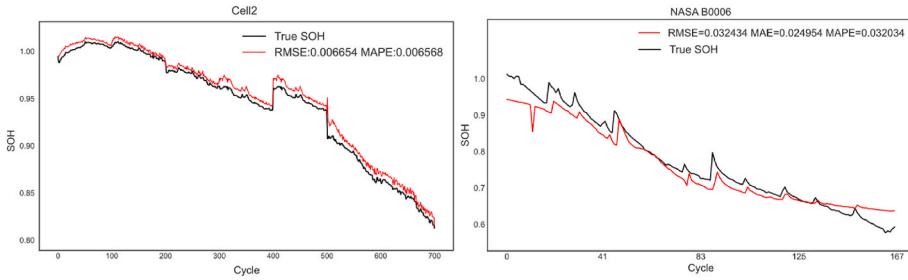


Fig. 7. The SOH estimation results of cell 2 in our dataset and B0006 in NASA dataset by our proposed CAE-SA model.

Table 8
SOH estimation errors of each method.

	CAE-SA	ELM	DCNN	ITDNN	LSTM
RMSE	Cell1 0.008741	0.011893	0.010676	0.018546	0.009841
	Cell2 0.007950	0.006249	0.010995	0.005787	0.007072
	Cell3 0.002254	0.007754	0.004111	0.008782	0.003415
	Cell4 0.002107	0.002479	0.002718	0.006136	0.003538
	Cell5 0.004094	0.003027	0.003232	0.013972	0.004119
	Cell6 0.003554	0.007875	0.005721	0.007298	0.004130
	Average 0.004831	0.006546	0.006242	0.010087	0.005352
MAPE	Cell1 0.009184	0.011859	0.010783	0.018087	0.009754
	Cell2 0.007874	0.005806	0.010994	0.004214	0.006557
	Cell3 0.001886	0.007546	0.003597	0.007738	0.002955
	Cell4 0.001722	0.002104	0.002345	0.005413	0.003105
	Cell5 0.003905	0.002622	0.002907	0.013996	0.003707
	Cell6 0.003381	0.007672	0.005505	0.007056	0.003805
	Average 0.004658	0.006268	0.006022	0.009418	0.004980

Table 9
The average training and testing time of each methods.

	CAE-SA	ELM	DCNN	ITDNN	LSTM
Avg training time (ms)	2715	304	759	1255	8660
Avg testing time (ms)	106	78	88	209	136

Table 10
Manually extracted features.

Feature name	Description
IC peak intensity [47,48]	The $\frac{dQ}{dV}$ value of incremental capacity (IC) curve peak.
V at IC peak [47,49]	The voltage value at the peak of IC curve.
Time stamp [50]	The time of a specific voltage value (4.1V in our case) in CC charging stage.
ΔU under fixed t [51]	Voltage change in a fixed time interval of CC charging stage.

high-dimensional data. The extracted features by CAE are in shape of (*n_cycles*, 20, 5), t-SNE can embed them into lower dimensional space, then we can visualize them on a 2D plate. Furthermore, we extracted 4 manual features, which were typical but well performing features. They are listed in **Table 10**, as well as the works which utilized them. A t-SNE dimension reduction is also applied to them for visualization.

The visualization results of cell 2 in our dataset and B0006 in the NASA dataset are depicted in **Fig. 8**. The meaning of the color of the points in **Fig. 8** is the cycle number, the 2 figures in the first raw, which are **Fig. 8 (a)** (**b**), shows the automatically extracted features of the 2 cells by CAE after t-SNE embedding on 2D plate, while **Fig. 8 (c)** (**d**) illustrate the manually extracted features of both cells.

The embedded features of NASA dataset in **Fig. 8 (b)** (**d**) seem to be more regular than that of our dataset in **Fig. 8 (a)** (**c**), but we can observe obvious patterns with among each feature with the cycle number in both

automatic and manual features of both cells. Considering the clarity of the embedded features on the 2D plate, the manual features of cell2 in **Fig. 8 (c)** have some overlapping areas over some feature points. And by comparing **Fig. 8 (b)** and **(d)**, it is found that although the feature points in each figure are arranged in an oblique line with respect to the cycle numbers, some points in **Fig. 8(d)** are scattered outside the line area, while the points in **Fig. 8(b)** are arranged more neatly. By these pieces of information shown on the embedding 2D plate, we can conclude that the CAE model we utilize can extract equally or even more efficient features for both our dataset and the public NASA dataset.

4.3.2. SOH estimation results

We further make conduct experiments comparing the automatic features extracted by our CAE model and the 4 manually extracted features listed in **Table 10** on their performance of SOH estimation. After features are obtained, the automatic features are subsequently processed by SA module as our proposed approach, while there is an ensemble of four regression models for the processing of the manual features, they are support vector regression model (SVR), Gaussian process regression model (GPR), multi-layer perceptron (MLP), and a CNN model. The average estimation MAPE results are shown in **Table 11**.

Note that for manual features for our dataset, the estimation model of SVR gets poor results of almost 50% of MAPE for each cell, therefore, it is excluded from the model ensemble. By comparing the proposed CAE-SA model with each model for manual features, the proposed model still has some slight performance gap by average, but not much. Moreover, the extraction of manual features requires much domain knowledge of LIB degradation behaviors, and by the errors shown in **Table 11**, we can see that for the same manual features, the selection of the regression model is also crucial for the final estimation accuracy, since the errors reached by the three or four models for our dataset or NASA dataset vary a lot in a large range. However, our proposed CAE-SA method has no such worries, for it is a straightforward approach to estimate by extracting features automatically by CAE without domain knowledge and

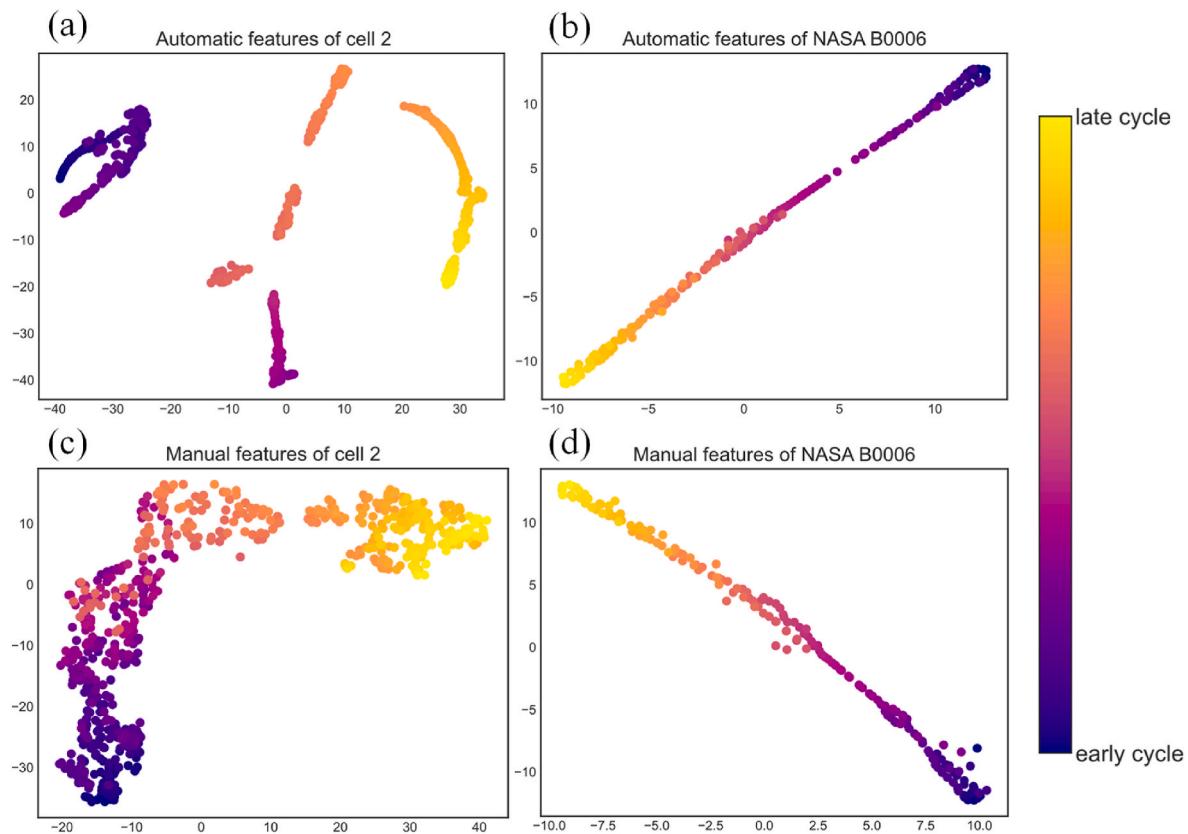


Fig. 8. The embedded features using t-SNE of cell2 in our dataset and B0006 in NASA dataset. (a), (b) Automatic features of by CAE; (c), (d) manual features in Table 10.

Table 11
The SOH estimation average MAPE results in both our and NASA dataset.

Dataset	Our 6 cells		NASA 4 cells		
	Estimation model	Respective	Average	Respective	Average
Automatic	CAE-SA	0.004658	0.004658	0.036853	0.036853
	SVR	\	\	0.056861	0.037686
Manual	GPR	0.003370		0.029806	
	MLP	0.004413		0.032008	
	CNN	0.004752		0.032070	

** The "\\" symbols in the SVR row mean that the SOH estimation errors are too large to be included.

subsequently processing the abstract features by SA module.

5. Conclusions

In this paper, a data-driven approach was proposed for SOH estimation of lithium-ion batteries. In detail, a convolutional autoencoder was proposed to automatically extract features from battery raw data, which significantly save time and labor from conventional manual feature engineering process. Then, a self-attention mechanism was used to further process the extracted high-dimensional features and perform SOH estimation. Specifically, during data preprocessing, a “cutting-laying down” method was used to intercept and align the raw data. By intercepting the raw charging current and voltage data collected from different cycles into specific range, and re-mapping the data using time as independent variable on x axis, the data can be properly preprocessed and degradation information can be retained. Besides, the utilized data ranges and number of resample points were discussed in details. It was found by experiments that the voltage and current ranges which are

nearest to the cut-off value possess the greatest correlation with battery SOH. Extensive experiments were conducted to prove that the proposed CAE was able to extract efficient health features, and the SA module was able to further properly process the automatic features and provide accurate SOH estimation results. The comparing among different parts of the model explored the effect of each model part, and the comparing among different models explored the efficiency and computational complexity of our model and other existing data-driven models. Moreover, the comparing between the automatic features extracted by the proposed CAE model with four typical manual features was also conducted on our dataset and the public NASA dataset. The visualization results of these features after t-SNE embedding on 2D plate have proved that the automatic features have equally efficient representation with manual features for the battery SOH. Furthermore, the SOH estimation results were compared between the automatic features and manual features, it was found that the proposed method was able to reach accurate SOH estimation results on both our dataset and the public NASA dataset. Compared to the manual features which could get slightly better results with an ensemble of regression models, the proposed automatic feature extraction method enjoyed the significant merits of reaching accurate SOH estimation results without the need of domain knowledge for conventional feature extraction process. Quantitatively, our model could provide SOH estimation results with an average RMSE of 0.0048 and MAPE 0.46% on test cells of our dataset, and 3.69% MAPE on the public NASA dataset. In the future work, further research of the SOH estimation with partially charged or discharged battery data will be a featured work. This work may provide solution to the real-world application problems of LIBs which are often under partially charging or discharging situations.

CRediT authorship contribution statement

Yiyue Jiang: Methodology, Software, Writing – original draft. **Yuan Chen:** Writing – review & editing. **Fangfang Yang:** Writing – review & editing. **Weiwen Peng:** Conceptualization, Supervision, 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.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the Key-Area Research and Development Program of Guangdong Province (Project No. 2020B090920002), the Shenzhen Fundamental Research Program (Project No. JCYJ20190807155203586), the Opening Project of Science and Technology on Reliability Physics and Application Technology of Electronic Component Laboratory (Project No. ZHD201909), the Postdoctoral Research Foundation of China (Project No. 2021M703686), the Guangdong Basic and Applied Basic Research Foundation (Project No. 2021A1515110306) and the Fundamental Research Funds for the Central Universities, Sun Yat-sen University (Project No. 22qntd1706, 22qntd1711).

References

- [1] A. Jagadale, X. Zhou, R. Xiong, D.P. Dubal, J. Xu, S. Yang, Lithium ion capacitors (LICs): development of the materials, *Energy Storage Mater.* 19 (May 2019) 314–329, <https://doi.org/10.1016/J.ENSM.2019.02.031>.
- [2] K.A. Severson, et al., Data-driven prediction of battery cycle life before capacity degradation, *Nat. Energy* 4 (5) (May 2019) 383–391, <https://doi.org/10.1038/s41560-019-0356-8>.
- [3] A. Barré, B. Deguilhem, S. Grolleau, M. Gérard, F. Suard, D. Riu, A review on lithium-ion battery ageing mechanisms and estimations for automotive applications, *J. Power Sources* 241 (Nov. 2013) 680–689, <https://doi.org/10.1016/J.JPOWSOUR.2013.05.040>.
- [4] X. Hu, L. Xu, X. Lin, M. Pecht, Battery lifetime prognostics, *Joule* 4 (2) (Feb. 19, 2020) 310–346, <https://doi.org/10.1016/j.joule.2019.11.018>. Cell Press.
- [5] E. Vanem, C.B. Salucci, A. Bakdi, Ø. Å. shei Alnes, Data-driven state of health modelling-A review of state of the art and reflections on applications for maritime battery systems, *J. Energy Storage* 43 (Nov. 2021), <https://doi.org/10.1016/j.est.2021.103158>.
- [6] H. Tian, P. Qin, K. Li, Z. Zhao, A review of the state of health for lithium-ion batteries: research status and suggestions, *J. Clean. Prod.* 261 (Jul. 10, 2020), <https://doi.org/10.1016/j.jclepro.2020.120813>. Elsevier Ltd.
- [7] R. Xiong, Y. Pan, W. Shen, H. Li, F. Sun, Lithium-ion battery aging mechanisms and diagnosis method for automotive applications: recent advances and perspectives, *Renew. Sustain. Energy Rev.* 131 (Oct. 2020), 110048, <https://doi.org/10.1016/J.RSER.2020.110048>.
- [8] 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. 2020) 1–6, <https://doi.org/10.1038/s41467-020-15235-7>, 2020 111.
- [9] H. Rauf, M. Khalid, N. Arshad, Machine learning in state of health and remaining useful life estimation: theoretical and technological development in battery degradation modelling, *Renew. Sustain. Energy Rev.* 156 (Mar. 01, 2022), <https://doi.org/10.1016/j.rser.2021.111903>. Elsevier Ltd.
- [10] A. Bhowmik, I.E. Castelli, J.M. Garcia-Lastra, P.B. Jørgensen, O. Winther, T. Vegge, A perspective on inverse design of battery interphases using multi-scale modelling, experiments and generative deep learning, *Energy Storage Mater.* 21 (Sep. 2019) 446–456, <https://doi.org/10.1016/J.ENSM.2019.06.011>.
- [11] M. Ecker, et al., Calendar and cycle life study of Li(NiMnCo)O₂-based 18650 lithium-ion batteries, *J. Power Sources* 248 (Feb. 2014) 839–851, <https://doi.org/10.1016/J.JPOWSOUR.2013.09.143>.
- [12] Y. Zhang, Y.F. Li, Prognostics and health management of Lithium-ion battery using deep learning methods: a review, *Renew. Sustain. Energy Rev.* 161 (Jun. 2022), 112282, <https://doi.org/10.1016/J.RSER.2022.112282>.
- [13] J. zhen Kong, F. Yang, X. Zhang, E. Pan, Z. Peng, D. Wang, Voltage-temperature health feature extraction to improve prognostics and health management of lithium-ion batteries, *Energy* 223 (May 2021), 120114, <https://doi.org/10.1016/J.ENERGY.2021.120114>.
- [14] W. Allafi, K. Uddin, C. Zhang, R. Mazuir Raja Ahsan Sha, J. Marco, On-line scheme for parameter estimation of nonlinear lithium ion battery equivalent circuit models using the simplified refined instrumental variable method for a modified Wiener continuous-time model, *Appl. Energy* 204 (Oct. 2017) 497–508, <https://doi.org/10.1016/J.APENERGY.2017.07.030>.
- [15] N. Tian, Y. Wang, J. Chen, H. Fang, One-shot parameter identification of the Thevenins model for batteries: methods and validation, *J. Energy Storage* 29 (Jun. 2020), 101282, <https://doi.org/10.1016/J.JEST.2020.101282>.
- [16] G.K. Prasad, C.D. Rahn, Model based identification of aging parameters in lithium ion batteries, *J. Power Sources* 232 (Jun. 2013) 79–85, <https://doi.org/10.1016/J.JPOWSOUR.2013.01.041>.
- [17] Y. Bi, Y. Yin, S.Y. Choe, Online state of health and aging parameter estimation using a physics-based life model with a particle filter, *J. Power Sources* 476 (Nov. 2020), 228655, <https://doi.org/10.1016/J.JPOWSOUR.2020.228655>.
- [18] X. Bian, L. Liu, J. Yan, Z. Zou, R. Zhao, An open circuit voltage-based model for state-of-health estimation of lithium-ion batteries: model development and validation, *J. Power Sources* 448 (Feb. 2020), 227401, <https://doi.org/10.1016/J.JPOWSOUR.2019.227401>.
- [19] J.Z. Kong, D. Cui, B. Hou, X. Liu, D. Wang, New short-long-term degradation model for precise battery health prognostics, *IEEE Trans. Ind. Electron.* (2022), <https://doi.org/10.1109/TIE.2022.3212365>.
- [20] D. Wang, J. zhen Kong, F. Yang, Y. Zhao, K.L. Tsui, Battery prognostics at different operating conditions, *Measurement* 151 (Feb. 2020), 107182, <https://doi.org/10.1016/J.MEASUREMENT.2019.107182>.
- [21] M.F. Ng, J. Zhao, Q. Yan, G.J. Conduit, Z.W. Seh, Predicting the state of charge and health of batteries using data-driven machine learning, *Nat. Mach. Intell.* 2 (3) (Mar. 2020) 161–170, <https://doi.org/10.1038/s42256-020-0156-7>, 2020 23.
- [22] J. Zhu, et al., Data-driven capacity estimation of commercial lithium-ion batteries from voltage relaxation, *Nat. Commun.* 13 (1) (Apr. 2022) 1–10, <https://doi.org/10.1038/s41467-022-29837-w>, 2022 131.
- [23] J. Wang, et al., State of health estimation based on modified Gaussian process regression for lithium-ion batteries, *J. Energy Storage* 51 (Jul. 2022), 104512, <https://doi.org/10.1016/J.JEST.2022.104512>.
- [24] D. Roman, S. Saxena, V. Robu, M. Pecht, D. Flynn, Machine learning pipeline for battery state-of-health estimation, *Nat. Mach. Intell.* 3 (5) (May 2021) 447–456, <https://doi.org/10.1038/s42256-021-00312-3>.
- [25] S. Huang, C. Liu, H. Sun, Q. Liao, State of health estimation of lithium-ion batteries based on the regional frequency, *J. Power Sources* 518 (Jan. 2022), 230773, <https://doi.org/10.1016/J.JPOWSOUR.2021.230773>.
- [26] B. Ma, et al., Remaining useful life and state of health prediction for lithium batteries based on differential thermal voltammetry and a deep-learning model, *J. Power Sources* 548 (Nov. 2022), <https://doi.org/10.1016/j.jpowsour.2022.232030>.
- [27] Y. Ma, C. Shan, J. Gao, H. Chen, Multiple health indicators fusion-based health prognostic for lithium-ion battery using transfer learning and hybrid deep learning method, *Reliab. Eng. Syst. Saf.* (Sep. 2022), 108818, <https://doi.org/10.1016/j.ress.2022.108818>.
- [28] Q. Gong, P. Wang, Z. Cheng, An encoder-decoder model based on deep learning for state of health estimation of lithium-ion battery, *J. Energy Storage* 46 (Feb. 2022), 103804, <https://doi.org/10.1016/J.JEST.2021.103804>.
- [29] G. Liu, H. Bao, B. Han, A stacked autoencoder-based deep neural network for achieving gearbox fault diagnosis, *Math. Probl. Eng.* 2018 (2018), <https://doi.org/10.1155/2018/5105709>.
- [30] L. Chen, et al., Short-term water demand forecast based on automatic feature extraction by one-dimensional convolution, *J. Hydrol.* 606 (Mar. 2022), 127440, <https://doi.org/10.1016/J.JHYDROL.2022.127440>.
- [31] S. Tang, Y. Zhu, S. Yuan, Intelligent fault identification of hydraulic pump using deep adaptive normalized CNN and synchrosqueezed wavelet transform, *Reliab. Eng. Syst. Saf.* 224 (Aug. 2022), 108560, <https://doi.org/10.1016/J.RESS.2022.108560>.
- [32] 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 (Apr. 2021), <https://doi.org/10.1016/j.ress.2020.107396>.
- [33] M. Wei, M. Ye, Q. Wang, Xinxin-Xu, J.P. Twajamahoro, Remaining useful life prediction of lithium-ion batteries based on stacked autoencoder and Gaussian mixture regression, *J. Energy Storage* (2021), <https://doi.org/10.1016/j.est.2021.103558>.
- [34] L. Phan, H. Tran, H. Nguyen, T.H. Trinh, V. Research, ViT5: Pretrained Text-To-Text Transformer for Vietnamese Language Generation, May 2022, <https://doi.org/10.48550/arxiv.2205.06457>.
- [35] N. Kitaev, L. Kaiser, A. Levskaya, G. Research, Reformer: the Efficient Transformer, Jan. 2020, <https://doi.org/10.48550/arxiv.2001.04451>.
- [36] H. Zhang, I. Goodfellow, D. Metaxas, A. Odena, Self-Attention Generative Adversarial Networks, *PMLR*, May 24, 2019, pp. 7354–7363 [Online]. Available: <https://proceedings.mlr.press/v97/zhang19d.html>. (Accessed 9 September 2022).
- [37] F.-J. Tsai, Y.-T. Peng, Y.-Y. Lin, C.-C. Tsai, C.-W. Lin, Stripformer: Strip Transformer for Fast Image Deblurring, Apr. 2022, <https://doi.org/10.48550/arxiv.2204.04627>.
- [38] A. Vaswani, et al., Attention is all you need, *Adv. Neural Inf. Process. Syst.* 30 (2017).
- [39] I. Goodfellow, Y. Bengio, A. Courville, Deep learning [Online]. Available: https://www.google.com/books?id=zh-CN&lpg=&id=omivDQAAQBAJ&oi=fnd&pg=PR5&ots=MNS0gukASY&sig=Y_JWelkD4oMVU3K7-fqaoG4Lsmg, 2016. (Accessed 9 September 2022).

- [40] Z. Chen, C.C. Mi, Y. Fu, J. Xu, X. Gong, Online battery state of health estimation based on Genetic Algorithm for electric and hybrid vehicle applications, *J. Power Sources* 240 (Oct. 2013) 184–192, <https://doi.org/10.1016/J.JPOWSOUR.2013.03.158>.
- [41] H. Pan, Z. Lü, H. Wang, H. Wei, L. Chen, Novel battery state-of-health online estimation method using multiple health indicators and an extreme learning machine, *Energy* 160 (Oct. 2018) 466–477, <https://doi.org/10.1016/J.ENERGY.2018.06.220>.
- [42] S. Shen, M. Sadoughi, M. Li, Z. Wang, C. Hu, Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries, *Appl. Energy* 260 (Feb. 2020), 114296, <https://doi.org/10.1016/J.APENERGY.2019.114296>.
- [43] H. Chaoui, C.C. Ibe-Ekeocha, H. Gualous, Aging prediction and state of charge estimation of a LiFePO₄ battery using input time-delayed neural networks, *Elec. Power Syst. Res.* 146 (May 2017) 189–197, <https://doi.org/10.1016/J.EPSR.2017.01.032>.
- [44] W. Li, N. Sengupta, P. Dechent, D. Howey, A. Annaswamy, D.U. Sauer, Online capacity estimation of lithium-ion batteries with deep long short-term memory networks, *J. Power Sources* 482 (Jan. 2021), 228863, <https://doi.org/10.1016/J.JPOWSOUR.2020.228863>.
- [45] G. Bin Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, *Neurocomputing* 70 (1–3) (Dec. 2006) 489–501, <https://doi.org/10.1016/J.NEUCOM.2005.12.126>.
- [46] J.Z. Kong, D. Wang, T. Yan, J. Zhu, X. Zhang, Accelerated stress factors based nonlinear wiener process model for lithium-ion battery prognostics, *IEEE Trans. Ind. Electron.* 69 (11) (Nov. 2022) 11665–11674, <https://doi.org/10.1109/TIE.2021.3127035>.
- [47] J. Wu, L. Fang, G. Dong, M. Lin, State of health estimation of lithium-ion battery with improved radial basis function neural network, *Energy* 262 (Jan. 2023), 125380, <https://doi.org/10.1016/J.ENERGY.2022.125380>.
- [48] 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 (Nov. 2020), 228654, <https://doi.org/10.1016/J.JPOWSOUR.2020.228654>.
- [49] 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) (Jun. 2022) 7432–7442, <https://doi.org/10.1109/TPEL.2022.3144504>.
- [50] Y. Tan, Y. Tan, G. Zhao, G. Zhao, Transfer learning with long short-term memory network for state-of-health prediction of lithium-ion batteries, *IEEE Trans. Ind. Electron.* 67 (10) (Oct. 2020) 8723–8731, <https://doi.org/10.1109/TIE.2019.2946551>.
- [51] H. Dai, G. Zhao, M. Lin, J. Wu, G. Zheng, A novel estimation method for the state of health of lithium-ion battery using prior knowledge-based neural network and Markov chain, *IEEE Trans. Ind. Electron.* 66 (10) (Oct. 2019) 7706–7716, <https://doi.org/10.1109/TIE.2018.2880703>.