

A Novel State-of-Health Estimation for the Lithium-ion Battery Using a Convolutional Neural Network and Transformer Model

Xinyu Gu ^{a,*}, K.W. See ^{a, c}, Penghua Li ^b, Kangheng Shan ^b, Yunpeng Wang ^c, Liang Zhao ^c, Kai Chin Lim ^c, Neng Zhang ^c

^a Faculty of Engineering, Institute for Superconducting & Electronic materials, University of Wollongong, Innovation Campus, Wollongong, NSW 2500, Australia

^b College of Automation, Chongqing University of Posts and Telecommunications, Chongqing, 400065, PR China

^c Azure Mining Technology Pty Ltd, CCTEG, Level 19, 821 Pacific Highway, Chatswood, NSW 2067, Australia

Abstract

State-of-health (SOH) estimation of lithium-ion batteries is crucial for ensuring the reliability and safety of battery operation while keeping maintenance and service costs down in the long run. This study suggests a novel SOH estimation based on data pre-processing methods and a convolutional neural network (CNN)-Transformer framework. In data pre-processing, highly related features are selected by the Pearson correlation coefficient (PCC). Principal correlation analysis (PCA) is also employed to minimize the computational burden of the estimation model by eliminating redundant feature information. Then, all the features are normalized by the min-max feature scaling method, which will speed up the training process to reach the minimum cost function. After pre-processing, all the features are fed into the CNN-Transformer model. The dataset of the battery from the NASA is employed as a training and testing dataset to build the proposed model. The simulations indicate that the proposed performance, proven by absolute estimation errors for each dataset, is within 1%. The estimation performance index is proven by mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE) are held within 0.55%. These show that the proposed model can estimate the battery SOH with high accuracy and stability.

Keywords: Lithium-ion battery, state-of-health (SOH), convolution neural network (CNN), long short-term memory (LSTM), Transformer

1. Introduction

Battery management systems (BMSs) are widely utilized in various battery powered applications, because they can improve the battery performance, especially during charging and discharging operations. The fundamental functions in a BMS are battery power management, data acquisition, and battery state internal estimation. Those battery internal states estimation, such as state-of-health (SOH) and state-of-charge (SOC), cannot be measured directly so they need to be acquired using some estimation methods [1–3].

The commercialization of portable electrical equipment has led to a growing need for long-life batteries. Due to the batteries' characteristics and electrochemical side reactions, they will inevitably experience gradual degradation across their life cycles. SOH estimation is generally applied to evaluate the degree of battery degradation, because SOH is an index that can reflect the general health condition of a battery and its ability to deliver the specified performance in comparison with its ideal conditions. As an indispensable indicator of a BMS, various approaches have been studied on SOH estimation algorithms. These approaches could be divided into three categories: (1) experimental approaches (e.g., coulomb counting, the open circuit voltage approach, etc.); (2) model-based approaches (e.g., extended Kalman filtering, electrochemical and circuit models, etc.); and (3) data-driven approaches (including deep learning and machine learning methods) [4,5].

Experimental approaches could be further divided into direct and indirect measurement approaches. Direct measurement approaches require a full charging and discharging operations to adjust the battery static capacity, and measurements of the capacity, impedance, and other tests are then used to directly assess battery SOH [6]. The indirect analysis approach is featured by multi-step derivation method and takes health indices associated with degradation of the battery internal resistance or capacity [7]. The experimental approach is straightforward and able to obtain abundant degradation information and accurate SOH estimation results. This kind of approach is time-consuming and costly, and besides, will permanently impair the battery lifespan. Therefore, the experimental measurement methods are more appropriate being done in specialized research laboratory.

Although experimental measurement results are well founded and accurate, online and real-time acquisition of the state of battery health become more desirable for a BMS in real world application scenarios. For the model-based methods, the impedance, capacity, and other parameters are used as indicators to characterize the state of battery health with adaptive filtering algorithms, and then the results are used to quantify the degradation of batteries. Kalman filtering (KF) method has received considerable amount of attentions [8]. Based on this algorithm, the derived ones are commonly used in strongly nonlinear and high-computation models, such as extended KF (EKF), particle filter (PF), unscented KF (UKF), adaptive EKF (AEKF), and fading KF (FKF) [9,10]. Nevertheless, due to its high implementation difficulty and extensive computational costs, the application of this algorithm in online state estimation is limited. So far, many efforts have been made on model simplification based on the traditional pseudo-two-dimensional (P2D) model [11–13]. Various simplification models such as single particle model (SPM) with lower computational demand and acceptable accuracy have been proposed and applied in state estimation [14,15].

Data-driven methods are faster and involve fewer complex mechanics than model-based methods [16]. The data-driven model behaves as a “black box” that utilizes routinely monitored and historically collected system operating data, such as temperature data, vibration signatures, and current measurements, to simulate the complex relationship among the external parameters of batteries [17]. Over the past few decades, approaches such as sparse Bayesian predictive modelling (SBPM) [18] and machine learning methods, such as random forest (RF) [19], support vector machine (SVM) [20], and support vector regression (SVR) [21], have been utilized to trace battery capacity degradation. In these days, studies based on neural networks (NN) have been more and more prosperous. For example, compared with traditional feed forward neural network (FNN), recurrent neural network (RNN) is an efficient method among all the neural networks to handle time-sequential data as its structure uses the hidden neurons to increase the circular connection due to its simplicity to extract and update the correlations of time-sequential data [22,23]. For instance, You et al. [24] implemented a recurrent neural network (RNN) to estimate lithium-ion battery SOH using measurable battery signals, and they achieved high accuracy, flexibility, and noise robustness. RNNs are limited for modeling temporal dependencies due to the vanishing or exploding gradient problem, which will have a negative effect on SOH estimation accuracy. Long short-term memory (LSTM) as an improved structure is applied to overcome this problem by adding additional interactions per module (or cell), which makes it well suited to remembering inputs over a long period of time [25,26]. Ma et al. [27] used differential-evolution grey wolf optimizer

(DEGWO) to improve the model's global search ability and combined it with long short-term memory to obtain accurate predictions for different kinds of batteries. Li et al. [28] implemented a variant attention-based spatial-temporal – long short-term memory (AST-LSTM) neural network to track cell states through a fixed connection actively by simultaneously determining old and new data, which can achieve a lower average root mean square error (RMSE) and conjunct error. Based on that work, Li et al. [29] further added a convolutional neural network (CNN) and proposed an automatic framework, CNN-ASTLSTM, which is an end-to-end prognostic framework that can automatically extract hierarchical features during battery degradation and optimize the model hyperparameters. The CNN-ASTLSTM can save the cost of manual modelling and reduce the impact of manual intervention on SOH prediction accuracy, as well as achieving the best trade-off performance among all recent related studies.

Although the previous estimation methods can provide accurate battery SOH estimation, it is a huge challenge to learn long-term dependencies due to their limited scalability for modeling long sequences and also the time-consuming training process. For neural network, the ability to learn a long dependency is mainly affected by the length of the path that the forward and backward signals must traverse. Furthermore, the efficiency for capturing long-range dependencies would be affected by deeper layers, due to the increasing lengths of paths between components in the sequence.

Considering the limitation on modeling long sequences, an attention mechanism is proposed, which can extract features from the whole sequence with a weighted sum of all the previous input sequence states and assigns greater weight or importance to a certain element of the input for each element of output [30,31]. However, each sequence can only be treated one element at a time, so it is necessary to wait until the completion of $t-1$ steps to process the t^{th} step. As a result, the attention mechanism is very time consuming and computationally inefficient when dealing with a large amount of data [32].

Transformer is a class of sequence transduction models that eschews recurrence and alternatively, relies totally on the attention mechanisms to find global dependencies between the input and output using encoder-decoder architectures [33]. To overcome the problem of time-consuming training, Transformer models introduce position feature embeddings to indicate the absolute or relative position information of data in the sequence, so that features can be described by positional encodings instead of dependencies. Transformer models achieve efficient sequence learning with the highly parallelizable self-attention mechanism, which can easily provide relationships among different features appearing in different locations. Unfortunately, Transformer tends to ignore local feature details, which decreases the discriminability within limited timestamps [34].

Our contributions of this research are summarized as follows:

- (1) Three data preprocessing methods, which are PCC, PCA, and feature scaling, are employed to couple with the proposed CNN-Transformer neural networks. Applying these preprocessing methods can reduce dimensionality and complexity of the model, thus further reduce the model computational burden and improve the performance of the proposed method.
- (2) Transformer was originally applied in natural language processing and widely used in image processing. In this research, we firstly apply the Transformer structure for time series problem estimation because its attention mechanisms and positional encodings can help handle sequential problems. The results of the experiment show that the Transformer can deal with long-term dependencies.
- (3) CNN is good at collecting local variables hierarchically via its convolutional operations and retains all the local cues as feature maps. Transformer can be used to aggregate global properties among the condensed reinforce embeddings through its self-attention mechanisms. CNN-Transformer inherits the structure and generalization advantages of both CNNs and Transformers and predict battery SOH with high accuracy. The experimental results demonstrate that the proposed method has great potential to handle sequential estimation.
- (4) Data visualization. The process of PCA is visualized by showing the dynamic changes of the dimensionality

reduction. Additionally, the dynamic process of feature extraction is also visualized by using activation map. By visualizing time series problems, the sequential data are transferred into a visual context, such as a graph or map, to make it easier to access large amounts of data and pull insight from.

In this research, a novel neural network structure, termed CNN-Transformer, is proposed to couple CNN-based local features with transformer-based global variables for battery state estimation. The arrangement of this paper is as follows. The framework of CNN, Transformer, and the CNN-Transformer are demonstrated in Section 2. The NASA dataset used for estimation experiment and data preprocessing based on PCC, PCA and feature scaling are explained in Section 3. In Section 4, the offline training and online testing results of CNN-Transformer and other three methods are analyzed. Finally, the conclusions are summarized in Section 5.

2. Methodology

2.1 Convolutional Neural Network (CNN)

As a particular type of multilayer perceptron (MLP), CNN is constructed in three layers, which are input layers, output layers, and hidden layers. The input layer is used to transfer the original data to the first hidden layer. The output layer is responsible for producing given outputs for the next program. The hidden layers comprise the fully connected layer, the max-pooling layer, and the convolutional layer [35].

Convolutional layers are the main block of a CNN, acquiring local properties from higher layer inputs and passing all information to the lower layers for more complex features. The outcomes of the vector o output for the first convolutional layer and it can be represented by following equation:

$$o_{ij}^1 = \sigma \left(b_j^1 + \sum_{f_v=1}^M w_{f_v j}^1 x_i + f_v^o - 1, j \right) \quad (1)$$

where σ , b_j , and w represents the sigmoid activation function, the bias for the j feature map, the weight of the kernel, respectively. f_v and x are the filter index and the power production input vector, respectively. Similarly, the outcome of the vector o output from the l convolutional layer can be expressed as follows:

$$o_{ij}^l = \sigma \left(b_j^l + \sum_{f_v=1}^M w_{f_v j}^l x_i + f_v^o - 1, j \right) \quad (2)$$

Max-pooling layers are utilized for dimensionality reduction of the representation, and thus, further reduce the computational burden of the model. The operation of max-pooling layer is given by:

$$p_{ij}^l = \max_{r \in R} y_i \times T^{l-1} + r, j \quad (3)$$

where R is the pooling size. T is the step that determines the distance for input data area is to be moved and, which is less than the input size y . Fully connected layers connect every neuron in one layer to every neuron in the output layer [36,37].

2.2 Transformer

Transformer follows an overall architecture including pointwise, stacked self-attention, and fully connected layers for both the encoder and decoder. The overall architecture was simplified by integrating some functions. By dispensing with recurrence and convolutions, Transformer model here has revolutionized the implementation of attention mechanisms by relying solely on the self-attention mechanism that is composed of scaled-dot-product attention and multi-head attention.

The scaled dot-product attention proposed by Vaswani et al. by firstly computes a dot product for each query, q , with all the keys, k . It subsequently divides each result by $\sqrt{d_k}$ and proceeds to apply a softmax function:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

The equation for the multi-head attention mechanism is built on the above scaled dot-product attention mechanism, as shown below:

$$\text{Multihead}(Q,K,V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (5)$$

here, each head_i , $i = 1, \dots, h$, implements a single attention function characterized by its own learned projection matrices as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (6)$$

where q and k represent vectors of dimensions, and d_k contains the queries and keys, respectively. v denotes a vector of dimensions and d_v contains the values. Q , K , and V which denote matrices that pack together sets of queries, keys, and values, respectively. W^Q , W^K , and W^V denote projection matrices which are employed to generate different subspace representations of the query, key, and value matrices, respectively. W^O denotes a projection matrix for the multi-head output. $W_i^Q \in R^{d_{model} \times d_k}$, $W_i^K \in R^{d_{model} \times d_k}$, $W_i^V \in R^{d_{model} \times d_v}$, and $W^O \in R^{hd_v \times d_{model}}$.

Besides, each layer in the encoder and the decoder of Transformer contains a fully connected feed-forward network (FNN), which is applied to each position separately and identically.

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (7)$$

Since the Transformer model has no recurrence and no convolution, “positional encodings” (PE) are applied to the input embeddings with information about the relative or absolute positions of tokens in the sequence.

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \quad (8)$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad (9)$$

where pos is the position and i represents the dimension [38,39].

2.3 Overview of CNN-Transformer structure

Time series problems can be regarded as a set of observations made continuously over specific time intervals where each random observed variable has its own distribution. Local properties and global representations are two significant complementary parts, which have been broadly studied in the development of techniques for dealing with time series. The local features and their descriptors of each sample model non-stationary over time with a stochastic process, and the global features encodes the time-independent characteristics at long distance. In deep neural network, CNN is used to collect local variables hierarchically via its convolutional operations and retains all the local cues as feature maps. Transformer aggregates global properties among the condensed reinforce embeddings through its self-attention mechanisms [40,41]. In this research, to take the greatest advantage of local and global properties, a concurrent network framework termed CNN-Transformer has been proposed, as shown in Fig. 1.

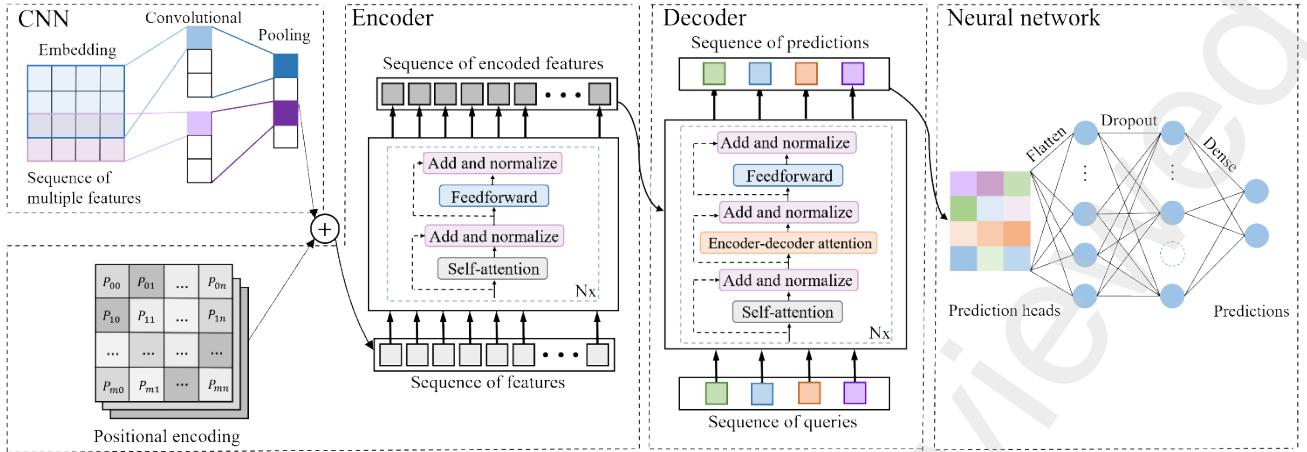


Fig. 1. The proposed CNN-Transformer framework.

Considering the complementarity of the two CNN-style and Transformer-style features, within CNN-Transformer, we progressively extract local features from the CNN branch to enrich the local details. After that, Transformer is used to reinforce the global perception capability of this model. In Transformer, positional encoding is used to describe the position of a datapoint in a sequence, so that each position is assigned a unique representation between 0 and 1. Then, encoder layers are responsible for generating encodings that contain information about which parts of the inputs are relevant to each other. The output from the last encoder layer is the input to the decoder of Transformer. Then the decoder receives the output of the encoder together with the output of the decoder at the previous time step to generate an output sequence. Considering the data capability and feature misalignment between CNN and Transformer, all the data in both structures are adjusted into a unified format. During model training, cross-entropy losses are utilized to couple the two style variables for both the CNN and Transformer. Such fusion procedure could significantly enhance the global perception capability of local variables and the local details of global properties. After the CNN and Transformer neural networks, flatten layers, dropout layers, and dense layers are coordinated to process the outcome from decoder blocks and achieve the final predictions. Here, flatten layers merge all multidimensional input into one-dimensional, so that all the data can be effectively passed to every single neuron of the model. Dropout layers ignore a set of neurons randomly to prevent model overfitting. The neuron of dense layer receives output from every neuron of its preceding layer.

3. Numerical experiment

3.1 NASA dataset

The studies of SOH prediction of lithium-ion battery cells come with the expense of large datasets that consist of multiple cycles of repeated charging and discharging procedures on the cells to validate the performance and accuracy of the proposed technique. In this work, the database of cycling profiles of lithium-ion cells has been extracted from the NASA Ames Prognostics Centre of Excellence. The indicated battery number corresponds to the respective battery dataset in the NASA database. Each of the battery cells was charged through a constant current – constant voltage (CC-CV) procedure with the upper voltage at 4.2 V until the current was reduced to 20 mA. For the discharging cycle, the battery cells were discharged with constant and pulse current waveforms until each of the cells reached its respective cut-off voltage. The number of cycles for each battery was determined by the percentage of faded capacity with respect to the individual corresponding rated capacity, which is denoted as the initial capacity. The percentage is set to 30% for each battery cell regardless of its initial capacity.

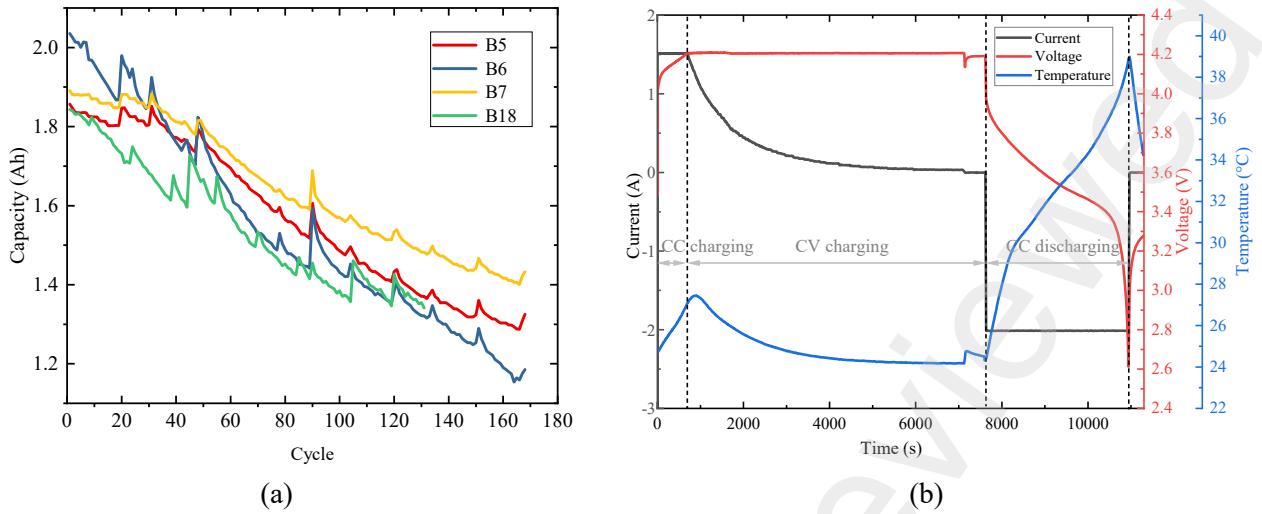


Fig. 2. (a) Capacity degradation curves of dataset. (b) Charging-discharging and temperature curves of dataset.

3.2 Data pre-processing

3.2.1 Feature selection

Pearson correlation coefficient (PCC) analysis provides a measure of the linear relationship between two variables. It represents the ratio between the covariance of two variables and the product of their standard deviations [42]. The PCC is calculated as follow:

$$PCC = \frac{\sum_{i=1}^n (z_i - \bar{z})(q_i - \bar{q})}{\sqrt{\sum_{i=1}^n (z_i - \bar{z})^2} \sqrt{\sum_{i=1}^n (q_i - \bar{q})^2}} \quad (10)$$

where z_i is the values of the x-variable in a sample, \bar{z} is the mean of the values of the x-variable, q_i is the values of the y-variable in a sample, and \bar{q} is the mean of the values of the y-variable [43].

The typical values for the correlation coefficient are ranged from -1.0 to 1.0. An absolute value of exactly 1.0 implies that a linear equation describes the relationship between variable X and variable Y perfectly, with all data points lying on a line. The value of zero implies that there is no linear dependency between the variables. If the correlation coefficient is greater than zero, as denoted by a positive integer, the relationship is linearly dependent. Conversely, if the value is less than zero, as denotes by a negative integer, the relationship is inversely proportional. The input selection for the SOH model is conducted through PCC analysis with five features extracted from each discharging cycle, which are capacity (Ah), output current (A), terminal voltage (V), sampling time (s), and temperature (°C). These features are then evaluated quantitatively with respect to each other for their linear dependency with the distributed scale of five different magnitudes of dependency, which are categorized into extremely strong (1-0.9), strong (0.89-0.7), moderate (0.69-0.4), weak (0.39-0.1), and negligible (0.1-0) correlations [44]. The integers denoted are in absolute value which means the same representation regardless of the sign. The permutation of the seven features provides a total number of 25 correlation coefficients, as illustrated in Fig. 3.

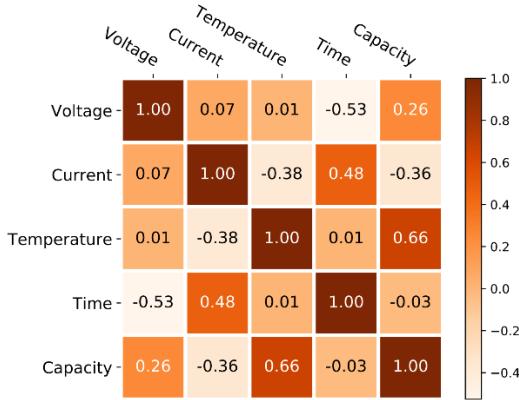


Fig. 3. PCC analysis of all five features.

Taking the battery capacity, which is the target of our prediction, the PCC analysis shows that it has a moderate correlation with temperature (0.66), a weak correlation with current (0.36) and voltage (0.26), and a negligible correlation with the sampling time (0.03). Through the above PCC analysis, we take the terminal voltage, output current, temperature, and capacity as the four input features of the prediction model to estimate battery SOH.

3.2.2 Dimensionality reduction

To avoid model overfitting and reduce the computational burden of neural networks during offline training process, principal component analysis (PCA) is used in this exploratory data analysis can reduce the dimensionality of large datasets by get rid of unrelated information while maintaining most of the relevant information. PCA is designed to find the directions of maximum variance in high-dimensional data space and then projects it onto a new subspace with equal or fewer dimensions than the original one [45].

Let the D -dimensional training set X denote the raw data matrix and the new (lower) dimensionality d (with $d \leq D$). The columns of X are firstly introduced for “autoscaling”. Autoscaling means adjusting the value of X to have zero mean and unit variance by dividing each column by its standard deviation. This is because variables are measure with various means and standard deviations in different units. Autoscaling will put variables on an equal basis for the analysis. Eq. (12) gives the covariance matrix of X :

$$Cov(x) = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x}_i)(x_i - \bar{x}_i)^T \quad (11)$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (12)$$

Then two algorithms, the singular value decomposition (SVD) or Nonlinear Iterative Partial Least Squares (NIPALS) are often used to decompose matrix X and obtain the eigenvectors $\xi_1, \xi_2, \dots, \xi_D$ and their corresponding eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_D$. Note that the eigenvalues are sorted, such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_D \geq 0$. For any $x \in R^D$, its new lower dimensional representation is

$$y = (\xi_1^T(x - \bar{x}), \xi_2^T(x - \bar{x}), \dots, \xi_d^T(x - \bar{x}))^T \in R^d \quad (13)$$

and the original x can be approximated as:

$$x \approx \bar{x} + (\xi_1^T(x - \bar{x}))\xi_1 + (\xi_2^T(x - \bar{x}))\xi_2 + \dots + (\xi_d^T(x - \bar{x}))\xi_d \quad (14)$$

Therefore, the PCA method reduces the original set of variables with D components to d principal components [46].

In this analysis, the maximum likelihood estimation (MLE) is the standard employed to determine the

approximation dimension d for our training data. Consider D-dimensional time series training set X and one-dimensional time series $x = (x_t)_{t=0}^{N-1}$. Given a model with parameter values θ , the task for a predictor is to output the next value $\hat{x}(t+1)$ conditional on the series' history, $x(0), \dots, x(t)$. This can be done by maximizing the likelihood function as:

$$p(x | \theta) = \prod_{t=0}^{N-1} p(x(t+1) | x(0), \dots, x(t), \theta) \quad (15)$$

Also, in this research, the estimation results set \hat{X} should also satisfy:

$$\frac{\frac{1}{N} \sum_{i=1}^N \|x_i - \hat{x}_i\|}{\frac{1}{N} \sum_{i=1}^N \|x_i\|^2} \leq 0.01 \quad (16)$$

As mentioned in section 3.1, four features (terminal voltage, output current, temperature, and capacity) are selected by PCC. Through PCA and the MLE method, which is obtained by the PCA built-in function scikit-learn with the ‘components’ in the function set to ‘mle’. After PCA, the number of features becomes three. Notably, the three features are not three of the four original input features, but new features that have been numerically transformed. Fig. 4 shows positional change before and after PCA. To make it clear, for each feature, 150 points are selected out of the total 420,000 points. After PCC and PCA, the total number of datapoints is reduced from 2,098,800 to 1,259,280.

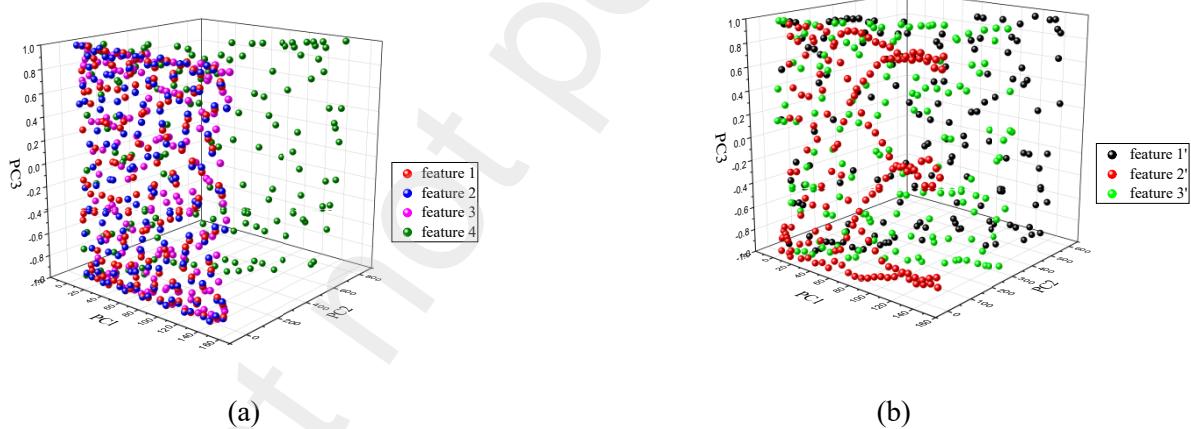


Fig. 4. (a) The positions of the four original features before PCA in three-dimensional PC space. (b) The positions of the three features after PCA in three-dimensional PC space.

3.2.3 Feature scaling

After feature selection and dimensionality reduction, we use the feature scaling method to normalize all features in the dataset into the interval of [0,1], which will help the optimization mechanisms of our model to quickly reach the minimal cost function and improve the performance of model training. Here, the max-min scaling method is used as our feature scaling method as follows:

$$x_{scaled} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (17)$$

where x_{scaled} is the scaled data, x_i is the raw data, x_{min} and x_{max} are the minimum and maximum values of the raw data [47].

3.3 SOH prediction based on CNN-Transformer NN

The SOH of a battery is defined as the ratio of the maximum charge to its rated capacity:

$$\text{SOH} = \frac{Q_{\max}}{C_r} \times 100\% \quad (18)$$

where Q_{\max} is the maximum charge available from the battery and C_r is the rated capacity. If the capacity is decreased to 80% of the initial rated capacity, the battery should be replaced. Based on this definition, estimating SOH is equivalent to estimating capacity [48].

3.3.1 Weight optimization

Weight optimization is used to select model parameters. The purpose of dataset training is to find the proper parameter values that can minimize the cost function (given by Eq. (19)), and consequently improves the performance of our model [49]:

$$E(\omega) = \frac{1}{N} \sum_{t=0}^{N-1} |\hat{x}(t+1) - x(t+1)| + \frac{\gamma}{2} \sum_{l=0}^L \sum_{h=1}^{M_{l+1}} (\omega_h^l)^2 \quad (19)$$

where ω_h^l is the weight of each layer and $\hat{x}(t+1)$ represents the forecast of $x(t+1)$ using $x(0), \dots, x(t)$. Considering too large weights, the L2 regularization method with regularization term γ are added to avoid model overfitting. After weight optimization, optimal parameters are saved for further dataset testing. A standard weight optimization is based on gradient descent, where one incrementally updates the weights based on the gradient of the error function:

$$\omega_h^l(\tau+1) = \omega_h^l(\tau) - \eta \nabla E(\omega(\tau)) \quad (20)$$

for $\tau = 1, \dots, T$, where T represents the number of training iterations. η is the learning rate of the model. To this end, each iteration τ should contain a forward run in which one computes the forecasted vector \hat{x} and the corresponding error $E(\omega(\tau))$, and a backward pass in which the gradient vector $\nabla E(\omega(\tau))$, the derivatives with respect to each weight ω_h^l , is computed and the weights are updated according to Eq. (20). In this research, the weights of the model ω_h^l are updated by Adam gradient descent [50,51]. The proposed model is trained with Adam gradient descent with a learning rate of 0.00055, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = \text{None}$, $\text{decay} = 0$ and $\text{amsgrad} = \text{False}$. For all tested models in this work, the cost function is categorical cross entropy. Then the model achieving the lowest training loss is adopted as the best model, and its performance is evaluated on the testing set. Despite an overfitted configuration, it is revealed that our model CNN-Transformer generalizes quite well. Unlike other benchmarks, our experiment precludes the cross-validation and hyperparameter tuning to provide a most unbiased baseline. These settings also largely reduce the computational complexity for training and deploying purposes of models.

3.3.2 CNN-Transformer estimation model

According to Eq. (18), estimating the battery SOH is equivalent to estimating the capacity. The on-line estimation of capacity is represented as:

$$\hat{c}_{ij} = \text{cnn_transformer}([x_{ij,t}, x_{ij,t+1}, \dots, x_{ij,t+n+1}] | c_{ij}) \quad (21)$$

where, for any specific cell (#5, #6, #7, or #18), \hat{c}_{ij} is the online estimated capacity of the j^{th} cell in i^{th} cycle, $x_{ij,t}$ is the online observed vector (any of voltage, current, and temperature) at sampling time t , m is the length of $x_{ij,t}$, and $n \in [1, m]$ is the length of the sliding window.

Algorithm SOH estimation

$$\text{Input: } X = \begin{bmatrix} x_{1j,t}, x_{1j,t+1}, \dots, x_{1j,t+n+1} & c_{1j} \\ x_{2j,t}, x_{2j,t+1}, \dots, x_{2j,t+n+1} & c_{2j} \\ \vdots & \vdots \\ x_{mj,t}, x_{mj,t+1}, \dots, x_{mj,t+n+1} & c_{mj} \end{bmatrix}$$

Output: SOH

- 1: pre-process(X) (Pearson correlation coefficient, principal component analysis, and feature scaling)
- 2: function TRAIN(Y)
- 3: for number of training iterations do
- 4: model = send Y to proposed LSTM, Transformer, CNN-LSTM, and CNN-Transformer for training
- 5: calculate loss function
- 6: end for
- 7: return model
- 8: end function
- 9: split X into **training** (60%, 70% and 80%) and **testing** datasets (40%, 30% and 20%)
- 10: model = Pre-process(**training** and **testing** datasets)
- 11: model = TRAIN(**training**)
- 12: calculate average estimation loss value to determine various hyperparameters
- 13: save the best model by weight optimization method and optimizers is Adam with learning rate=0.00055, $\beta_1=0.9$, $\beta_2=0.999$, $\epsilon=None$, decay=0.0, amsgrad=False.
- 14: load the optimal weight of the best model
- 15: $\hat{c} = \text{model.predict}(\text{testing})$
- 16: calculate SOH by Eq. (18)
- 17: evaluate and compare the estimation results by MAE, MAPE, RMSE, and R^2

In the CNN layer, to match the requirement of 3D input, the 2D data (none, 1980) are reshaped into 3D data (none, 660, 3). Here, “none” denotes the number of pieces of data, “660” denotes the length of data features, and “3” denotes the number of features selected by pre-processing. Notably, the rectified linear unit (ReLU) is the most common activation function used in CNN. Then, the output is operated on by the max-pooling layers, where “34” is the number of kernel and “7” is the pooling window size. In Transformer, to avoid overfitting, 10% of the datapoints are dropped out (dropout ratio = 0.1). The first add layer is used to combine the output information from max pooling and the attention layer, while the second add layer combines the data from the past dropout layer and the normalization layer. The flatten layer is used to reduce the dimensionalities and after that the shape of datasets is processed into (none, 3196). The dense layer is used to extract the correlation between the previously extracted features in the nonlinear changes of the dense layer and finally map them to the output. The Gaussian error linear unit (GELU) activation function. In the end, the Dropout_2 layer is connected by dense layer to obtain the prediction results.

layer type	activation layer	output shape
Input		(none, 660, 3)
Conv-1D	ReLU	(none, 660, 34)
Max-pooling-1D		(none, 94, 34)
Multi-head-attention		(none, 94, 34)
Add_1		(none, 94, 34)
Layer-normalization_1		(none, 94, 34)
Dense_1	GELU	(none, 94, 34)

Dropout_1	(none, 94, 34)
Add_2	(none, 94, 34)
Layer-normalization_2	(none, 94, 34)
Flatten	(none, 3196)
Dropout_2	(none, 3196)
Dense_2	(none, 1)
Dropout_2	(none, 1)

Table 1. The numerical changes of each layer in the CNN-Transformer model.

4. Results and discussion

4.1 The evaluation criteria

To evaluate the performance of the suggested model, four different evaluation indicators are employed, which are root mean square error (RMSE), mean absolute percentage error (MAPE), mean absolute error (MAE), and coefficient of determination (R^2). MAE is the average of the absolute difference between the estimation and the reference value of SOH, which is employed to measure the average magnitude of errors of the proposed method. MAPE represents the relative percentage error between the prediction and the reference value. RMSE indicates the deviation of the estimation value and the reference value, which is used for evaluation of the quality of estimation [52]. R^2 is used for the indication of how well the regression model fits the observed data, with the value range of [0,1]. MAE, MAPE, RMSE, and R^2 are given as:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (22)$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (23)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (24)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (\bar{y} - y_i)^2} \quad (25)$$

where y_1, y_2, \dots, y_n are the actual values, $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$ are the predicted values, \bar{y} is the mean of y_i , and n is the number of testing samples [53].

3.2 Compared with other methods

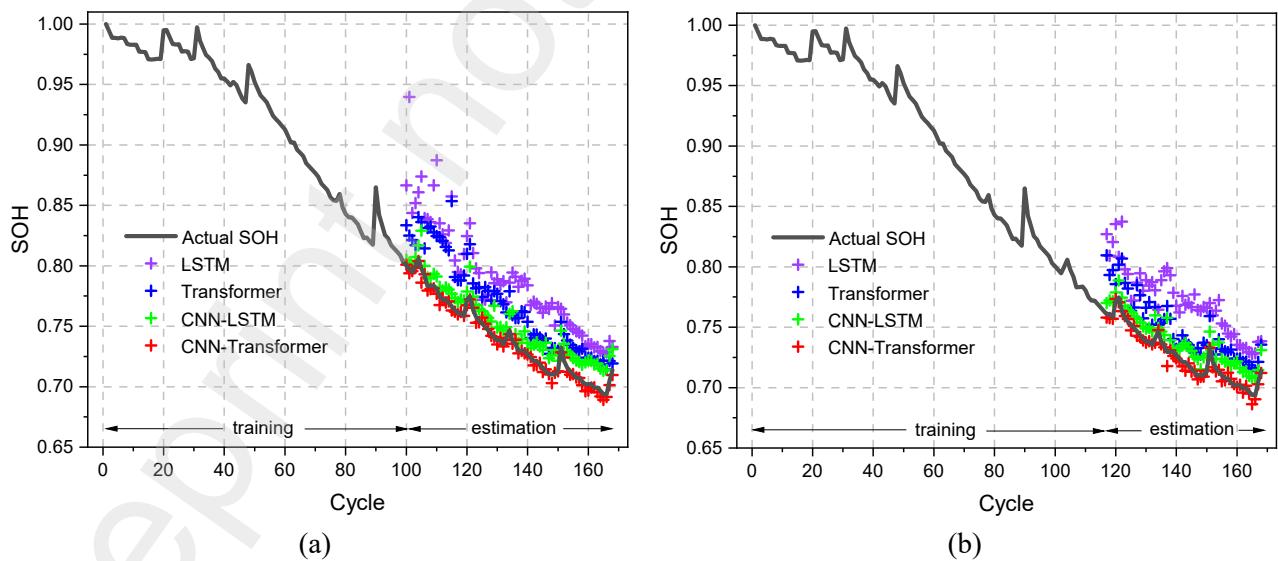
As above mentioned in section 3.1, four batteries data, labelled as B5, B6, B7, and B18, were sourced from NASA and utilized to validate the prediction performance of the CNN-Transformer method. In our experiment, the full dataset with a proportion of 60%, 70%, and 80% were adopted for offline training while the rest of the dataset (40%, 30% and 20%) were used for online testing. To evaluate model robustness and effectiveness, other three methods

LSTM, Transformer, and CNN-LSTM were employed to estimate battery SOH through the same offline training strategy.

Table 2 shows the evaluation criteria for the mentioned methods of SOH estimation for B5. Compared with other three methods, CNN-Transformer model has higher accuracy regarding the better estimation of SOH reference values. To further assess the models, Fig. 5 shows the SOH estimation from this model together with the relative errors of each cycle. As illustrated in Fig. 5(a)-(c), the predicted SOH value is in line with the SOH reference value, demonstrating the efficacy of the CNN-Transformer method. Moreover, Fig. 5(d)-(f) show all the absolute errors of each cycle are within 1%, proving that CNN-Transformer is the most precise method to estimate SOH.

Battery	Method	MAE	MAPE	RMSE	R^2
B5 (60%)	CNN-Transformer	0.0027	0.0037	0.0034	0.9987
	CNN-LSTM	0.0226	0.0366	0.0236	0.9402
	Transformer	0.0287	0.0385	0.0301	0.8125
	LSTM	0.0450	0.0610	0.0490	0.7277
B5 (70%)	CNN-Transformer	0.0026	0.0036	0.0028	0.9974
	CNN-LSTM	0.0135	0.0187	0.0119	0.9503
	Transformer	0.0223	0.0306	0.0190	0.8320
	LSTM	0.0439	0.0604	0.0387	0.7468
B5 (80%)	CNN-Transformer	0.0018	0.0025	0.0027	0.9902
	CNN-LSTM	0.0104	0.0147	0.0116	0.9345
	Transformer	0.0146	0.0204	0.0167	0.8793
	LSTM	0.0305	0.0425	0.0323	0.7725

Table 2. Comparison of prediction errors among CNN-Transformer and other three methods based on B5.



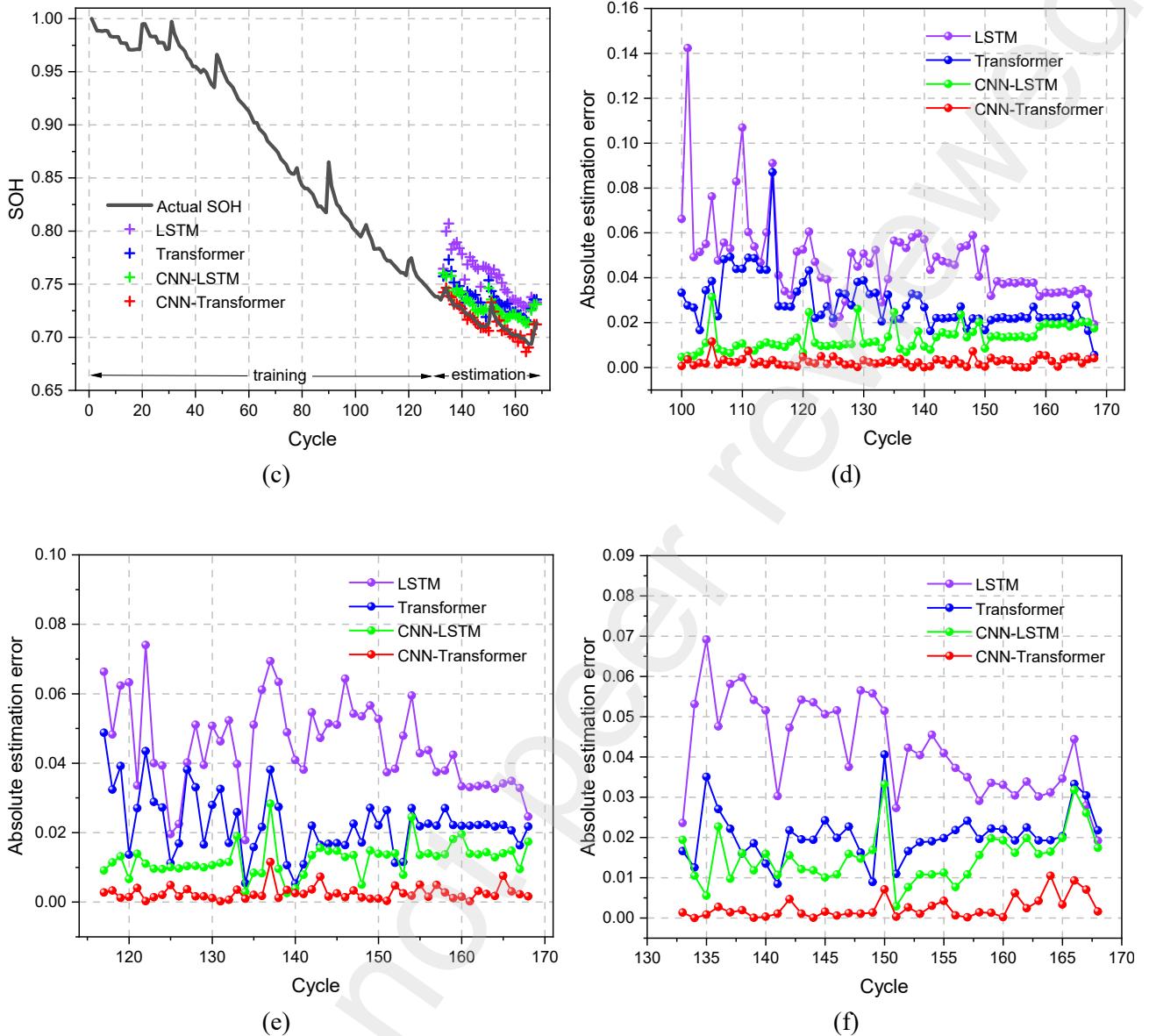


Fig. 5. The SOH estimation regarding B5 by 60% (a), 70% (b), 80% (c), and estimation error by 60% (d), 70% (e), 80% (f) training data.

The loss function is an indicator for evaluating algorithm accuracy in modeling the dataset by computing the distance between the current outcome and the actual value. The smaller output value of the loss function is, the more accurate algorithm for dataset modelling will achieve. A variety of loss functions exist in machine learning, e.g., regression loss function, multi-class classification loss function, and MAE loss. In this work, MAE is used as the loss function. The training loss can be monitored in each epoch. Fig. 6(a)-(c) illustrates the B5 evaluation of training and testing loss with epochs based on 60%, 70%, and 80% training data, respectively. In each case, the training loss by the CNN-Transformer is lowest among all the methods, indicating the most accurate algorithm for modeling the dataset.

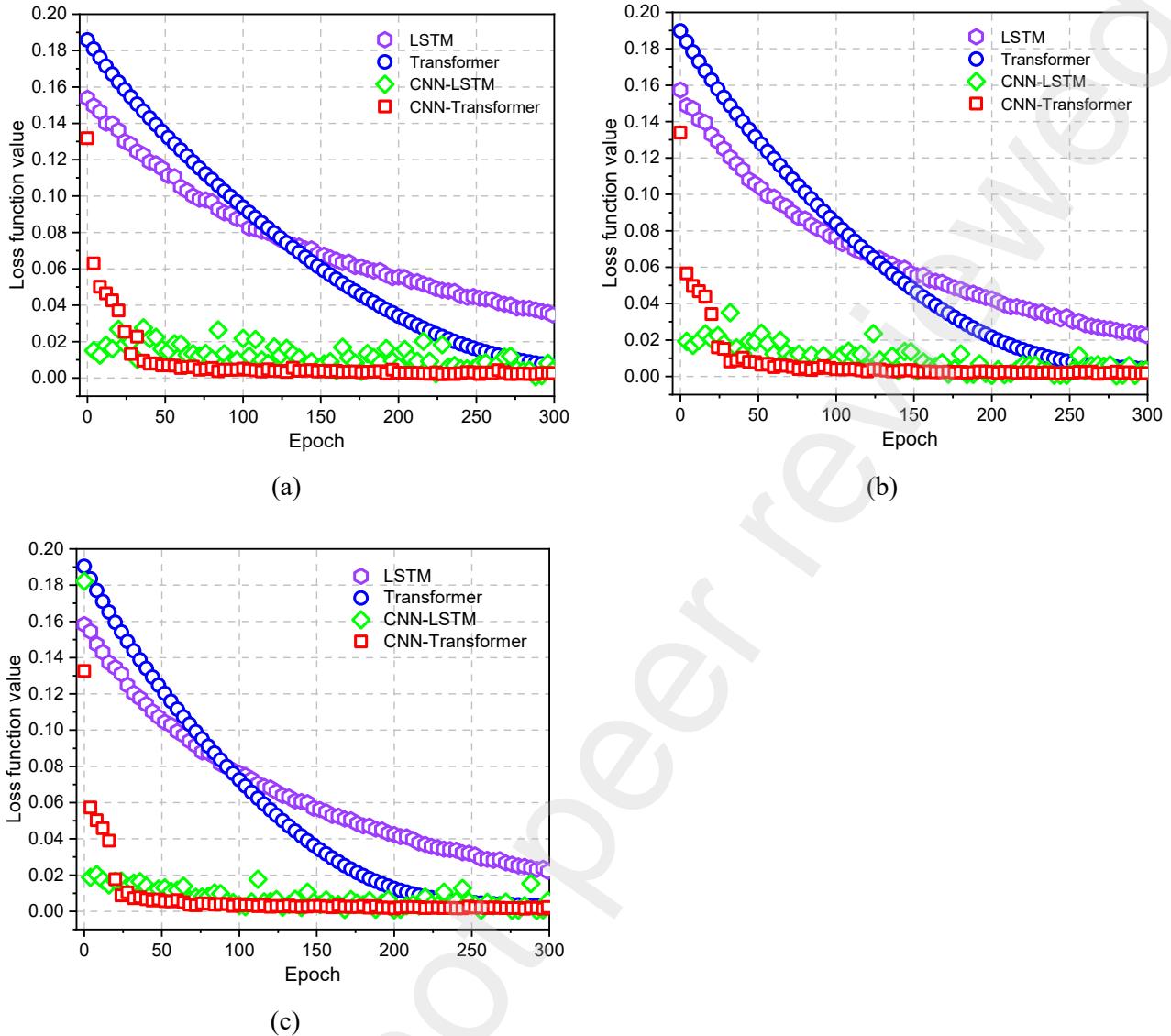


Fig. 6. Loss function values of independent training process for B5 by 60% (a), 70% (b), 80% (c) training dataset.

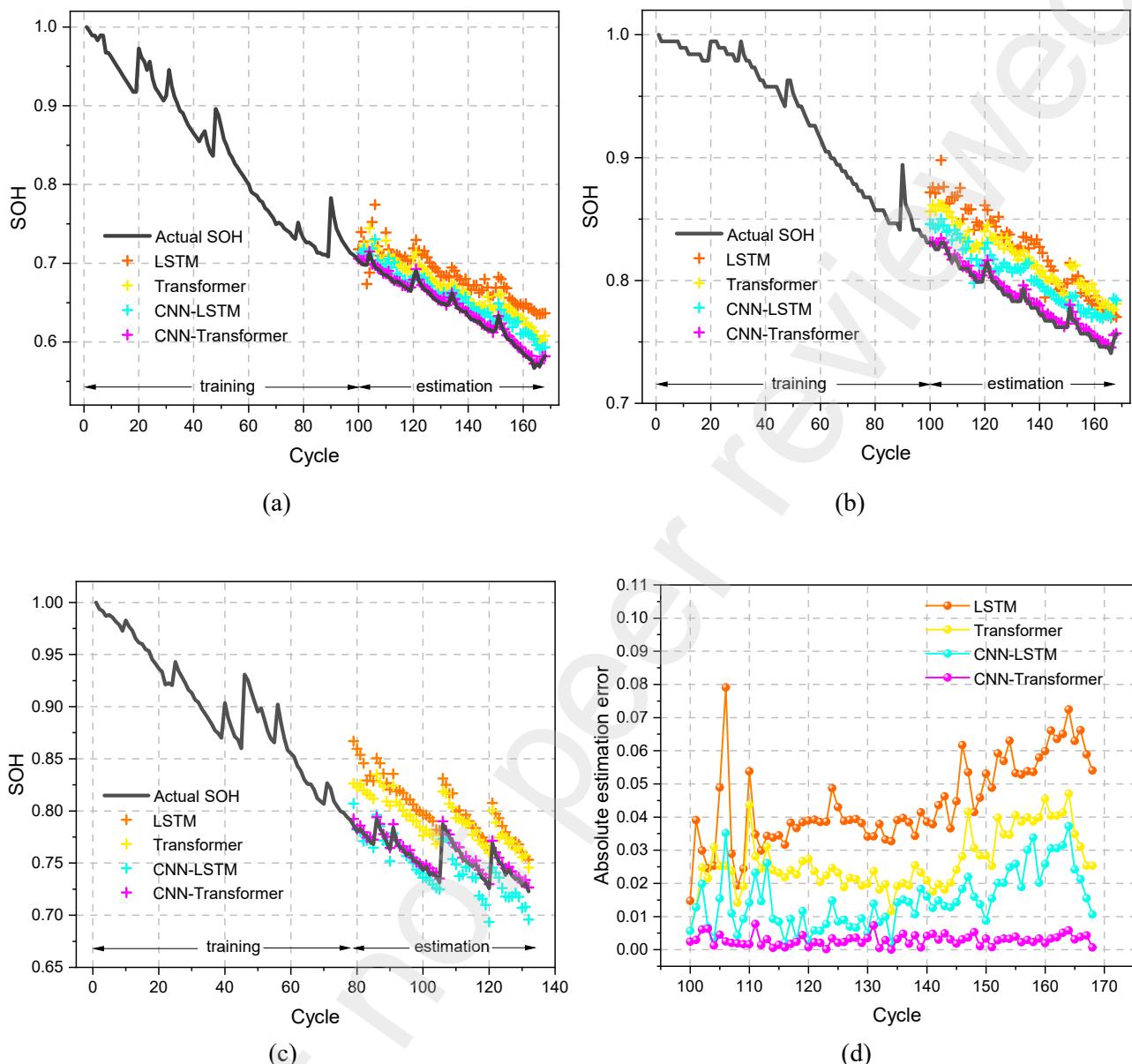
As shown in Table 3, the MAEs, MAPEs, and RMSEs of the CNN-Transformer model are within 0.55% and R^2 is over 99.5%, showing that CNN-Transformer model has better performance on SOH estimation. Moreover, significant improvement is observed regarding the SOH estimation error of lithium-ion batteries by using the CNN-Transformer method.

Fig. 7(a)-(c) illustrates the prediction results of the four methods for B6, B7, and B18 based on 60% training data, respectively. Fig. 7(d)-(f) displays the estimation error of these three cases. It should be noted that the predictions are made for each test cell individually. The absolute estimation errors of CNN-Transformer are less than 1%, while for LSTM, Transformer, and CNN-LSTM, the errors are within 8%, 5%, and 4%, respectively.

Battery	Method	MAE	MAPE	RMSE	R^2
B6 (60%)	CNN-Transformer	0.0030	0.0054	0.0032	0.9989
	CNN-LSTM	0.0192	0.0263	0.0235	0.9233
	Transformer	0.0409	0.0568	0.0514	0.8593
	LSTM	0.0574	0.0746	0.0681	0.7302

B6 (70%)	CNN-Transformer	0.0028	0.0047	0.0036	0.9991
	CNN-LSTM	0.0158	0.0294	0.0190	0.9390
	Transformer	0.0375	0.0565	0.0433	0.8634
	LSTM	0.0474	0.0646	0.0503	0.7524
B6 (80%)	CNN-Transformer	0.0025	0.0031	0.0033	0.9995
	CNN-LSTM	0.0092	0.0164	0.0107	0.9633
	Transformer	0.0359	0.0568	0.0394	0.8793
	LSTM	0.0404	0.0689	0.0481	0.7602
B7 (60%)	CNN-Transformer	0.0032	0.0041	0.0037	0.9979
	CNN-LSTM	0.0298	0.0360	0.0325	0.9224
	Transformer	0.0436	0.0511	0.0469	0.8165
	LSTM	0.0559	0.0704	0.0615	0.7529
B7 (70%)	CNN-Transformer	0.0026	0.0040	0.0029	0.9973
	CNN-LSTM	0.0267	0.0165	0.0115	0.9374
	Transformer	0.0452	0.0590	0.0533	0.7929
	LSTM	0.0677	0.0893	0.0806	0.7373
B7 (80%)	CNN-Transformer	0.0029	0.0042	0.0043	0.9958
	CNN-LSTM	0.0162	0.0332	0.0271	0.9451
	Transformer	0.0336	0.0511	0.0443	0.8328
	LSTM	0.0419	0.0704	0.0565	0.7805
B18 (60%)	CNN-Transformer	0.0029	0.0038	0.0032	0.9977
	CNN-LSTM	0.0118	0.0162	0.0144	0.9460
	Transformer	0.0359	0.0453	0.0361	0.8195
	LSTM	0.0481	0.0594	0.0492	0.7055
B18 (70%)	CNN-Transformer	0.0032	0.0049	0.0037	0.9985
	CNN-LSTM	0.0234	0.0389	0.0285	0.9498
	Transformer	0.0487	0.0635	0.0540	0.9528
	LSTM	0.0445	0.0627	0.0526	0.7957
B18 (80%)	CNN-Transformer	0.0024	0.0037	0.0027	0.9982
	CNN-LSTM	0.0204	0.0391	0.0278	0.9320
	Transformer	0.0569	0.0765	0.0653	0.8494
	LSTM	0.0425	0.0700	0.0494	0.7876

Table 3. Comparison of estimation errors among CNN-Transformer and other methods based on B6, B7 and B18.



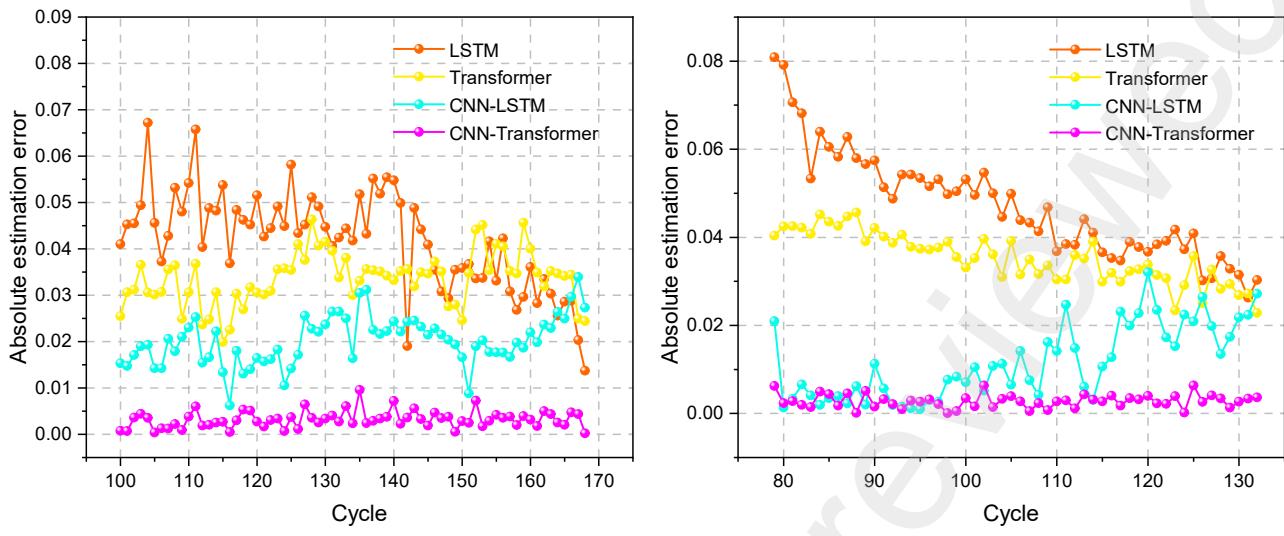
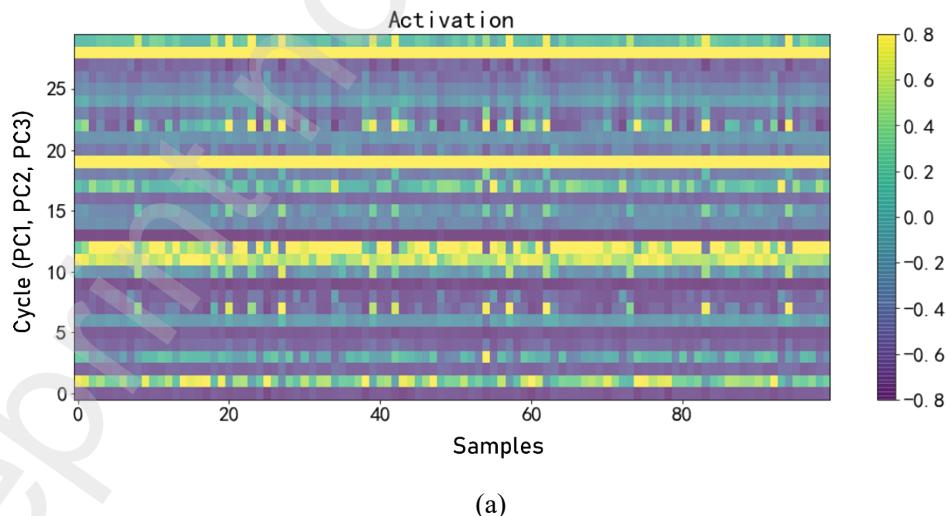


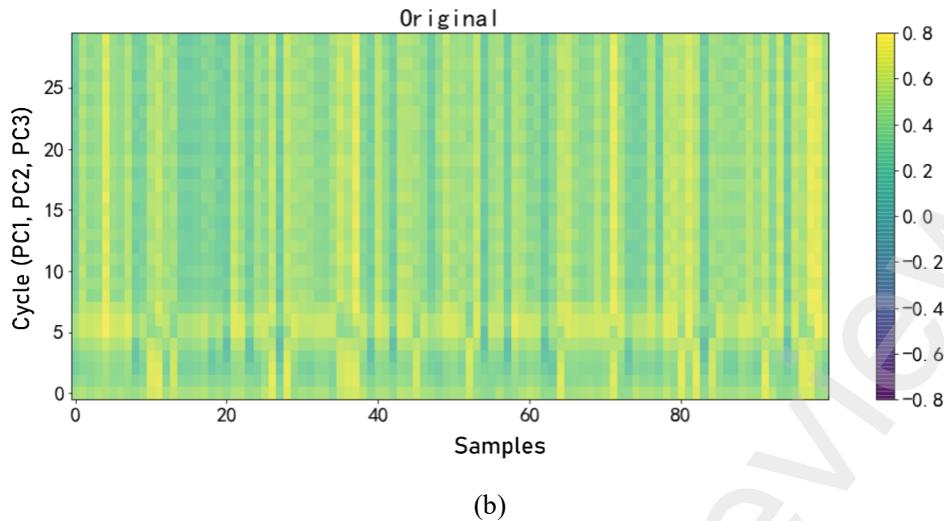
Fig. 7. The results of SOH estimation for the other batteries based on 60% training dataset, including B6 (a), B7 (b), and B18 (c). The errors of SOH estimation for B6 (d), B7 (e), and B18 (f).

3.3 Comprehensive performance analysis of CNN-Transformer model

An activation map of the well-trained models is shown in Fig. 8, by taking 30 cycles of battery data as an example. Two original data matrices are shaped with the length of 100 samples and the width of 30 cycles (with each cycle containing three features, which are PC1, PC2, PC3). For evaluating the model feature effects on prediction, interest areas with different levels are superimposed on the original data. As increased battery data fed to the CNN-Transformer neural network (from the 1st to the 30th cycle) lead to augmented activation levels, the features captured from CNN-Transformer provide a more significant impact on prediction.



(a)



(b)

Fig. 8. Activation maps of features. (a) After CNN-Transformer neural network. (b) Before CNN-Transformer neural network.

There are few evaluation indicators include FLOPs, training time, parameters, and storage size. FLOPs, the total number of floating-point operations required for a single forward pass, are used to evaluate the computation complexity of neural networks. The training time represents the model training efficiency, which is counted by the `time.process_time` function of Python. The parameters, an indirect indicator affecting the model's storage size, are counted by the `model.summary` function built-in to Keras. The storage size is calculated by the `os.path.getsize` function of Python. From Table 4, we can see that, CNN-Transformer model has no obvious advantage over the other three models except for the training time.

Methods	Performance index			
	FLOPs (Million)	Training time (h)	Parameters (Million)	Storage size (KB)
LSTM	0.695	0.263	0.0048	754
Transformer	0.845	0.203	0.0108	1258
CNN-LSTM	0.648	0.297	0.0078	958
CNN-Transformer	1.101	0.224	0.0261	2235

Table 4. The comprehensive evaluation of the proposed method.

4. Conclusion

A novel method, CNN-Transformer, is proposed to achieve accurate estimation of battery SOH. With CNN-Transformer, we leverage the convolution mechanisms to extract local information and the self-attention operators to capture global representations. In particular, to supervise both the CNN and Transformer branches to couple the CNN-style and Transformer-style variables, cross-entropy losses are used, and all the data are specialized into the same dimensionality. The results show that it can better capture local and global information than some traditional sequential prediction methods.

In this study, LSTM, Transformer, CNN-LSTM, and CNN-Transformer models are employed and compared in predicting the battery SOH. The original dataset is first built up as the input for neural networks, and then the four methods are used to train the models. Finally, the output from the neural networks is used for predicting the SOH estimation. Four batteries dataset, B5, B6, B7, and B18 from NASA are considered for the tests. For each battery, three proportions of training data are considered: 60%, 70%, and 80%. In the training effects, the CNN-Transformer

model can produce lower loss values and outperforms LSTM, Transformer, and CNN-LSTM models in different cases. In the testing process, the CNN-Transformer model achieved more competitive performance than the other three models, as evidenced by lower absolute estimation errors, MAE, MAPE, and RMSE, and higher R^2 . Moreover, compared with other methods, this method is able to train the model with less training time and has the foreseeable potential for further study and development.

References

- [1] Zhou D, Zheng W, Chen S, Fu P, Zhu H, Song B, et al. Research on state of health prediction model for lithium batteries based on actual diverse data. *Energy* 2021;230:120851. <https://doi.org/10.1016/j.energy.2021.120851>.
- [2] See KW, Wang G, Zhang Y, Wang Y, Meng L, Gu X, et al. Critical review and functional safety of a battery management system for large-scale lithium-ion battery pack technologies. *Int J Coal Sci Technol* 2022;9. <https://doi.org/10.1007/s40789-022-00494-0>.
- [3] Li X, Wang Z, Zhang L, Zou C, Dorrell DD. State-of-health estimation for Li-ion batteries by combining the incremental capacity analysis method with grey relational analysis. *J Power Sources* 2019;410–411:106–14. <https://doi.org/10.1016/j.jpowsour.2018.10.069>.
- [4] Zhang Y, Liu Y, Wang J, Zhang T. State-of-health estimation for lithium-ion batteries by combining model-based incremental capacity analysis with support vector regression. *Energy* 2022;239:121986. <https://doi.org/10.1016/j.energy.2021.121986>.
- [5] Ospina Agudelo B, Zamboni W, Monmasson E. Application domain extension of incremental capacity-based battery SoH indicators. *Energy* 2021;234:121224. <https://doi.org/10.1016/j.energy.2021.121224>.
- [6] Hu X, Jiang J, Cao D, Egardt B. Battery health prognosis for electric vehicles using sample entropy and sparse Bayesian predictive modeling. *IEEE Trans Ind Electron* 2016;63:2645–56. <https://doi.org/10.1109/TIE.2015.2461523>.
- [7] Zheng Y, Wang J, Qin C, Lu L, Han X, Ouyang M. A novel capacity estimation method based on charging curve sections for lithium-ion batteries in electric vehicles. *Energy* 2019;185:361–71. <https://doi.org/10.1016/j.energy.2019.07.059>.
- [8] Di Domenico D, Stefanopoulou A, Fiengo G. Lithium-Ion Battery State of Charge and Critical Surface Charge Estimation Using an Electrochemical Model-Based Extended Kalman Filter. *J Dyn Syst Meas Control* 2010;132. <https://doi.org/10.1115/1.4002475>.
- [9] Shu X, Li G, Shen J, Lei Z, Chen Z, Liu Y. An adaptive multi-state estimation algorithm for lithium-ion batteries incorporating temperature compensation. *Energy* 2020;207:118262. <https://doi.org/10.1016/j.energy.2020.118262>.
- [10] Lim KC, Bastawrous HA, Duong VH, See KW, Zhang P, Dou SX. Fading Kalman filter-based real-time state of charge estimation in LiFePO₄ battery-powered electric vehicles. *Appl Energy* 2016;169:40–8. <https://doi.org/10.1016/j.apenergy.2016.01.096>.
- [11] Jokar A, Rajabloo B, Désilets M, Lacroix M. Review of simplified Pseudo-two-Dimensional models of lithium-ion batteries. *J Power Sources* 2016;327:44–55. <https://doi.org/10.1016/j.jpowsour.2016.07.036>.
- [12] Deng Z, Yang L, Deng H, Cai Y, Li D. Polynomial approximation pseudo-two-dimensional battery model for online application in embedded battery management system. *Energy* 2018;142:838–50. <https://doi.org/10.1016/j.energy.2017.10.097>.
- [13] Han S, Tang Y, Khaleghi Rahimian S. A numerically efficient method of solving the full-order pseudo-2-dimensional (P2D) Li-ion cell model. *J Power Sources* 2021;490:229571. <https://doi.org/10.1016/j.jpowsour.2021.229571>.
- [14] Lin X, Hao X, Liu Z, Jia W. Health conscious fast charging of Li-ion batteries via a single particle model with

- aging mechanisms. *J Power Sources* 2018;400:305–16. <https://doi.org/10.1016/j.jpowsour.2018.08.030>.
- [15] Li J, Adewuyi K, Lotfi N, Landers RG, Park J. A single particle model with chemical/mechanical degradation physics for lithium ion battery State of Health (SOH) estimation. *Appl Energy* 2018;212:1178–90. <https://doi.org/10.1016/j.apenergy.2018.01.011>.
- [16] Qu J, Liu F, Ma Y, Fan J. A Neural-Network-Based Method for RUL Prediction and SOH Monitoring of Lithium-Ion Battery. *IEEE Access* 2019;7:87178–91. <https://doi.org/10.1109/ACCESS.2019.2925468>.
- [17] Feng X, Weng C, He X, Han X, Lu L, Ren D, et al. Online State-of-Health Estimation for Li-Ion Battery Using Partial Charging Segment Based on Support Vector Machine. *IEEE Trans Veh Technol* 2019;68:8583–92. <https://doi.org/10.1109/TVT.2019.2927120>.
- [18] Dong G, Han W, Wang Y. Dynamic Bayesian Network-Based Lithium-Ion Battery Health Prognosis for Electric Vehicles. *IEEE Trans Ind Electron* 2021;68:10949–58. <https://doi.org/10.1109/TIE.2020.3034855>.
- [19] Li Y, Zou C, Berecibar M, Namini-Maury E, Chan JCW, van den Bossche P, et al. Random forest regression for online capacity estimation of lithium-ion batteries. *Appl Energy* 2018;232:197–210. <https://doi.org/10.1016/j.apenergy.2018.09.182>.
- [20] Meng J, Cai L, Luo G, Stroe DI, Teodorescu R. Lithium-ion battery state of health estimation with short-term current pulse test and support vector machine. *Microelectron Reliab* 2018;88–90:1216–20. <https://doi.org/10.1016/j.microrel.2018.07.025>.
- [21] Lin M, Yan C, Meng J, Wang W, Wu J. Lithium-ion batteries health prognosis via differential thermal capacity with simulated annealing and support vector regression. *Energy* 2022;250:123829. <https://doi.org/10.1016/j.energy.2022.123829>.
- [22] Urolagin S, Sharma N, Datta TK. A combined architecture of multivariate LSTM with Mahalanobis and Z-Score transformations for oil price forecasting. *Energy* 2021;231:120963. <https://doi.org/10.1016/j.energy.2021.120963>.
- [23] Guo R, Shen W. A data-model fusion method for online state of power estimation of lithium-ion batteries at high discharge rate in electric vehicles. *Energy* 2022;254:124270. <https://doi.org/10.1016/j.energy.2022.124270>.
- [24] You GW, Park S, Oh D. Diagnosis of Electric Vehicle Batteries Using Recurrent Neural Networks. *IEEE Trans Ind Electron* 2017;64:4885–93. <https://doi.org/10.1109/TIE.2017.2674593>.
- [25] Cheng G, Wang X, He Y. Remaining useful life and state of health prediction for lithium batteries based on empirical mode decomposition and a long and short memory neural network. *Energy* 2021;232:121022. <https://doi.org/10.1016/j.energy.2021.121022>.
- [26] Ren X, Liu S, Yu X, Dong X. A method for state-of-charge estimation of lithium-ion batteries based on PSO-LSTM. *Energy* 2021;234:121236. <https://doi.org/10.1016/j.energy.2021.121236>.
- [27] Ma Y, Shan C, Gao J, Chen H. A novel method for state of health estimation of lithium-ion batteries based on improved LSTM and health indicators extraction. *Energy* 2022;251:123973. <https://doi.org/10.1016/j.energy.2022.123973>.
- [28] Li P, Zhang Z, Xiong Q, Ding B, Hou J, Luo D, et al. State-of-health estimation and remaining useful life prediction for the lithium-ion battery based on a variant long short term memory neural network. *J Power Sources* 2020;459:228069. <https://doi.org/10.1016/j.jpowsour.2020.228069>.
- [29] Li P, Zhang Z, Grosu R, Deng Z, Hou J, Rong Y, et al. An end-to-end neural network framework for state-of-health estimation and remaining useful life prediction of electric vehicle lithium batteries. *Renew Sustain Energy Rev* 2022;156:111843. <https://doi.org/10.1016/j.rser.2021.111843>.
- [30] Wang F-K, Amogne ZE, Chou J-H, Tseng C. Online remaining useful life prediction of lithium-ion batteries using bidirectional long short-term memory with attention mechanism. *Energy* 2022;254:124344. <https://doi.org/10.1016/j.energy.2022.124344>.

- [31] Yang K, Tang Y, Zhang S, Zhang Z. A deep learning approach to state of charge estimation of lithium-ion batteries based on dual-stage attention mechanism. *Energy* 2022;244:123233. <https://doi.org/10.1016/j.energy.2022.123233>.
- [32] Zhou K, Wang W, Hu T, Deng K. Time series forecasting and classification models based on recurrent with attention mechanism and generative adversarial networks. *Sensors (Switzerland)* 2020;20:1–20. <https://doi.org/10.3390/s20247211>.
- [33] Zhou H, Zhang Y, Yang L, Liu Q, Yan K, Du Y. Short-Term photovoltaic power forecasting based on long short term memory neural network and attention mechanism. *IEEE Access* 2019;7:78063–74. <https://doi.org/10.1109/ACCESS.2019.2923006>.
- [34] Hu J, Zheng W. Multistage attention network for multivariate time series prediction. *Neurocomputing* 2020;383:122–37. <https://doi.org/10.1016/j.neucom.2019.11.060>.
- [35] Xie Z, Gu X, Shen Y. A Machine Learning Study of Predicting Mixing and Segregation Behaviors in a Bidisperse Solid-Liquid Fluidized Bed. *Ind Eng Chem Res* 2022;61:8551–65. <https://doi.org/10.1021/acs.iecr.2c00071>.
- [36] Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. *Pattern Recognit* 2018;77:354–77. <https://doi.org/10.1016/j.patcog.2017.10.013>.
- [37] Kim TY, Cho SB. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy* 2019;182:72–81. <https://doi.org/10.1016/j.energy.2019.05.230>.
- [38] Borovykh A, Bohte S, Oosterlee CW. Conditional time series forecasting with convolutional neural networks. *J Syst Eng Electron* 2017;28:162–9. <https://doi.org/10.1016/j.jse.2017.09.001>.
- [39] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention Is All You Need. *Adv Neural Inf Process Syst* 2017;5:5999–6009. <https://doi.org/10.48550/arXiv.1706.03762>.
- [40] Peng Z, Huang W, Gu S, Xie L, Wang Y, Jiao J, et al. Conformer: Local Features Coupling Global Representations for Visual Recognition. 2021 IEEE/CVF Int. Conf. Comput. Vis., IEEE; 2021, p. 357–66. <https://doi.org/10.1109/ICCV48922.2021.00042>.
- [41] Sen R, Yu H-F, Dhillon I. Think Globally, Act Locally: A Deep Neural Network Approach to High-Dimensional Time Series Forecasting. *Adv Neural Inf Process Syst* 2019;32:1–10. <https://doi.org/10.48550/arXiv.1905.03806>.
- [42] Kong J zhen, Yang F, Zhang X, Pan E, Peng Z, Wang D. Voltage-temperature health feature extraction to improve prognostics and health management of lithium-ion batteries. *Energy* 2021;223:120114. <https://doi.org/10.1016/j.energy.2021.120114>.
- [43] Jebli I, Belouadha FZ, Kabbaj MI, Tilioua A. Prediction of solar energy guided by pearson correlation using machine learning. *Energy* 2021;224:120109. <https://doi.org/10.1016/j.energy.2021.120109>.
- [44] Schober P, Schwarte LA. Correlation coefficients: Appropriate use and interpretation. *Anesth Analg* 2018;126:1763–8. <https://doi.org/10.1213/ANE.0000000000002864>.
- [45] Berk I, Ediger V. A historical assessment of Turkey's natural gas import vulnerability. *Energy* 2018;145:540–7. <https://doi.org/10.1016/j.energy.2018.01.022>.
- [46] Wang L, Yao Y, Wang K, Adenutsi CD, Zhao G, Lai F. Hybrid application of unsupervised and supervised learning in forecasting absolute open flow potential for shale gas reservoirs. *Energy* 2022;243:122747. <https://doi.org/10.1016/j.energy.2021.122747>.
- [47] Bao LN, Le DN, Nguyen GN, Bhateja V, Satapathy SC. Optimizing feature selection in video-based recognition using Max-Min Ant System for the online video contextual advertisement user-oriented system. *J Comput Sci* 2017;21:361–70. <https://doi.org/10.1016/j.jocs.2016.10.016>.

- [48] Zou Y, Hu X, Ma H, Li SE. Combined State of Charge and State of Health estimation over lithium-ion battery cell cycle lifespan for electric vehicles. *J Power Sources* 2015;273:793–803. <https://doi.org/10.1016/j.jpowsour.2014.09.146>.
- [49] Guney K, Yildiz C, Kaya S, Turkmen M. Artificial neural networks for calculating the characteristic impedance of air-suspended trapezoidal and rectangular-shaped microshield lines. *J Electromagn Waves Appl* 2006;20:1161–74. <https://doi.org/10.1163/156939306777442917>.
- [50] Kingma DP, Ba JL. Adam: A method for stochastic optimization. 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., 2015, p. 1–15. <https://doi.org/10.48550/arXiv.1412.6980>.
- [51] Wang Z, Yan W, Oates T. Time series classification from scratch with deep neural networks: A strong baseline. 2017 Int. Jt. Conf. Neural Networks, IEEE; 2017, p. 1578–85. <https://doi.org/10.1109/IJCNN.2017.7966039>.
- [52] Wang S, Takyi-Aninakwa P, Jin S, Yu C, Fernandez C, Stroe DI. An improved feedforward-long short-term memory modeling method for the whole-life-cycle state of charge prediction of lithium-ion batteries considering current-voltage-temperature variation. *Energy* 2022;254:124224. <https://doi.org/10.1016/j.energy.2022.124224>.
- [53] Chicco D, Warrens MJ, Jurman G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput Sci* 2021;7:1–24. <https://doi.org/10.7717/PEERJ-CS.623>.