Journal of Energy Storage 88 (2024) 111626

Contents lists available at ScienceDirect

Journal of Energy Storage

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

Research papers

Remaining useful life prediction for lithium-ion batteries incorporating spatio-temporal information

Zihao Lv [a], Yi Song [b], Chunlin He [c], Liming Xu [c,*]

[a] School of Electronic Information Engineering, China West Normal University, Nanchong City 637009, China
[b] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu City 611756, China
[c] Artificial Intelligence Research Institute, School of Computer Science, China West Normal University, Nanchong City 637009, China

ARTICLE INFO

Keywords:
Deep learning
Lithium-ion battery
Remaining useful life prediction
Grid search algorithm
Feature engineering

ABSTRACT

Accurately predicting the remaining useful life (RUL) of lithium-ion batteries (LIBs) is crucial in the field of industrial intelligence. However, existing generalized models face challenges in battery life prediction due to the presence of prevalent noise and limited degradation data. To address this issue, we proposed a spatiotemporally integrated RUL prediction model, namely, the DAE-MSCNN-LSTM model, which leverages the combination of multiscale convolutional neural network (MSCNN) and long-short-term memory network (LSTM). This approach can be used to effectively extract feature information from limited available data. Moreover, the model incorporates a denoising autoencoder (DAE) to refine the raw data by mitigating noise and handling outliers. Subsequently, the processed data are fed into parallel MSCNN and LSTM networks, allowing us to capture both spatial and temporal information. The fused features from these networks are subsequently input into a multilayer perceptron (MLP) for RUL prediction. Additionally, we introduced a unified learning framework to address data denoising and model prediction simultaneously. Finally, the optimal hyperparameters were determined using the grid search algorithm. Through extensive experiments on the NASA and CALCE datasets, the superiority of our proposed LIB RUL prediction for model is demonstrated.

1. Introduction

Rechargeable LIBs provide the advantages of high specific energy, low self-discharge rate, long life and environmental, friendliness and are widely used in portable electronic devices and other fields [1,2]. Monitoring capacity degradation and predicting the RUL are challenging tasks in the use of LIBs [3]. In practical applications, with increasing number of charge/discharge cycles, the performance decreases subsequently, especially as the battery capacity gradually decreases until the end of life (EOL) [4]. After EOL stage, the battery power and capacity decrease more rapidly, a phenomenon that seriously affects device reliability and can even create catastrophic problems [5,6]. Therefore, studying the safety of power battery systems and realizing accurate RUL prediction for LIBs are highly important for improving the safety of electronic devices [7]. Fig. 1 shows a diagram of RUL prediction, with the capacity as a health factor.

Therefore, to predict the RUL of LIBs, we must analyze battery operational data. Currently, the prediction methods for LIBs mainly include model-driven methods and data-driven methods [8]. Model-based approaches, such as electrochemical models [9] and equivalent circuit models [10], can observe the internal state variables of a cell through an iterative mechanism and explain the degradation mechanism between two adjacent cycles [11]. At the present time, popular model-based approaches aim to fit the degradation trajectory of LIBs through a mathematical model [12]. Tian et al. [13] proposed a fractional-order model based on electrochemical impedance spectroscopy to solve the problems whereby the aging mechanism of LIBs is unclear and sensitive to usage conditions. Song et al. [14] proposed a fully coupled electro-chemo-mechanical theoretical model, that accounts for several mechanical and electrochemical factors. Wang et al. [15] used electronic components with fixed-topology circuits to simulate the characteristics of external voltages and currents to effectively achieve the desired accuracy. Although model-based methods have been widely used in predicting the RUL of LIBs, they still suffer two significant drawbacks. (1) Accurately analyzing complex models of dynamic systems is usually very difficult, especially when the systems operate in highly fluctuating environments [16]. (2) Due to the complexity of the reaction mechanisms within LIBs, there are no available model-based methods that can accurately predict the aging behavior of LIBs under various operating conditions.

[*] Corresponding author.
E-mail addresses: zihao_lv@163.com (Z. Lv), songyi.swjtu.edu.cn@my.swjtu.edu.cn (Y. Song), chunlin_he@163.com (C. He), xulm@cwnu.edu.cn (L. Xu).

https://doi.org/10.1016/j.est.2024.111626
Received 3 November 2023; Received in revised form 23 January 2024; Accepted 4 April 2024
Available online 9 April 2024

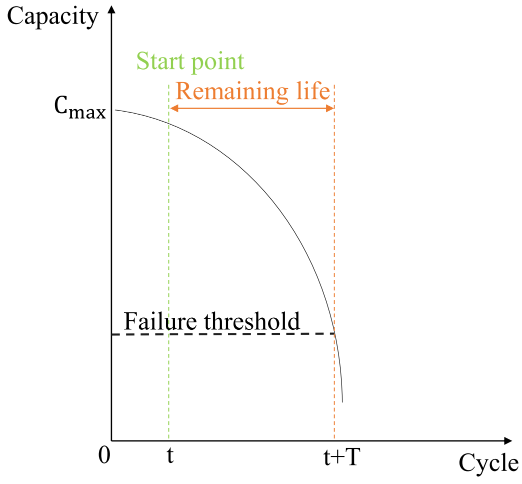


Fig. 1. LIB degradation process and RUL prediction.

The data-driven approach based on comparing a battery to a black box, rather than an actual mathematical model, entails the use of intelligent algorithmic models (e.g., neural networks (NNs) [17], support vector machines (SVMs) [18], and Bayesian regression [19]) to analyze the relationship between lithium battery life characteristic parameters and battery state of health (SOH) by learning from a large amount of historical data [20]. Cheng et al. [21] designed a prediction model incorporating the empirical mode decomposition approach and LSTM network to accurately predict the SOH and RUL of batteries. Li et al. [22] used support vector regression to model battery, degradation and subsequently estimated the battery capacity by phased incremental curves. Lyu et al. [23] proposed an indirect hybrid model for online battery diagnosis and health management for accurate SOH estimation and RUL prediction. Zhou et al. [24] established a transfer learning strategy combined with cycle life prediction technology to effectively solve the long-term aging trajectory prediction problem of LIBs. Zraibi et al. [25] proposed a hybrid method, named the CNN-LSTM-DNN, for the estimation of the battery's RUL and improving prediction accuracy with acceptable execution time. Although the data-driven approach overcomes the shortcomings of the model-driven approach, it still has limitations. (1) Data-driven methods exhibit high data quantity and quality requirements, and the training and prediction ability of models is affected if the data quantity is too small or if there is excessive noise. (2) Most existing models do not consider multilayer feature information of the data, which largely limits their prediction accuracy.

In recent years, data-driven methods based on deep learning have shown good performance in the field of RUL prediction, but they still suffer from the following three main problems. (1) Noise. The nonlinear capacity curve of a battery is inherently noisy and contains outliers. Training neural network models directly on raw data could lead to issues such as over- or underfitting. (2) Insufficient degradation data. Since battery data collection is a tedious and expensive process, the available battery data are often very limited. Therefore, when designing a network model, it is necessary to enable the model to learn as much feature information as possible. (3) Shortcomings of the single-path architecture. The existing prediction methods mostly involve single-path network architectures, which suffer from limited anti-interference ability and low robustness.

Considering the limitations of previous studies, in this paper, a data-driven RUL prediction model, namely, DAE-MSCNN-LSTM, for LIBs was proposed based on an MSCNN and LSTM. The main contributions of this paper are described below.

- We proposed an end-to-end deep RUL prediction method based on a dual-channel architecture. Experiments on two widely used LIB datasets, i.e., NASA and CALCE, showed that our method significantly outperforms current state-of-the-art RUL prediction methods. To our knowledge, we are one of the first teams to introduce a multichannel deep learning approach in the field of RUL prediction. This method can fully utilize degradation information to predict the battery RUL.

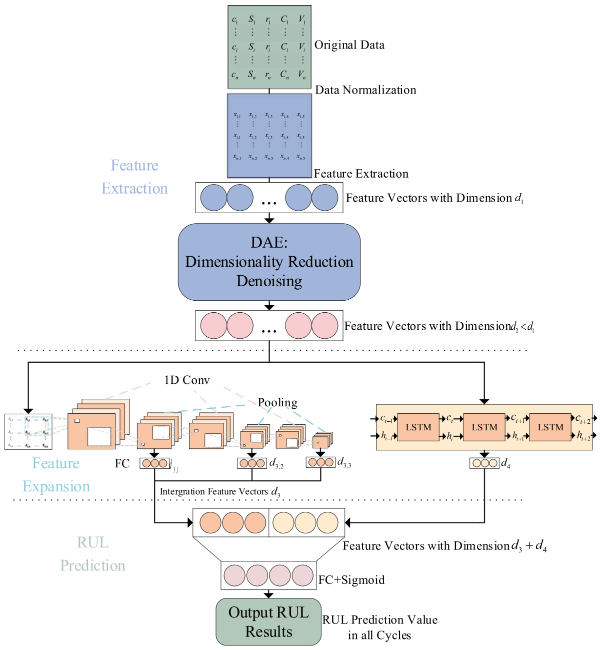

c_1 S_1 r_1 C_1 V_1
c_i S_i r_i C_i V_i
c_n S_n r_n C_n V_n
Original Data
Data Normalization
$x_{1,1}$ $x_{1,2}$ $x_{1,3}$ $x_{1,4}$ $x_{1,5}$
$x_{i,1}$ $x_{i,2}$ $x_{i,3}$ $x_{i,4}$ $x_{i,5}$
$x_{n,1}$ $x_{n,2}$ $x_{n,3}$ $x_{n,4}$ $x_{n,5}$
Feature Extraction
Feature Extraction
Feature Vectors with Dimension d_1
DAE: Dimensionality Reduction Denoising
Feature Vectors with Dimension$d_2 < d_1$
1D Conv
Pooling
Feature Expansion
FC
$d_{3,1}$
$d_{3,2}$
$d_{3,3}$
Intergration Feature Vectors d_3
c_{t-1} h_{t-1} LSTM c_t h_t LSTM c_{t+1} h_{t+1} LSTM c_{t+2} h_{t+2}
d_4
RUL Prediction
Feature Vectors with Dimension $d_3 + d_4$
FC+Sigmoid
Output RUL Results
RUL Prediction Value in all Cycles

Fig. 2. DAE-MSCNN-LSTM network for RUL prediction.

- Based on the DAE, our model can remove noise and redundant values without domain knowledge. Our novel dual-channel design better represents the interrelationships among features and efficiently captures the temporal and spatial information of battery degradation sequences. With limited degradation data, the model prediction performance is improved, and the model is very robust.
- RUL prediction via our method is accurate and efficient for helping maintenance personnel develop better maintenance strategies and exhibits great potential for future applications.

The remainder of this paper is organized as follows: in Section 2, the methodology used for RUL prediction of LIBs is described. In Section 3, the experiments are presented. In Section 4, the experimental results are examined. Finally, the paper is systematically summarized in Section 5.

2. Methodology

Since the MSCNN aims to fuse shallow and deep features to fully exploit spatial feature information, and the LSTM can extract temporal information from time series. To this end, we proposed dual-path feature aggregation networks that fuse spatiotemporal information (Fig. 2). Considering the difficulty in effectively utilizing the advantages of two models by simply concatenating them, inspired by the literature [26], we adopted the idea of feature fusion to concatenate the models and improve the performance of the fused model. In addition, the predictive ability of a model depends largely on the data quality. Therefore, we utilized the DAE to learn a compressed representation of the data and generated reconstructed output through a decoder for noise reduction and feature extraction. Moreover, to improve the generalization ability of the established model, we proposed a unified learning framework that combines the two tasks of data denoising and model prediction.

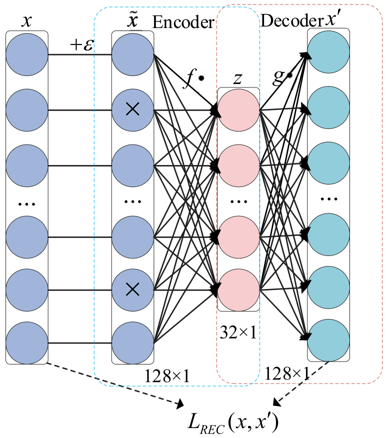


Fig. 3. Structure of the DAE.

The model aims to predict the RUL of LIBs using deep learning techniques. First, the feature vectors of the raw data are subjected to dimensionality reduction and denoising via a deep DAE. The downscaled feature vectors are transformed into feature maps, and the fusion of shallow and deep features is performed by MSCNN in order to fully exploit the spatial feature information of the sequences. Meanwhile, an LSTM-based method is used to extract global features from the time series. The output features produced by the two branches of the MSCNN and LSTM are fused and fed into the MLP layer to achieve RUL prediction. Our proposed model could theoretically improve accurate prediction of the remaining battery life and contribute innovative research ideas in the field of battery health management.

2.1. Denoising autoencoder of the DAE-MSCNN-LSTM model

A denoising autoencoder is a type of autoencoder and an unsupervised neural network structure that maps inputs to outputs [27], its network structure is shown in Fig. 3. It consists of an input layer, a hidden layer and an output layer. Compared to the traditional self-encoder, the DAE introduces noise into the input data, forcing the model to learn the underlying structure and features of the data. This approach improves the model ability to remove noise and extract features.

In the network training process, in this paper, Gaussian noise is added to a fixed dimension of the input vector to obtain input vector $\tilde{x}$ containing noise as follows:

$$\tilde{x} = x + \varepsilon. \tag{1}$$

where $x \in R^{d_1}$ and $\tilde{x} \in R^{d_1}$, d_1 signifies the dimensions of the input vector. $\varepsilon \in N\left(0, \sigma^2\right)$, σ represents the standard deviation of the Gaussian noise. Here, ε denotes the same Gaussian noise associated with the x dimension.

Subsequently, the vector $\tilde{x}$, which encompasses the introduced noise, is supplied to the DAE to encode z and decode x by the following equation.

$$z = f(\tilde{x}) = f(W \cdot \tilde{x} + b). \tag{2}$$

$$x' = g(z) = g\left(W' \cdot z + b'\right). \tag{3}$$

where $z \in R^r$ denotes the feature expression of the hidden layer, r denotes the number of neurons in the hidden layer. $x' \in R^{d_1}$ is the decoder output vector. $W \in R^{r \times d_1}$ and $W' \in R^{d_1 \times r}$ denote the weight matrix. $b \in R^r$ and $b' \in R^{d_1}$ are the bias vectors. f and g are the nonlinear activation functions.

In previous studies [28], model denoising and prediction were two separate tasks, and the relevance of these two tasks was ignored.

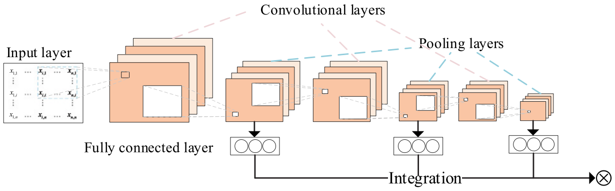


Fig. 4. Structure of the MSCNN.

Inspired by Ref. [29], in this paper, the two tasks of data denoising and model prediction are unified to establish a unified learning framework. Therefore, the reconstruction loss function is defined as follows:

$$\begin{aligned} loss &= L_{MSE} + L_{REC} \\ &= \frac{1}{N} \sum_{i=1}^{N} \left\| y_i - o_i \right\|^2 + \lambda \frac{1}{N} \sum_{i=1}^{N} \left\| x_i - g(f(\tilde{x})) \right\|^2 . \end{aligned} \tag{4}$$

where L_{MSE} is the mean square error loss function and L_{REC} is the reconstruction error loss function. N is the minimum batch size, y_i denotes the true label of the ith sample, and o_i denotes the predicted output of the model for the ith sample. The DAE-MSCNN-LSTM network updates the weights and biases according to Eq. (4) and performs backpropagation.

After training, in this paper, the encoder output generated of $z = f(\tilde{x})$ is used as the input for the next layer of network training, and the decoder output generated by the function $x' = g(z)$ is retained for reconstructing the error loss function, as shown in Fig. 3.

2.2. MultiScale convolutional neural networks

A general convolutional neural network (CNN) consists of a convolutional layer, a pooling layer and a fully connected layer [25]. The fully connected layer is usually connected on top of the output of the deepest convolutional (or pooling) layer, the discrete 2D convolutional model can be expressed as:

$$A(i,j) = \sum_{m=-s}^{s} \sum_{n=-s}^{s} K(m,n) \cdot z(i-m, j-n). \tag{5}$$

where $K(m,n)$ denotes the convolution kernel matrix and $z(i-m, j-n)$ denotes the input matrix processed by the encoder. The convolution kernel functions as a filter in convolution operation, and the result of convolution operation $A(i,j)$ can be obtained by multiplying and summing element by element according to the corresponding positions of the input matrix.

However, CNN tend to focus only on the local features of temporal sequences and ignore their global information, which is obviously unreasonable. To effectively capture the spatial dependencies of time series data and alleviate the problem of an insufficient amount of LIB degradation data, the MSCNN network is designed herein, which fuses shallow and deep convolutional features to fully utilize LIB features at different scales and achieve more accurate predictions.

As shown in Fig. 4, the MSCNN contains three convolutional layers and three pooling layers, and each pooling layer is connected to a fully connected layer to obtain LIB features at different scales. Finally, the output features of the three fully connected layers are subjected to feature interaction to obtain l_4, as expressed in Eq. (6).

$$\begin{aligned} l_4 &= \left(l_1 \times l_2\right) + \left(l_1 \times l_3\right) \\ &= \left[s\left(W_1 \cdot z_1 + b_1\right) \times s\left(W_2 \cdot z_2 + b_2\right)\right] \\ &+ \left[s\left(W_1 \cdot z_1 + b_1\right) \times s\left(W_3 \cdot z_3 + b_3\right)\right]. \end{aligned} \tag{6}$$

where z_i is the output z of the encoder, which comprises the features obtained after the ith pooling layer, W_i and b_i are the weight matrix and bias term corresponding to the fully connected layer, respectively, and l_i denotes the ith fully connected layer.

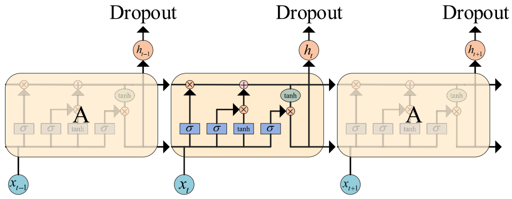


Fig. 5. LSTM structure in DAE-MSCNN-LSTM.

The MSCNN can learn the change rules and trends of LIBs during different charging and discharging cycles, effectively utilize the spatial properties between adjacent charging and discharging cycles extracted by the DAE, and learn convolutional features at different scales and combine them organically. With the layer-by-layer iteration process of the deep MSCNN and the original LIB feature maps, after a series of operations such as convolution and pooling, edge information and detailed features of the feature maps gradually disappear, instead highlighting the global information of each feature map and focusing more on the overall spatial distribution and structural features.

Finally, the extracted convolutional feature vectors, which are part of the combination with the LSTM, are used for final RUL prediction.

2.3. Long short-term memory

A recurrent neural network (RNN) is a neural network model specialized for processing sequence data and is commonly used for SOH estimation and RUL prediction of LIBs [30]. RNN contains self-feedback connections, which can account for the temporal order and dependency of the sequence data and help to process the sequence data step by step. However, since RNNs must transfer gradient information between different moments and balance the relationship between short-term and long-term memory, it is very difficult to train RNNs when the sequence length of the input samples is large, which can lead to problems such as gradient vanishing or explosion of the network. For this reason, an improved RNN model, LSTM, is used in this paper as an alternative to the traditional RNN (see Fig. 5).

The LSTM is a variant of the RNN model that effectively addresses the long-distance dependency problem [31]. The model comprises four main components: the forget gate, input gate, cell state, and output gate. The forget gate regulates whether the cell should retain or forget the memory state from the previous time step, enabling the LSTM to update the current input information while preserving the previous memory for more accurate predictions. The input gate calculates the relevant information from the current input and output of the previous time step to adapt to different time series patterns. Moreover, the output gate controls the strength of influence of the cell state on the next time step, allowing it to effectively transfer essential information. The computations for each gate function and state transfer process in the LSTM module can be expressed by Eqs. (7)–(11).

$$i^t = \sigma\left(W_i \cdot x^t + U_i \cdot h^{t-1} + b_i\right). \tag{7}$$

$$f^t = \sigma\left(W_f \cdot x^t + U_f \cdot h^{t-1} + b_f\right). \tag{8}$$

$$o^t = \sigma\left(W_o \cdot x^t + U_o \cdot h^{t-1} + b_o\right). \tag{9}$$

$$c^t = i^t \odot g\left(W_c \cdot x^t + U_c \cdot h^{t-1} + b_c\right) + f^t \odot c^{t-1}. \tag{10}$$

$$h^t = o^t \cdot tanh\left(c^t\right). \tag{11}$$

where i denotes the input gate output, f denotes the forget gate output, o denotes the output gate output, h^t denotes the hidden layer output at moment t, x^t denotes the input at moment t, W and U are the weight matrices, b is the corresponding bias vector, and c^t is the cell state update.

Although the MSCNN structure mentioned in the previous section can capture the global features of the feature map, the RNN structure is still preferred for the sequence modeling problem. The RNN model introduces additional learning parameters for each time step of the input data and addresses the presence of long-term dependencies in the sequences by remembering the historical information, thus retaining richer temporal features and facilitating more accurate modeling of the sequence data. In this paper, LSTM networks are used to extract the time domain features of LIBs, enabling better modeling of the long-term sequence dependencies. In the forward propagation process of the LSTM, by adopting the output features of the DAE, as extracted by the LSTM, as a sequence of time-ordered features and combining them with the features extracted from the convolutional features in the previous subsection, the LIB features at all levels can be fully fused, which helps to improve the RUL prediction ability for LIBs.

2.4. Hyperparameter optimization of DAE-MSCNN-LSTM

The goal of RUL prediction using a DAE-MSCNN-LSTM network is to obtain a nonlinear function $f(x)$ that minimizes the prediction error from a battery sample D that conforms to a natural distribution [32]. Specifically, the finite battery dataset is mapped to the function that minimizes the expected loss by learning the algorithm $f(x;\theta)$. The function can be expressed as:

$$f(x) = argminL(f(x), y). \tag{12}$$

where L is the loss function, $f(x)$ is the model-predicted value, and y is the true value. Our goal is to obtain the optimal model parameter θ that minimizes the value of $L(f(x), y)$. Here, this can be expressed as an optimization problem,

$$\theta^* = argmin\theta E(L(f(x), y)). \tag{13}$$

where θ is a parameter of the model, which is determined by a set of hyperparameters α in high-dimensional space and the training set D, $E(L(f(x), y))$ is the expected loss function, and θ^* is the optimal set of hyperparameters for the model.

The essence of the optimization process is to determine the optimal set of hyperparameters θ^* that minimizes the loss function on the training set D, resulting in a function $E(L(f(x), y))$ that minimizes the desired loss. This can be written as:

$$\theta^* = argminL(D;\theta). \tag{14}$$

where $L(D;\theta)$ denotes the loss function on the training set D.

In this paper, we use a grid search method to automatically select the optimal configuration during hyperparameter optimization. The grid search algorithm is an exhaustive search method that iterates through given hyperparameter combinations, calculates the performance of each combination, and selects the hyperparameter combination with the best performance [33].

First, let the hyperparameter search space be S, where s is the set of optional hyperparameters in dimension i of the search space. Suppose the objective function is $f(\theta)$, where θ denotes the hyperparameter combination. Grid search is performed by traversing all hyperparameter combinations in the search space and computing the value of the objective function, as shown below.

$$S = \left\{s_1, s_2, \dots, s_n\right\}. \tag{15}$$

$$\theta = \left\{\theta_1, \theta_2, \dots, \theta_k\right\}. \tag{16}$$

$$\varepsilon(\theta) = f(\theta) = \left\{f\left(\theta_1\right), f\left(\theta_2\right), \dots, f\left(\theta_k\right)\right\}. \tag{17}$$

Then, the model is constructed and trained using the current combination of hyperparameters to predice the objective function.

$$\hat{f}(\theta) = Net\left(\theta_i, D\right). \tag{18}$$

The above operation is performed for each hyperparameter combination θ_i in the grid search. Finally, based on the gird search results, the hyperparameter combination that yields the best performance on the objective function is selected as the optimal setting,

$$\theta_{best} = argmin_{\theta}\hat{f}(x). \tag{19}$$

2.5. Application of DAE-MSCNN-LSTM modeling

Assuming an EOL of C_{max}×70%, the actual and predicted RUL values of the LIB can be described as:

$$RUL = c_{eol} - c_s. \tag{20}$$

$$\widehat{RUL} = \hat{c}_{eol} - c_s. \tag{21}$$

where c_{eol} is the number of cycles to reach the termination state from the initial state in the EOL case, $\hat{c}_{eol}$ is the predicted number of cycles to reach the termination state from the initial state in the EOL case, and c_s is the number of cycles already used by the battery at the beginning of the prediction process. Moreover, $\hat{c}_{eol}$ is the number of cycles already used for the prediction by training the DAE-MSCNN-LSTM model on the existing feature data and the corresponding actual number of cycles,

$$\hat{c}_{eol}^{train} = f\left(U_{C,c_s}^H, U_{C,c_s-1}^H, \dots, U_{C,c_s-s_w+1}^H \mid U_{C,c_s+1}^H, \dots, U_{C,c_s+p_w}^H\right). \tag{22}$$

where U_{C,c_s}^H is the historical capacity before RUL prediction, s_w is the sliding window length, and p_w is the prediction window length. Algorithm 1 is a description of the method proposed in this paper.

Algorithm 1 Training process for spatio-temporal integration using DAE-MSCNN-LSTM networks.

```
Require: $\theta, trainingset \left\{U^H\right\}$,number of training iterations $N$
Ensure: $\widehat{RUL}$
while $N \neq 0$ do
    $\left\{x^h\right\}, y^h \Leftarrow D$
    $N \Leftarrow N - 1$
    function $Train_{rul}\left(U^H\right)$
        $x^h \Leftarrow Normalize\left\{x^h\right\}$
        for $e \Leftarrow 1$ to $Epoch_{rul}$ do
            $Model_{rul}, Loss_{rul} \Leftarrow$ Equation (1)-Equation (10)
        end for
    end function
end while
function $GridSearch(D)$
    $\theta_i \Leftarrow S$
    while $\theta_i$ is not null do
        $\hat{f}(\theta) \Leftarrow Train_{rul}(D)$ use $\theta_i$
        $\theta_i \Leftarrow \theta_{i+1}$
    end while
    $\theta_{best} \Leftarrow \theta_i$ by Equation (19)
end function
Step:
$\theta_{best} \Leftarrow GridSearch\left(U^H\right)$
$Model_{rul} \Leftarrow Train_{rul}\left(U^H\right)$ use $\theta_{best}$
$\hat{c}_{eol} \Leftarrow$ Equation (22)
$\widehat{RUL} = \hat{c}_{eol} - c_s$
```

3. Experiments

3.1. Experimental settings

3.1.1. Datasets

To evaluate the accuracy of our proposed method for long-term dependent prediction of the battery capacity under different operating conditions, several cycling test datasets of NASA[1] and CALCE[2] batteries with strong localized fluctuations and different capacity profiles were selected in this paper.

The NASA battery degradation data use one type of Li-ion battery through 3 distinct operating profiles: charge, discharge and impedance [34]. Four types of batteries widely used at NASA were selected for comparison, i.e., batteries B0005, B0006, B0007, and B0018. The detailed specifications of the selected NASA LIBs are provided in Table 1. Similarly, the CALCE team provides baseline cycle life data for LIB studies, a dataset that provides more useful baselines, such as continuous full and partial cycle profiles, storage profiles, and dynamic drive profiles, for comparison with more complex discharge profiles [35]. We selected batteries CS2_35, CS2_36, CS2_37 and CS2_38 from the CALCE dataset. The detailed specifications of these devices are provided in Table 2. In this study, the EOL of the batteries was set to 70% of the rated capacity, which is 1.40 and 0.77Ah for the NASA and CALCE datasets, respectively. Moreover, using the leave-one-out method, each of the sample batteries was used for testing, and the remaining batteries were used for training. Fig. 6 shows the capacity decay curves for the NASA and CALCE datasets.

3.1.2. Dataset preprocessing

To reduce the influence of input data distribution changes on neural networks, ensure the quality of data and improve the performance of the algorithm, the data must be normalized. For this reason, the min–max normalization method is used in this paper to normalize the data, thereby scaling the range of the values to [0,1]. This method can be expressed as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}. \tag{23}$$

where x is the input vector and x' is the normalized vector. Normalizing the data in this way reduces the data variance, avoids excessive influence of certain features on the model, and improves the algorithm's convergence speed and generalization ability.

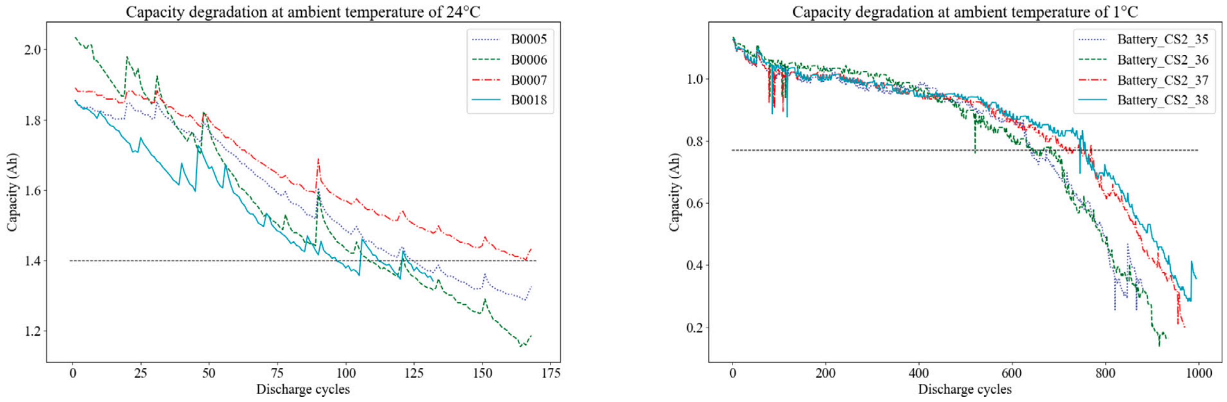
3.1.3. Baseline approaches

The DAE-MSCNN-LSTM method used in this article was compared with the following deep models:

- MLP [36]: Multiple fully connected layers are employed for RUL prediction of LIBs.
- LSTM [37]: Multiple LSTM units learn the nonlinear relationship between the battery capacity and time and predict the remaining battery life.
- GRU [38]: Multiple GRU units learn features from sequences.
- DeTransformer [29]: The DAE and transformer are combined for RUL prediction of LIBs.
- Auto-CNN-LSTM [12]: A fusion network based on an improved CNN and LSTM is used to mine deeper information from limited data.
- JFO-CFNN [39]: RUL prediction is achieved by combining cascaded forward neural networks (CFNNs) with innovative jellyfish optimization (JFO) techniques.
- DCNN [40]: A direct relationship between the condition monitoring data and RUL is constructed without a priori information.

[1] https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository

[2] https://web.calce.umd.edu/batteries/data.htm



(a) NASA batteries capacity decay data

(b) CALCE batteries capacity decay data

Fig. 6. Battery aging trends for two types of battery datasets.

Table 1
Detailed information of NASA dataset.

Battery	Discharge current (A)	Rated capacity (Ah)	Charing/discharing cut-off voltage (V)	Minimal charge current (mA)	Failure threshold (Ah)
B0005	2	2	4.2/2.7	20	1.4
B0006	2	2	4.2/2.5	20	1.4
B0007	2	2	4.2/2.3	20	1.4
B0018	2	2	4.2/2.5	20	1.4

Table 2
Detailed information of CALCE dataset.

Battery	Discharge current (A)	Rated capacity (Ah)	Charing/discharing cut-off voltage (V)	Minimal charge current (mA)	Failure threshold (Ah)
CS2_35	1.1	1.1	4.2/2.7	50	0.77
CS2_36	1.1	1.1	4.2/2.7	50	0.77
CS2_37	1.1	1.1	4.2/2.7	50	0.77
CS2_38	1.1	1.1	4.2/2.7	50	0.77

3.2. Evaluation metrics

In this paper, three evaluation metrics, i.e., the relative error (RE), mean absolute error (MAE) and root mean square error (RMSE), are chosen to evaluate the RUL prediction performance. In regard to RE, MAE and RMSE, the lower the value is, the higher the prediction accuracy. The formulas for calculating the three metrics are as follows.

$$RE = \frac{\left|RUL^{pred} - RUL^{true}\right|}{RUL^{true}} = \frac{\left|\hat{y}_i - y_i\right|}{y_i} \times 100\%. \tag{24}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|. \tag{25}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2} \tag{26}$$

where $\hat{y}_i$ is the predicted value, y_i is the true value, and n denotes the length of the sequence.

Then, leave-one-out evaluation is used in this paper to randomly select one battery sample as the test set and the remaining battery samples as the training set. The model can be robustly tested on different data points and that the model can accurately and reliably predict the RUL of LIBs. Finally, after five iterations, the average score of all the batteries is determined.

3.3. Parameter settings

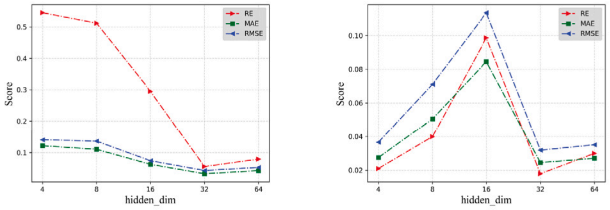
The model proposed in this paper encompasses five key parameters: sliding window size (s), learning rate (l), hidden layer size (h),number of layers (n), and rate per task (λ). Note that (s) is generally set to 5%-10% of the sequence length. In this experiment, (s) is set to 16 and 64 for the NASA and CALCE datasets, respectively. In addition, we verify the effect of the sliding time window size on the model in the Experimental section. (λ) is applied to DeTransformer and our model only.

Table 3
Optimal parameters of RE scores for two datasets.

Method	NASA					CALCE				
	s	l	h	n	λ	s	l	h	n	λ
MLP	16	0.01	8	2	–	64	0.01	64	4	–
LSTM	16	0.001	64	2	–	64	0.001	32	2	–
GRU	16	0.001	64	2	–	64	0.001	64	2	–
DeTransformer	16	0.005	32	1	10^{-5}	64	0.001	32	1	10^{-3}
Ours	16	0.005	64	2	10^{-4}	64	0.0005	64	2	10^{-3}

All the parameters except (s) are passed through the grid search algorithm to find the optimal parameter combination. l is choose from $\left\{10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}\right\}$, h is choose from $\{4, 8, 16, 32, 64\}$, n is choose from $\{1, 2, 3, 4\}$, λ is set from $\left\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\right\}$.

RE is strongly correlated with RUL prediction of LIBs. Table 3 shows the optimal parameters of some of the methods for both datasets in terms of RE. In addition, to verify the accuracy of the grid search algorithm in selecting the optimal parameter combination, we adjust the size of the hidden layer of the model while ensuring that the other metrics remain unchanged. The results for the two datasets are shown in Fig. 7. Notably, on both datasets, the model performs best when the number of hidden layers is 32, which is consistent with our search results.



(a) Effect of hidden_dim in NASA

(b) Effect of hidden_dim in CALCE

Fig. 7. Verifying the accuracy of the Grid Search algorithm by adjusting hidden_dim.

Table 4
Performances of deep learning models in NASA and CALCE datasets.

Method	NASA			CALCE		
	RE	MAE	RMSE	RE	MAE	RMSE
MLP	0.3851	0.1379	0.1541	0.4018	0.1557	0.2038
LSTM	0.2648	0.0829	0.0905	0.0902	0.0582	0.0736
GRU	0.3044	0.0806	0.0921	0.1319	0.0671	0.0946
DeTransformer	0.2252	0.0713	0.0802	0.0764	0.0613	0.0705
Auto-CNN-LSTM	–	–	0.0484	–	–	–
JFO-CFNN	–	–	0.0461	–	–	–
DCNN	0.0790	0.0469	0.0609	0.0198	0.0319	0.0427
Ours	**0.0554**	**0.0325**	**0.0429**	**0.0179**	**0.0245**	**0.0319**

This experiment is based on the PyCharm development environment and a Windows operating system, including an AMD Ryzen 7 6800H with a Radeon Graphics 3.20 GHz CPU and a GeForce RTX 3080 GPU, with 12 GB of video memory.

4. Results and analysis

4.1. Overall performance

First, experiments are conducted on two datasets, namely, NASA and CALCE, and a fair comparison with the baseline model is provided. For the reproducible baseline model, we first apply the grid search algorithm to find the optimal parameter configuration and subsequently reproduce the relevant experiments to obtain prediction results. In addition, to mitigate the effect of randomness, we repeat all the comparison experiments 10 times and provide the average performance of the models in this paper. For the baseline models that are difficult to reproduce, we use the best results from their papers for comparison. Table 4 provides the average performance of all the models for each dataset under the three evaluation metrics, with the best average performance marked in bold.

Table 4 clearly indicates the following: (1) the model proposed in this paper performs significantly better than the currently commonly used prediction models in terms of the three evaluation indices of RE, MAE and RMSE in RUL prediction for LIBs. (2) On both datasets, the DAE-MSCNN-LSTM model achieves excellent performance regardless of whether the capacity sequence is long or short. For short sequence data, it is often difficult to provide enough information to support accurate prediction. The DAE-MSCNN-LSTM model can extract sequence feature information from both temporal and spatial perspectives at multiple scales with high prediction accuracy. Due to its multilevel structure and multiscale feature extraction capability, the model can better adapt to data complexity and variability under different scenarios, which improves the model generalization ability and prediction accuracy. (3) In RUL prediction, for LIBs, the consideration of temporal order information between sequence data is indispensable due to the temporal correlation of sequence data of LIBs. Among the baseline methods, the MLP achieves the worst effect because it does not consider temporal correlation. The DAE-MSCNN-LSTM model uses the parallel structure of the MSCNN and LSTM networks. The MSCNN network extracts spatially localized features from the input data at multiple time scales, while the LSTM network extracts the long-term dependencies of the sequence data and fuses the features at each level in the LIB sequence data, providing an effective model architecture for RUL prediction of LIBs. (4) DeTransformer performs slightly better than other RNN network variants, mainly because of the attention mechanism and parallel computations in the structure of the Transformer model, which can better capture long-term dependent features and global contextual information. However, considering the computational and storage costs and task specificity, as well as the ability to better support practical application scenarios, the LSTM network is ultimately chosen in this paper to extract the temporal features of LIB data sequences.

Table 5
Comprehensive performance of different methods in RUL prediction task.

Method	LSTM	DeTransformer	Ours
FLOPs(G)	0.0233	0.0002	0.0138
Parameters(M)	0.0543	0.0005	0.0326
Training time(s)	44.4531	289.3437	275.8125
Inference time(s)	0.0937	0.5312	0.4843

Table 6
Results of ablation experiments.

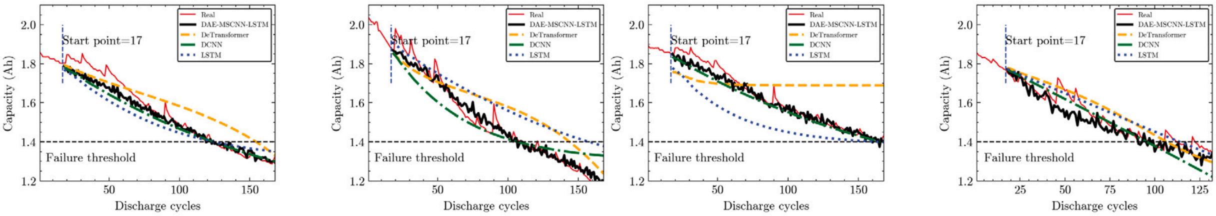
Method	NASA			CALCE		
	RE	MAE	RMSE	RE	MAE	RMSE
MSCNN-LSTM	0.1256	0.0553	0.0690	0.0543	0.0360	0.0501
DAE-CNN-LSTM	0.0924	0.0553	0.0679	0.0568	0.0403	0.0464
Ours arranged in series	0.1015	0.0574	0.0664	0.0532	0.0290	0.0468
Ours	**0.0554**	**0.0325**	**0.0429**	**0.0179**	**0.0245**	**0.0319**

To more intuitively analyze the prediction effect of our DAE-MSCNN-LSTM model, we randomly selected a cell from each of the two cell degradation datasets and compared the prediction results with the actual capacity degradation curves. In addition, we compared several of the above baseline methods. All the results are shown in Figs. 8 and 9. Notably, compared to the other methods, our model prediction results exhibit the most similar trajectories to the real RUL values. Moreover, most of our predictions are close to or smaller than the real RUL values, which is expected in practical applications since RUL overestimation may yield more serious consequences than underestimation.

4.2. Comprehensive performance analysis of the DAE-MSCNN-LSTM model

Due to the limitation of the existing conditions, it is difficult to directly obtain the computational efficiency of our proposed network architecture when embedded in a real Battery Management System (BMS) system. Therefore, it is necessary to analyze the comprehensive performance of the above methods to find a better RUL prediction model that weighs the various metrics, and to laterally verify the effectiveness of our network architecture in terms of computational performance after embedding the BMS system.

The evaluation metrics include the number of FLOPs, parameters, training time, and inference time. The number of FLOPs is the number of multiplications and is an indirect measure of the computational complexity. This parameter is an indirect metric that affects the storage size of the model. Specifically, we adopt the NASA dataset as an example and compare the performance of the DAE-MSCNN-LSTM model with that of two representative methods, namely, the LSTM and DeTransformer, in the same experimental setting, as summarized in Table 5. Although our method does not perform the best, in practical applications, although the LSTM is fast in regard to training and inference, it requires a large number of parameters, and its prediction effect is not very satisfactory. DeTransformer solves the problem of a low prediction accuracy that exists in the LSTM, but this comes at the cost of time. Our proposed method is the best in terms of the prediction results, and the other metrics are within acceptable limits. Therefore, compared to existing methods, we believe that the computational efficiency of our proposed model is optimal when embedded in a real BMS system.



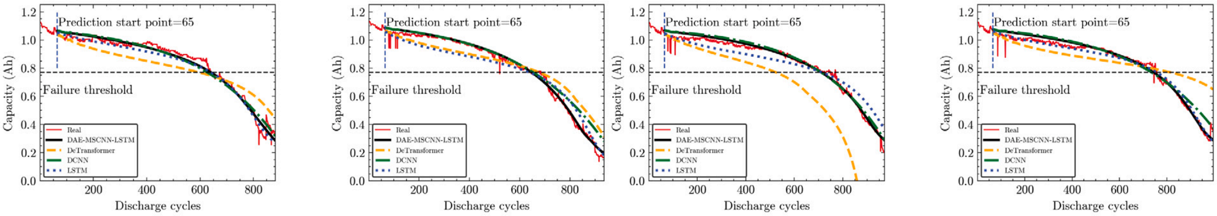
(a) Battery B0005

(b) Battery B0006

(c) Battery B0007

(d) Battery B0018

Fig. 8. RUL prediction results of the proposed method for NASA dataset.



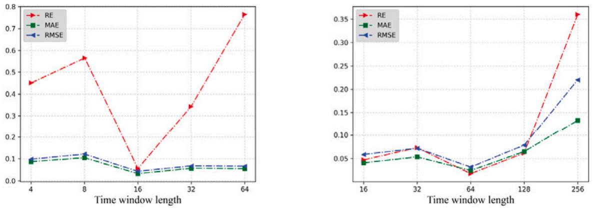
(a) Battery CS2_35

(b) Battery CS2_36

(c) Battery CS2_37

(d) Battery CS2_38

Fig. 9. RUL prediction results of the proposed method for CALCE dataset.



(a) Performance on the NASA dataset.

(b) Performance on the CALCE dataset.

Fig. 10. Impact of different time window sizes on model performance.

4.3. Impact of the sliding time window size

To verify the effect of the sliding time window size on the model, we perform experiments on NASA and CALCE datasets respectively. Using the NASA dataset as an example, we set s to 4, 8, 16, 32, and 64. The results are shown in Fig. 10. Fig. 10 reveals that on the NASA dataset, the model predicts best with the smallest RE, MAE and RMSE values for a time window of 16. This occurs because smaller time windows do not provide sufficient historical information to capture the long-term evolution of the battery, while excessively large time windows cause the mixing of measurements from different operating conditions and blur information on the current state of the battery.

4.4. Ablation experiments

Finally, to validate the mechanism of action of the model, extensive ablation experiments were conducted. Specifically, we evaluated two key components, namely, the DAE and MSCNN, as well as the parallel feature extraction design. To this end, we perform three experiments in this section: an MSCNN-LSTM network without the DAE, a DAE-CNN-LSTM network without the MSCNN, and a network with the DAE, MSCNN and LSTM arranged in series. In the third experiment, the input data are first processed via the DAE for noise reduction, and the output is regarded as the input of the tandem MSCNN and LSTM networks. To verify the generality of the experimental results, we conduct the experiments on the two datasets separately, and the experimental results are listed in Table 6. The experimental results show that the model performance significantly decreases when either the DAE or MSCNN is removed, indicating that data denoising and multifeature extraction are valuable for RUL prediction of LIBs. Moreover, our proposed model outperforms the version with serial feature extraction, indicating that our parallelization design can effectively mitigate the interactions between information sequences.

5. Conclusion

In this paper, the DAE-MSCNN-LSTM model is proposed for RUL prediction of LIBs. First, a denoising autoencoder is used to enhance the original data. Then, the MSCNN and LSTM networks are used in parallel, and the MSCNN network is employed to mine spatial information and extract features at multiple scales. The LSTM network extracts the time series, and the features obtained by the two networks are fused and fed into the MLP layer to predict the RUL of the LIB. Finally, to obtain the optimal parameter combination for the model, the grid search algorithm is used in this paper to fully traverse the search space and obtain the global optimal solution. The proposed DAE-MSCNN-LSTM model outperforms the existing generalized prediction models on both the public NASA and CALCE datasets. In the future, we plan to improve our model by exploring advanced computational methods such as adaptive multimodel fusion and model combination optimization. In addition, we will incorporate additional information external to the battery, such as different operating temperatures and currents, to improve the lithium-ion battery life prediction accuracy.

CRediT authorship contribution statement

Zihao Lv: Conceptualization, Formal analysis, Methodology, Writing – original draft. **Yi Song:** Investigation, Validation, Visualization. **Chunlin He:** Conceptualization, Resources, Supervision. **Liming Xu:** Data curation, Funding acquisition, 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

The authors do not have permission to share data.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62206224), the Postdoctoral Science Foundation of China (No. 2023M732428), the Natural Science Foundation of Sichuan Province, China (No. 2022NSFSC0866), and the Doctoral Research Innovation Project, China (No. 21E025).

References

[1] H. Li, K. Ota, M. Dong, Energy cooperation in battery-free wireless communications with radio frequency energy harvesting, ACM Trans. Embed. Comput. Syst. 17 (2) (2018).

[2] Y. Su, H. Cai, J. Huang, The cooperative output regulation by the distributed observer approach, Int. J. Netw. Dyn. Intell. (2022) 20–35.

[3] K. Liu, Y. Shang, Q. Ouyang, W.D. Widanage, A data-driven approach with uncertainty quantification for predicting future capacities and remaining useful life of lithium-ion battery, IEEE Trans. Ind. Electron. 68 (4) (2020) 3170–3180.

[4] D. Roman, S. Saxena, V. Robu, M. Pecht, D. Flynn, Machine learning pipeline for battery state-of-health estimation, Nat. Mach. Intell. 3 (5) (2021) 447–456.

[5] S. Yuchen, L. Datong, H. Yandong, Y. Jinxiang, P. Yu, Satellite lithium-ion battery remaining useful life estimation with an iterative updated RVM fused with the KF algorithm, Chin. J. Aeronaut. 31 (1) (2018) 31–40.

[6] Q. Xu, M. Wu, E. Khoo, Z. Chen, X. Li, A hybrid ensemble deep learning approach for early prediction of battery remaining useful life, IEEE/CAA J. Autom. Sin. 10 (1) (2023) 177–187.

[7] H. He, F. Sun, Z. Wang, C. Lin, C. Zhang, R. Xiong, J. Deng, X. Zhu, P. Xie, S. Zhang, Z. Wei, W. Cao, L. Zhai, China's battery electric vehicles lead the world: Achievements in technology system architecture and technological breakthroughs, Green Energy Intell. Transp. 1 (1) (2022) 100020.

[8] S. Suh, P. Lukowicz, Y.O. Lee, Generalized multiscale feature extraction for remaining useful life prediction of bearings with generative adversarial networks, Knowl.-Based Syst. 237 (2022) 107866.

[9] Y. Wang, J. Tian, Z. Sun, L. Wang, R. Xu, M. Li, Z. Chen, A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems, Renew. Sustain. Energy Rev. 131 (2020) 110015.

[10] L. Yang, Y. Cai, Y. Yang, Z. Deng, Supervisory long-term prediction of state of available power for lithium-ion batteries in electric vehicles, Appl. Energy 257 (2020) 114006.

[11] X. Sui, S. He, S.B. Vilsen, J. Meng, R. Teodorescu, D.-I. Stroe, A review of non-probabilistic machine learning-based state of health estimation techniques for lithium-ion battery, Appl. Energy 300 (2021) 117346.

[12] L. Ren, J. Dong, X. Wang, Z. Meng, L. Zhao, M.J. Deen, A data-driven auto-CNN-LSTM prediction model for lithium-ion battery remaining useful life, IEEE Trans. Ind. Inform. 17 (5) (2020) 3478–3487.

[13] J. Tian, R. Xiong, Q. Yu, Fractional-order model-based incremental capacity analysis for degradation state recognition of lithium-ion batteries, IEEE Trans. Ind. Electron. 66 (2) (2018) 1576–1584.

[14] X. Song, Y. Lu, F. Wang, X. Zhao, H. Chen, A coupled electro-chemo-mechanical model for all-solid-state thin film Li-ion batteries: The effects of bending on battery performances, J. Power Sources 452 (2020) 227803.

[15] Y. Wang, M. Li, Z. Chen, Experimental study of fractional-order models for lithium-ion battery and ultra-capacitor: Modeling, system identification, and validation, Appl. Energy 278 (2020) 115736.

[16] B. Gou, Y. Xu, X. Feng, State-of-health estimation and remaining-useful-life prediction for lithium-ion battery using a hybrid data-driven method, IEEE Trans. Veh. Technol. 69 (10) (2020) 10854–10867.

[17] Z. Wang, N. Liu, Y. Guo, Adaptive sliding window LSTM NN based RUL prediction for lithium-ion batteries integrating LTSA feature reconstruction, Neurocomputing 466 (2021) 178–189.

[18] X. Shu, G. Li, Y. Zhang, J. Shen, Z. Chen, Y. Liu, Online diagnosis of state of health for lithium-ion batteries based on short-term charging profiles, J. Power Sources 471 (2020) 228478.

[19] Z. Xue, Y. Zhang, C. Cheng, G. Ma, Remaining useful life prediction of lithium-ion batteries with adaptive unscented kalman filter and optimized support vector regression, Neurocomputing 376 (2020) 95–102.

[20] C. Zhang, M. Dong, K. Ota, Enabling computational intelligence for green Internet of Things: Data-driven adaptation in LPWA networking, IEEE Comput. Intell. Mag. 15 (1) (2020) 32–43.

[21] G. Cheng, X. Wang, Y. He, 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 232 (2021) 121022.

[22] X. Li, C. Yuan, Z. Wang, State of health estimation for Li-ion battery via partial incremental capacity analysis based on support vector regression, Energy 203 (2020) 117852.

[23] Z. Lyu, G. Wang, R. Gao, Li-ion battery prognostic and health management through an indirect hybrid model, J. Energy Storage 42 (2021) 102990.

[24] Z. Zhou, Y. Liu, M. You, R. Xiong, X. Zhou, Two-stage aging trajectory prediction of LFP lithium-ion battery based on transfer learning with the cycle life prediction, Green Energy Intell. Transp. 1 (1) (2022) 100008.

[25] B. Zraibi, C. Okar, H. Chaoui, M. Mansouri, Remaining useful life assessment for lithium-ion batteries using CNN-LSTM-DNN hybrid method, IEEE Trans. Veh. Technol. 70 (5) (2021) 4252–4261.

[26] V.-T. Hoang, K.-H. Jo, Pydmobilenet: Pyramid depthwise separable convolution networks for image classification, in: International Symposium on Industrial Electronics, 2019, pp. 1430–1434.

[27] J. Chen, X. Feng, L. Jiang, Q. Zhu, State of charge estimation of lithium-ion battery using denoising autoencoder and gated recurrent unit recurrent neural network, Energy 227 (2021) 120451.

[28] Y.-W. Lu, C.-Y. Hsu, K.-C. Huang, An autoencoder gated recurrent unit for remaining useful life prediction, Processes 8 (9) (2020).

[29] D. Chen, W. Hong, X. Zhou, Transformer network for remaining useful life prediction of lithium-ion batteries, Ieee Access 10 (2022) 19621–19628.

[30] X. Ren, S. Liu, X. Yu, X. Dong, A method for state-of-charge estimation of lithium-ion batteries based on PSO-LSTM, Energy 234 (2021) 121236.

[31] Y. Huang, Y. Tang, J. VanZwieten, Prognostics with variational autoencoder by generative adversarial learning, IEEE Trans. Ind. Electron. 69 (1) (2021) 856–867.

[32] S. Ferahtia, A. Djeroui, H. Rezk, A. Chouder, A. Houari, M. Machmoum, Optimal parameter identification strategy applied to lithium-ion battery model, Int. J. Energy Res. 45 (11) (2021) 16741–16753.

[33] L. Yao, Z. Fang, Y. Xiao, J. Hou, Z. Fu, An intelligent fault diagnosis method for lithium battery systems based on grid search support vector machine, Energy 214 (2021) 118866.

[34] J. Wang, S. Liu, S. Wang, Q. Liu, H. Liu, H. Zhou, J. Tang, Multiple indicators-based health diagnostics and prognostics for energy storage technologies using fuzzy comprehensive evaluation and improved multivariate grey model, IEEE Trans. Power Electron. 36 (11) (2021) 12309–12320.

[35] M. Lin, X. Zeng, J. Wu, State of health estimation of lithium-ion battery based on an adaptive tunable hybrid radial basis function network, J. Power Sources 504 (2021) 230063.

[36] I. Sanz-Gorrachategui, P. Pastor-Flores, M. Pajovic, Y. Wang, P.V. Orlik, C. Bernal-Ruiz, A. Bono-Nuez, J.S. Artal-Sevil, Remaining useful life estimation for LFP cells in second-life applications, IEEE Trans. Instrum. Meas. 70 (2021) 1–10.

[37] W. Li, N. Sengupta, P. Dechent, D. Howey, A. Annaswamy, D.U. Sauer, One-shot battery degradation trajectory prediction with deep learning, J. Power Sources 506 (2021) 230024.

[38] J.C. Chen, T.-L. Chen, W.-J. Liu, C. Cheng, M.-G. Li, Combining empirical mode decomposition and deep recurrent neural networks for predictive maintenance of lithium-ion battery, Adv. Eng. Inform. 50 (2021) 101405.

[39] S. Ansari, A. Ayob, M.H. Lipu, A. Hussain, M.G. Abdolrasol, M.A.A.M. Zainuri, M.H.M. Saad, Optimized data-driven approach for remaining useful life prediction of lithium-ion batteries based on sliding window and systematic sampling, J. Energy Storage 73 (2023) 109198.

[40] H. Li, W. Zhao, Y. Zhang, E. Zio, Remaining useful life prediction using multi-scale deep convolutional neural network, Appl. Soft Comput. 89 (2020) 106113.