A life-prediction method for lithium-ion batteries based on a fusion model and an attention mechanism*

WANG Xian-bao (王宪保)**, **WU Fei-teng** (吴飞腾), **and YAO Ming-hai** (姚明海)
College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

(Received 17 December 2019; Revised 5 February 2020)

The current life-prediction models for lithium-ion batteries have several problems, such as the construction of complex feature structures, a high number of feature dimensions, and inaccurate prediction results. To overcome these problems, this paper proposes a deep-learning model combining an autoencoder network and a long short-term memory network. First, this model applies the characteristics of the autoencoder to reduce the dimensionality of the high-dimensional features extracted from the battery data set and realize the fusion of complex time-domain features, which overcomes the problems of redundant model information and low computational efficiency. This model then uses a long short-term memory network that is sensitive to time-series data to solve the long-path dependence problem in the prediction of battery life. Lastly, the attention mechanism is used to give greater weight to features that have a greater impact on the target value, which enhances the learning effect of the model on the long input sequence. To verify the efficacy of the proposed model, this paper uses NASA's lithium-ion battery cycle life data set.
Document code: A **Article ID:** 1673-1905(2020)06-0410-8
DOI https://doi.org/10.1007/s11801-020-9214-y

Lithium-ion batteries possess many advantages over traditional batteries, such as high output voltage, high energy density[1], low self-discharge rate, long cycle life, and high reliability[2]. They have been widely applied in vehicles, household equipment, communications, the aerospace industry, and so on[3]. Lithium-ion batteries that exceed their service life often lead to accidents, such as fires and explosions. The accurate prediction of lithium-ion batteries' remaining useful life (RUL) plays an important role in state estimation and health management of lithium-ion batteries.

The methods for predicting a lithium-ion battery's RUL can be divided into three categories: experience-based, model-based, and data-driven. The experience-based method uses the battery's history data to estimate its service life, but it cannot describe the physical and chemical changes inside the battery. This method is applicable only under special conditions and has gradually been abandoned in favor of other methods.

The model-based method integrates the material characteristics, the decay mechanism, and the operating environment inside the battery[4]. However, due to the complexity of chemical reactions in lithium-ion batteries and the fact that the state of lithium-ion batteries is affected by temperature[5] and humidity in the working environment[6], predictions from this method cannot meet the accuracy requirement.

The data-driven method does not consider the complex material characteristics and changes inside the battery; it relies on previously observed data to predict the state of the system[7] or infer the RUL by matching historically similar models[8]. With the development of computer technology, neural networks have also been widely applied[9]. Due to their self-learning ability, neural networks can obtain more accurate data without relying on an accurate model[10]. Hence, this technology has also been adopted to predict the RUL of lithium-ion batteries. However, the methods mentioned above still face many challenges, such as determining the threshold, determining which data describe normal behavior, and solving the problem of over-fitting[11].

The key to improving the accuracy of predictions of lithium-ion batteries' RUL is to select the characteristics that can significantly reflect the health state of the battery. The characteristics commonly applied to characterize the degradation of battery health include capacity and impedance. However, these parameters cannot be measured directly. The parameters that can be measured directly, such as current and voltage, have higher dimensions. Thus, they cannot be used directly. In addition, most of the current mainstream methods are based on a single data-driven model; they do not benefit from the advantages of combining different prediction models, and so the performance of such models is less than ideal.

* This work has been supported by the National Natural Science Foundation of China (No.61871350), the Zhejiang Science and Technology Plan Project (No.2019C011123), and the Zhejiang Province Basic Public Welfare Research Project (No.LGG19F030011).
** E-mail: wxb@zjut.edu.cn

To overcome the problems described above, this paper proposes a deep-learning method for the RUL prediction of lithium-ion batteries using the fusion model. The method integrates an autoencoder (AE) network[12] and a long short-term memory (LSTM) network[13]. The dimensions of the extracted multi-dimensional key features are reduced to realize the fusion of complex time-domain features, which overcomes the problems of model information redundancy and low calculation efficiency. The use of LSTM can solve the long-range dependence problem in battery life prediction. An attention mechanism is introduced in the LSTM network, so that features that have a greater impact on the target value have greater weight in the optimization of the output result.

The overall framework of the RUL prediction method for lithium-ion batteries is shown in Fig.1. The method consists of four modules: feature-extraction module, feature-fusion module, data-conversion and normalization module, and life-prediction module. Among them, the feature-extraction module includes the features extracted from the charging stage and the discharging stage. The feature-fusion module uses an autoencoder to reduce the features to 15 dimensions. The data-conversion and normalization module needs to convert the input of unsupervised data into supervised data and perform normalization. These three modules are collectively referred to as the data preprocessing module. The life-prediction module uses a long short-term memory network and an attention mechanism.

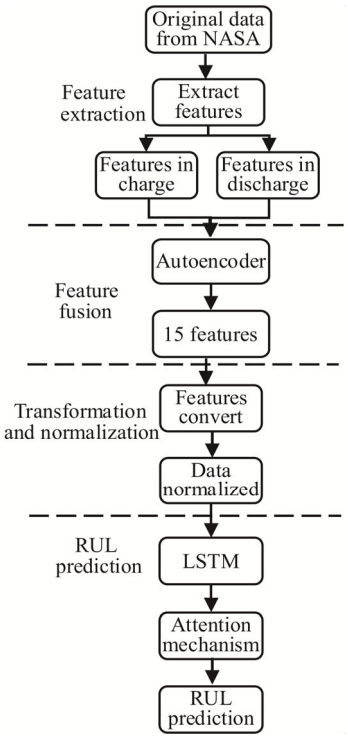

Fig.1 Overall framework of RUL prediction for lithium-ion batteries

The method proposed in this paper preprocesses the data as follows. First, the features are extracted from the original data. Second, the AE model is applied to complete the feature fusion. Third, the unsupervised data are converted into supervised data through feature transformation, and the data are normalized. This chapter discusses these three processes.

In general, the information representation is positively correlated with the information dimension. The more features that are extracted from the original data, the more accurate the prediction. However, an excessive number of features would make the calculation more complex. The characterization parameters of battery life are very abundant, and so it is necessary to select appropriate parameters to characterize the aging and actual performance of the batteries, ensuring that the parameters are still highly reliable and accurate under different aging conditions. After a great deal of investigation and experiment, the typical characteristics of the charge–discharge cycle are extracted in the following way.

The features extracted from the charging process are different from those extracted from the discharge process. During charging, the feature extraction of the terminal voltage is shown as

$$(t_{\min[i]},\ v_i),\quad s.t.\ v_i \geq 4.2\ \text{V},\qquad i=1,2,3,...n, \tag{1}$$

where $t_{\min[i]}$ refers to the time when the terminal voltage first reaches the maximum value, v_i refers to the value when the terminal output voltage first reaches the maximum value, and n is the sample size. The feature extraction of the output current is shown as

$$(t_{\min[i]}, A_i),\quad s.t.\ A_i \leq 1.5\ \text{V},\qquad i=1,2,3,...n, \tag{2}$$

where $t_{\min[i]}$ refers to the time when the battery output current starts to decrease, A_i stands for the current value when the battery terminal current starts to decrease, and n is the sample size. The feature extraction of the temperature is shown as

$$(t_{\text{bc}}T,\ T_{\text{bc}})=\{(t_i,T_i)|\max(n)\}\ ,\qquad i=1,2,3,...n, \tag{3}$$

where $t_{\text{bc}}T$ refers to the time when the battery temperature reaches its maximum, T_{bc} stands for the maximum temperature value, and n is the sample size. The feature extraction of the current is shown as

$$(t_{\min[i]}, A_i),\quad s.t.\ A_i \leq 1.5\ \text{V},\qquad i=1,2,3,...n, \tag{4}$$

where $t_{\min[i]}$ refers to the time when the battery's charge current starts to decrease, A_i represents the current value when the measured current starts to decrease, and n is the sample size. The feature extraction of the voltage is shown as

$$(t_{\text{mcv}},v_{\text{bc}})=\{(t_i,v_i)|\max(v_i)\},\qquad i=1,2,3,...n, \tag{5}$$

where t_{mcv} refers to the time when the measured voltage of the battery reaches its maximum, v_{bc} stands for the maximum value of the measured voltage, and n is the sample size.

During discharge, the feature extraction of the terminal voltage is shown as

$$(t_{\text{bc}}T,\ T_{\text{dc}})=\{(t_i,v_i)|\max(v_i)\},\qquad i=1,2,3,...n, \tag{6}$$

where $t_{\text{bc}}T$ refers to the time when the terminal voltage of the battery reaches its minimum value, T_{dc} represents the

minimum voltage value, and n is the sample size. Feature extraction of the output current is shown as

$$(t_{\min[i]}, A_i), \quad s.t.\ A_i > -2\ \text{A}, \qquad i=1,2,3,...n, \tag{7}$$

where $t_{\min[i]}$ refers to the time when the battery's output current starts to rise, A_i indicates the current value when the output current starts to rise, and n is the same size. Feature extraction of the temperature is shown as

$$(t_{\text{bc}}T, T_{\text{dc}})=\{(t_i,T_i)|\max(T_i)\}, \qquad i=1,2,3,...n, \tag{8}$$

where $t_{\text{bc}}T$ indicates the time when the battery temperature reaches its maximum value, T_{dc} represents the maximum temperature value, and n is the sample size. Feature extraction of the load measurement current is shown as

$$(t_{\min[i]}, A_i), \quad s.t.\ A_i > -2\ \text{A}, \qquad i=1,2,3,...n, \tag{9}$$

where $t_{\min[i]}$ stands for the time when the load measurement current starts to rise, A_i indicates the current value when the load measurement current starts to rise, and n is the sample size. Feature extraction of the load voltage is shown as

$$(t_{\text{bc}}T, v_{\text{bc}})=\{(t_i,v_i)|\max(v_i)\ s.t.\ v_i\neq 0\},\ i=1,2,3,...n, \tag{10}$$

where $t_{\text{bc}}T$ represents the time when the load voltage reaches its maximum value, v_{bc} refers to the maximum load voltage value, and n is the sample size. The battery capacitance C is extracted directly.

As the number of extracted features increases, so too does the correlation of features, which would lead to redundancy in the model information and low computational efficiency. It is therefore necessary to reduce the dimension of features to improve the efficiency of the model. There are many time-domain features in the data set. Moreover, the combination of different time-domain features is very complex. It is therefore necessary to determine the appropriate method of dimension reduction.

Common dimension-reduction methods include principal component analysis (PCA)[14], independent component analysis (ICA)[15], and linear discriminant analysis (LDA). The PCA method finds the direction of the greatest change in the data and then projects each datum into the coordinate system formed by these directions. This method is widely applied in data compression, redundancy elimination, elimination of data noise, and so on. The ICA method is an effective multi-dimensional signal-processing technology that aims to extract unknown independent signals from known mixed signals, thus removing the correlation between the signals. The LDA method, widely applied in the field of pattern recognition, is a dimension-reduction technique for supervised learning. It has the advantage of using prior knowledge and experience of categories in the process of dimension reduction. However, PCA and LDA require that the data conform to the Gaussian distribution, and ICA needs strict data distribution. Hence, none of them can be applied to the data set used in this paper.

An autoencoder is an unsupervised machine-learning technology that uses the low-dimensional data generated by a neural network to represent high-dimensional input. In the traditional linear dimension-reduction method, the linearity limits the dimensions of features that can be extracted. In contrast to these traditional methods, AE uses an inherent nonlinear neural network to overcome these limitations. Furthermore, its requirements for data distribution are not strict, and its dimension-reduction performance is good. This paper therefore uses an AE as the dimension-reduction network.

An AE is a non-supervised algorithm with a fully symmetric network structure[16]. It makes an attempt to learn a constant function so that the output result is equal to the tag given by the input data. In this case, when the output is equal to the input, a hidden neuron may be used to represent the input, as shown in Fig.2 (taking a three-layer network as an example).

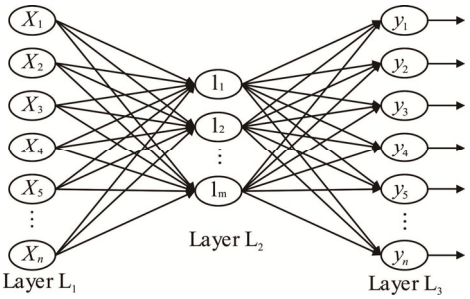

Fig.2 Three-layer autoencoder network structure

In the figure, $\{x^1, x^2,... x^n\}$refer to the input data of the neural network, and n is the dimension of the input data. After the encoding process, the n-dimensional input is compressed to become m-dimensional, expressed as $\{l^1, l^2,... l^m\}$. Subsequently, L_2 output is sent to the decoder, which is decoded to obtain the neural network's output: $\{y^1, y^2,... y^n\}$. By training the encoder and decoder, we make y^i equal to x^i. After the training is complete, $\{l^1, l^2,... l^m\}$ can be used to express $\{x^1, x^2,... x^n\}$, so as to realize the dimension reduction. The algorithm is as follows.

The encoding process of the initial feature set from the input layer to the hidden layer is shown as

$$h_i=g\theta_1(x_i)=\sigma(w_1+x_i+b_1), \qquad i=1,2,3,...m, \tag{11}$$

where h_i is the fusion feature set, x_i is the initial feature set, w_1 is the weight set, b_1 is the fixed parameter, and m is the number of samples.

The decoding process from the hidden layer to the output layer is shown as

$$x_i'= g\theta_2(x_i)= \sigma(w_2h_i+b_2), \qquad i=1,2,3,...m, \tag{12}$$

where x_i' is the decoding feature set, w_2 is a set of weights, and b_2 is the threshold. The reconstruction error loss function of the data set x is shown as

$$J_E(W,b)=\frac{1}{m}\sum_{i=1}^{m}\frac{\left\|x_i'-x_i\right\|^2}{2}X_i, \tag{13}$$

where m is the number of samples.

In this paper, through the coding function of the AE, the dimensions in the fusion of time-domain features are

reduced from 21 to 15. Due to the characteristics of the LSTM network, it is necessary to convert the unsupervised input data into supervised data. The process is as follows.

First, the total number of battery charging and discharge cycles is calculated as the battery's total cycle life j. Second the label value y_i of the i-th charge-discharge cycle ($0 < i < j$) is calculated by

$$y_i = j + 1 - i . \tag{14}$$

Finally, x_i and y_i are combined to obtain the supervised data (x_i, y_i) as the input of the LSTM-prediction model.

To eliminate the negative effects of different values, the range of all extracted feature values is converted to [0,1] by the following normalization method:

$$X^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{15}$$

where $x_{\max}$ is the maximum value of the sample data, and $x_{\min}$ is the minimum value of the sample data.

As a data-driven prediction method, the LSTM network is widely applied in the field of prediction and battery-health management. The network does not know the internal physical and chemical reactions of the prediction object; to carry out the prediction, it can use only the historical sequence training network of the prediction object. In contrast to the traditional data-driven methods (such as autoregressive models and autoregressive moving-average models), LSTM networks have stronger nonlinear approximation capabilities. The sample data set for a lithium battery is a long time series. Its prediction task is very complex, and the input information is too far apart so that the common neural network is difficult to remember. Through improvement of the classic recurrent neural network implicit layer, LSTM enables the network to handle long-distance information, which also overcomes the problem of gradient disappearance and gradient explosion. The LSTM network thus satisfies the prediction demand in this paper. This section describes how to apply an LSTM network in combination with an attention mechanism to predict battery life.

Long short-term memory is a time-recurrent neural network, whose structure is shown in Fig.3.

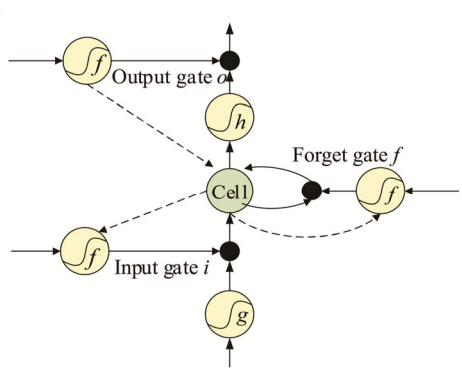

Fig.3 Basic LSTM network structure

As shown in this figure, the main components of the LSTM are the input gate i, the output gate o, the forgetting gate f, and the memory cell "Cell." The door can selectively allow information to pass, and the LSTM enables the protection and control of information through the door structure[17]. The calculation method of each structure comprises the following steps:

$$i^t=\delta(W_i x^t+R_i y^{t-1}+p_i c^{t-1}+b_i), \tag{16}$$

$$o^t=\delta(W_o x^t+R_o y^{t-1}+p_o c^t+b_o), \tag{17}$$

$$f^t=\delta(W_f x^t+R_f y^{t-1}+p_f c^{t-1}+b_f). \tag{18}$$

The cell is similar to a conveyor belt and operates throughout a recurrent network to determine whether to keep or discard information. With the storage and modification of state information, the LSTM unit can achieve long-range memory. Finally, the output o^t of the output layer and the current state c^t of the cell determines the entire LSTM's output y^t, which can be expressed as follows:

$$z^t=\tanh(W_z x^t+R_z y^{t-1}+b_z), \tag{19}$$

$$c^t=i^t z^t+f^t c^{t-1}, \tag{20}$$

$$y^t=o^t h(c^t), \tag{21}$$

where $x^t \in R^d$ is the input vector, $y^t \in R^d$ is the output vector containing the battery characteristics of the LSTM at time t, δ and h represent the activation functions sigmoid and tanh, and w_*, R_*, p_* and b_* represent the coefficient matrix and the offset vector, respectively.

In recent years, inspired by the attention mechanism of the human brain, an attention mechanism has been widely applied in neural networks. It enables the neural network to focus on its input (or feature) subset[18]. It first measures the similarity between the query vector "Query" and the key "Key"; it then scales and normalizes "Query" and "Key", before conducting the weighted process on weight and value "Value." The formula for calculating the output of the attention mechanism is as follows:

$$Attention\left(Q,K,V\right) = soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{22}$$

where Q represents the query vector "Queries", K stands for the key "Keys", V stands for the value "Values", and d_k represents the dimension.

The impact of each feature on the result is different. The traditional LSTM network cannot determine which dimension is more relevant to the prediction of the target value. By adding the attention mechanism, the attention weight of each feature can be learned from the sequence, and then the feature can be merged according to the attention weight.

The influence of each characteristic on the decreasing trend of the lithium battery's life is different. To better identify this difference and improve the accuracy of the model, the attention mechanism is introduced. If the current period is t, then model needs to obtain the output

matrix of h_i in the LSTM layer. It can calculate the corresponding attention weight a_i of n_t with the following formula:

$$e_{ti} = f_{att}\left(h_t\right)$$

$$a_i = \frac{\exp\left(e_{ti}\right)}{\sum_{k=1}^{L}\exp\left(e_{tk}\right)+\varepsilon} \quad , \tag{23}$$

where f_{att} is this paper's tanh function and L is the size of n_t. By adding the attention layer, the degree of influence of each dimension in the hidden layer on the result can be determined. Moreover, features with larger weight coefficients have a greater impact on the results, focusing the network on changes in one or several dimensions.

The LSTM prediction algorithm, based on an attention mechanism, consists of the following steps.

Step 1: After preprocessing the data set T_a and dividing it into a training set and test set, N is the number of samples:

$$T_{tr}=(x_1, x_2, x_3, x_4,...x_n), \tag{24}$$

$$T_{test}=(x_{m+1}, x_{m+2}, x_{m+3}, x_{m+4},...x_n),$$

$$1\le m\le n,\ m,\ n\in N. \tag{25}$$

Step 2: To adapt to the characteristics of the hidden layer input, a short-time input sequence is constructed, and the length of the time series is determined by a fixed step size l. The input of the network model is

$$x_l=\{x'_l, x'_{l+1}, x'_{l+2}, x'_{l+3},...x'_m\},\quad 1\le l\le m-l,\quad l\in N. \tag{26}$$

Step 3: After x_l enters the network, the output of the hidden layer is:

$$P=\{P_1, P_2, P_3, P_4,...P_{m-l}\}, \tag{27}$$

$$P_l=LSTM\ (x'_l,\ c^{<t-1>},\ h^{<t-1>}), \tag{28}$$

where $c^{<t-1>}$ and $h^{<t-1>}$ are the cell status and hidden layer status of the previous moment, respectively.

Step 4: The attention mechanism is added to adjust the weight of the feature:

$$W_l=Soft\max(P_l)*P_l\ . \tag{29}$$

Step 5: The predicted value is output:

$$p=\{p_1, p_2, p_3,...p_m\}. \tag{30}$$

The diagram of the improved network structure is shown in Fig.4.

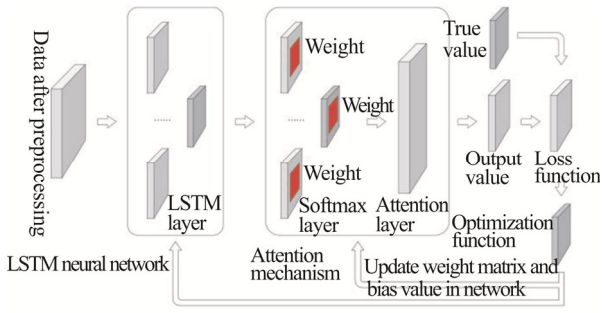

Fig.4 Improved network structure diagram

The root-mean-square error (*RMSE*) is applied as the loss function, and the calculation method is:

$$RMSE=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\left(\ \tilde{Y}_i-Y_i\right)\right)^2}\ , \tag{31}$$

where $\tilde{Y}_i$ is the predicted value of the i-th sample and Y_i is the true value of the i-th sample. The Adam optimization function is applied for the network optimization.

The data sets used were those of lithium-ion batteries #5, #6, and #7 of NASA's Prognostic Center of Excellence laboratory, and the experimental environment was 25 °C. Data in three modes of charge-discharge and electrochemical impedance measurement were recorded.

The charging process was carried out in 1.5 A constant current mode until the battery voltage reached 4.2 V. The battery was charged continuously in constant voltage mode until the charging current was reduced to 20 mA. The discharge process was conducted at the constant current level of 2 A until the voltage decreased to 2.7 V (#5), 2.5 V (#6), and 2.2 V (#7). In this experiment, #5 and #6 were taken as training sets, and #7 was taken as the test set.

The repeated charge-discharge cycle is the main cause of battery aging. With battery #6 as an example, the relationship between the battery's actual capacity and charge-discharge cycle is shown in Fig.5. The battery capacity degradation curve fluctuates from high to low because the battery capacity is directly extracted from the battery data set, which is closely related to the battery data set. The deeper reason lies in the capacity recovery effect of the battery, which is also known as "self-charging" or "self-healing". It refers to the phenomenon that the internal ion concentration of the battery tends to be balanced due to the diffusion effect when the battery is in the shelving state, and the voltage rises accordingly, and the battery life is also prolonged. NASA holds that when the actual capacity of a lithium battery drops to 70% of its rated capacity, the battery has failed and can no longer be effectively powered. The number of charge-discharge cycles between the current capacity and the failure threshold is the remaining useful life of the battery.

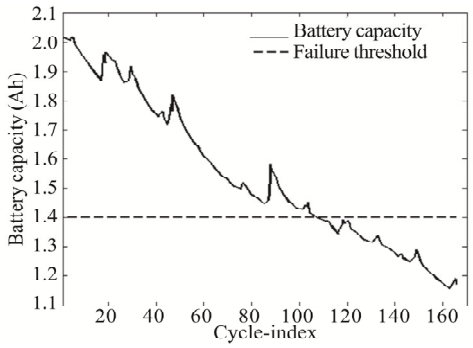

Fig.5 The degradation curve of the battery capacity

After extracting 21 dimensional features, features were input into the AE network to reduce the number of dimensions. The maximum number of experimental iterations

was set to 200. The accuracy of each iteration is shown in Fig.6. As can be seen from the figure, the accuracy rate exhibits a large fluctuation before the 25th iteration, after which it gradually increases. The reason for the small peak in the graph is that, at the beginning of the iteration, network used the local optimal solution, but it eventually used the global optimal solution. After 200 iterations, the accuracy of the AE network reached 99.74%.

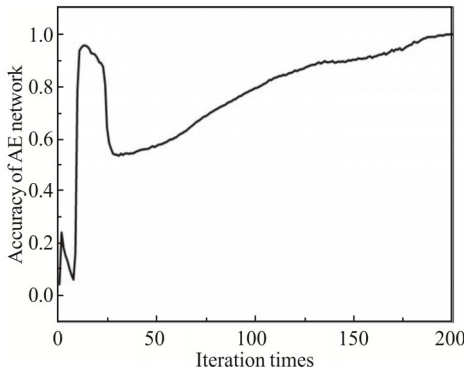

Fig.6 The reduced-dimension network's accuracy

To verify the effectiveness of the dimension-reduction network, we compare the model's training time before and after the dimension reduction. The experimental prediction model uses linear regression, support-vector machines (SVM), adaptive LSTM (ALSTM) with no attention mechanism, and ALSTM with an attention mechanism. Each training time is shown in Tab.1, from which it can be seen that the AE network significantly reduces the training time of the model.

Tab.1 Comparison of the model's training time before and after dimension reduction (s)

Predictive model	Before dimension reduction	After dimension reduction
Linear regression	0.012 0	0.004 5
Support-vector machines	47.191	24.814
ALSTM (No attention mechanism)	14 min 44.447 s	8 min 46.847 s
ALSTM	15 min 37.594 s	9 min 26.701 s

The prediction experiment is divided into two stages: training and testing. In the training stage, the data set comes from the data of batteries #5 and #6. The maximum number of iterations is set to 1 200, and the loss curve of the loop in the iterative process is shown in Fig.7. As can be seen from the figure, after more than 400 iterations, the loss is reduced to a steady value.

To increase the contrast of the experiment, the linear regression and the SVM algorithm were used to predict the outcome of the experiment. The experiment was then conducted with the model of the fusion AE and the LSTM. Finally, the experiment was conducted after the attention mechanism was introduced into the ALSTM model. The results of the experiments were then compared.

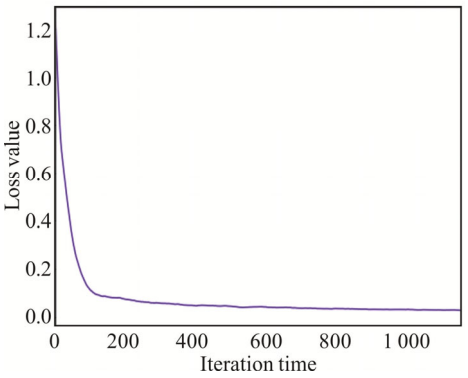

Fig.7 The loss map

The results of the linear regression, SVM, and the ALSTM (no attention mechanism) experiments are shown in Fig.8. Except for the true-value curve, the prediction curves exhibit jitter, which is a natural prediction phenomenon in the model. Furthermore, before 52 cycles of charge and discharge, the fitting of the three methods is similar. However, the amplitude of the linear regression method is larger than those of the other two. As the number of charge-discharge cycles increases, the prediction curves of the linear regression and the SVM algorithm begin to deviate gradually from the true value. The deviations become larger, and the prediction results are poor. In contrast, the fitting of the ALSTM (no attention mechanism) algorithm proposed in this paper is superior in the early and middle period. After the number of charge-discharge cycles reaches 93, however, the curve begins to deviate gradually from the true value.

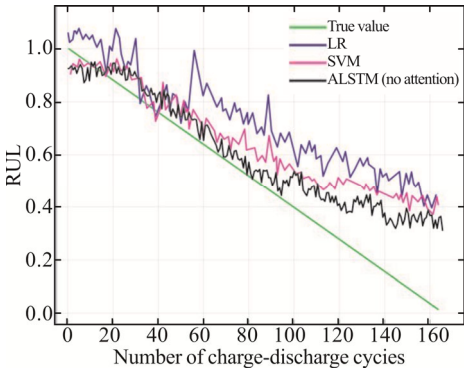

Fig.8 Comparison chart for accuracy of predictive model

The prediction results of the ALSTM model with and without the attention mechanism are shown in Fig.9. After the addition of the attention mechanism, the fitting in the first and middle period clearly improved. The number of charge-discharge cycles reached 120 before any deviation occurred, which was a significant improvement on the

ALSTM model without the attention mechanism. The attention mechanism increased the accuracy of the ALSTM model by 4.1%, reaching 97.22%. This result indicates that the attention mechanism can effectively improve the accuracy of the prediction.

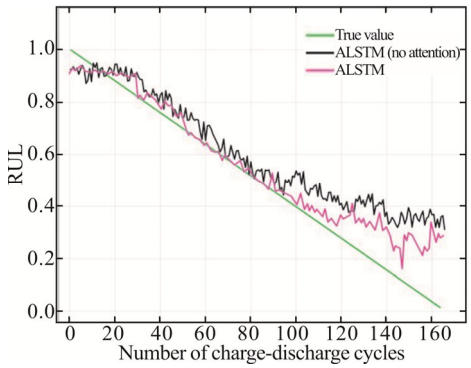

Fig.9 ALSTM model's prediction accuracy before and after the introduction of the attention mechanism

The prediction accuracy of each model is shown in Tab.2.

Tab.2 Prediction accuracy of each RUL model

Predictive model	Root-mean-square error	Accuracy
Linear regression	24.66%	75.34%
Support-vector machines	18.87%	81.13%
ALSTM (No attention mechanism)	5.88%	93.12%
ALSTM	**2.78%**	**97.22%**

It can be seen from the table that the accuracy of the linear regression and SVM models is much lower than the other two models. Moreover, neither of them can meet the prediction accuracy requirement. The accuracy of the prediction in the ALSTM model was higher. Compared with the SVM model, the prediction accuracy was 12.99% higher. With the attention mechanism in the ALSTM model, the accuracy was further improved by 4.1%, which proves the effectiveness of the attention mechanism. The experimental results reveal that the ALSTM fusion model algorithm proposed in this paper can predict the remaining service life of lithium-ion batteries more accurately than other methods.

The algorithm in this paper not only is superior to the traditional algorithm, but also has remarkable advantages over more recent algorithms, as shown in Tab.3.

Tab.3 Comparison with the latest prediction algorithm

Prediction algorithm	Accuracy
ELM indirect method[19]	94.20%
EKF[20]	95.78%
ALSTM	**97.22%**

The accuracy of the algorithm proposed in this paper is 3.02% higher than that of the ELM indirect prediction method proposed by Jiang et al[19] and 1.44% higher than that of the extended Kalman filter-based prediction method proposed by Wang et al[20].

This paper proposes a deep-learning model that combines an AE network and an LSTM network. In the model, dimension reduction is conducted on the extracted multi-dimensional key features to realize the fusion of the complex time-domain features, according to the characteristics of the AE network. The LSTM network is then used for prediction and the attention mechanism is introduced in the LSTM network. The experiment showed that the ALSTM model proposed in this paper improves the accuracy of the RUL prediction and that the attention mechanism has a positive effect on the fusion model. Although this paper has greatly improved the traditional algorithm, there is still a large error in the prediction of the later period of battery life, because by this time, the internal structure of the battery has changed significantly compared with the initial structure. How to improve the accuracy at the later stage for the RUL prediction of the lithium-ion battery should be the focus of future work.

References

[1] Jingliang Zhang and Jay Lee, Journal of Power Sources **196**, 6007 (2011).

[2] Wei He, Nicholas Williard, Michael Osterman and Michael Pecht, Journal of Power Sources **196**, 10314 (2011).

[3] Xiaosong Hu, Changfu Zou, Caiping Zhang and Yang Li, IEEE Power Energy Magazine **15**, 20 (2017).

[4] Liu Datong, Zhou Jianbao and Guo Limeng, Journal of Instrumentation **36**, 1 (2015).

[5] Xin Xu and Nan Chen, Reliability Engineering & System Safety **159**, 47 (2017).

[6] Xiaosong Hu, Dongpu Cao and Bo Egardt, IEEE/ASME Transactions on Mechatronics **23**, 161 (2018).

[7] Meru A. Patil, Piyush Tagade, Krishnan S. Hariharan, Subramanya M. Kolake, Taewon Song, Taejung Yeo and Seokgwang Doo, Applied Energy **159**, 285 (2015).

[8] WANG D, MIAO Q and PECHT M, Journal of Power Sources **239**, 253 (2013).

[9] LIU JIE and SAXENA, An Adaptive Recurrent Nm-ion Batteries, Annual Conference of the Prognostics and Health Management Society, 1 (2010).

[10] Rezvani Mohammad, AbuAil Mohamed, Lee Seungchul and Jay Lee, A Comparative Analysis of Techniques for Electric Vehicle Battery Prognostics and Health Management (PHM), Sae Technical Paper, 2011-01-2247.

[11] WU J, ZHANG C and CHEN Z, Applied Energy **173**, 134 (2016).

[12] REN Lei, Sun Yaqiang, Cui Jin and Zhang Lin, Journal of Manufacturing Systems **48**, 71 (2018).

[13] Wu Yuting, Mei Yuan, Dong Shaopeng, Lin Li and Liu Yingqi, Neurocomputing **275**, 167 (2018).

[14] ZOU H, HASTIE T and TIBSHIRANI R, Journal of Computational and Graphical Statistics **15**, 265 (2006).
[15] Md Shamim Reza and Jinwen Ma, ICA and PCA Integrated Feature Extraction for Classification, IEEE 13th International Conference on Signal Processing (ICSP), Chengdu, China, 453 (2016).
[16] BURDA Y, GROSSE R and SALAKHUTDINOV R, Computer Science **4**, 112 (2016).
[17] Chen Geheng, Information Technology **2**, 149 (2018).
[18] CHOI H, CHO K and BENGIO Y, Neurocomputing **284**, 171 (2018).
[19] Jiang Yuanyuan, Liu Zhu, Luo Hui and Wang Hui, Journal of Electronic Measurement and Instrument **30**, 179 (2016).
[20] Wang Haixia and Li Kaiyong, Computer Measurement and Control **27**, 271 (2019).