State of Charge Estimation Model for Lithium-ion Batteries Based on Deep Learning Neural Networks

Song-Bo Zhang, Xiao-Tian Wang, Jie-Sheng Wang*, Xun Liu

Abstract—**As a new generation of high-performance batteries, lithium-ion batteries have found extensive applications in electric cars, as well as energy storage systems and various other industries. State of charge (SOC) estimation is one of the most important indicators. SOC estimation model of lithium-ion battery based on deep learning neural networks employs diverse external measurement parameters and internal battery parameters as input information, and adopts feed-forward neural network (FNN), convolutional neural network (CNN) and long short-term memory network (LSTM) as predictors to realize the accurate SOC estimation. The model based on deep learning neural networks takes into account the influence of various input parameters and can understand the state of the battery more comprehensively. By using FNN, CNN and LSTM networks, the influence of noise and instability of battery data on SOC estimation can be effectively avoided. After many times of training and verification, the high accuracy and stability of the model can meet the need of SOC estimation for lithium-ion batteries.**

Index Terms—**SOC estimation, feed-forward neural network, convolutional neural network, long short-term memory**

I. Introduction

State of Charge (SOC) estimation technology plays an important role in battery management, which enables us to know the remaining power in the battery more accurately, thus improving the service efficiency and life of the battery [1]. The technology of SOC estimation can be used to determine various battery parameters such as voltage, current, temperature, and other related factors. Additionally, it enables the acquisition of remaining battery power by employing a specific chemical model for the battery [2]. Therefore, SOC estimation technology is very important for improving the efficiency and prolonging the service life of batteries. It is worth noting that different types and manufacturers of batteries have different physical and chemical properties, so it is necessary to study different SOC estimation methods for different types and specifications of batteries [3-4]. In addition, the usage of the battery will also affect the accuracy of SOC estimation, for example, the charging and discharging rate and depth of the battery will have a certain impact on SOC estimation. Early SOC estimation techniques were mainly based on traditional mathematical models, such as RC model, EKF [5] model and intelligent filtering model. These models provide an estimation accuracy by considering the electrochemical characteristics and internal resistance characteristics of the battery from various perspectives during both charging and discharging processes. However, the influence of these models on the battery temperature [6], changing electrochemical characteristics and battery internal resistance in charge and discharge state has not been fully considered, so there is still a great error in predicting battery SOC [7]. In addition, a large number of tests and experimental data of batteries and their characteristic parameters are needed in the process of SOC estimation, which limits their development prospects.

As an important branch of artificial intelligence, deep learning has surpassed the traditional statistical learning methods in prediction and classification tasks [8]. In battery management system, accurate state estimation is very important for battery life and performance. Among them, the state estimation of lithium-ion battery is a challenging task, because lithium-ion battery itself is a nonlinear and time-varying system. Therefore, it is often difficult to adopt traditional modeling and control methods, and deep learning provides a brand-new method, which can accurately model and predict complex systems. At present, deep learning has been widely used to estimate the SOC of lithium-ion batteries, including feed-forward neural network (FNN) [9-10], convolutional neural network (CNN) [11] and long short-term memory neural network (LSTM) [12].

In Ref. [13], a new architecture is proposed that utilizes load classification neural networks to estimate SOC. The method uses data-driven nonlinear models, namely neural networks and learning machines. Firstly, the battery input is preprocessed, and the battery working modes are classified into idle, charging and discharging, and three neural networks are trained simultaneously. Based on the experimental findings, it can be inferred that the machine learning approach employed in this study demonstrates a commendable level of estimation accuracy. Ref. [14]

Manuscript received July 1, 2023; revised October 31, 2023. This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. LJKZ0293), and Postgraduate Education Reform Project of Liaoning Province (Grant No. LNYJG2022137).

Song-Bo Zhang is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: 2092926911@qq.com).

Xiao-Tian Wang is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail:2207942300@qq.com).

Jie-Sheng Wang is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (Corresponding author, phone: 86-0412-2538246; fax: 86-0412-2538244; e-mail: wang_jiesheng@126.com).

Xun Liu is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail:heidengxiaguo@163.com).

proposed an innovative approach to enhance the state estimation technique of model-based Kalman filter (KF) by introducing a novel training method based on extreme learning machine algorithm. Subsequently, various estimation algorithms including extended Kalman filter (EKF), unscented Kalman filter (UKF), adaptive extended Kalman filter (AEKF) and adaptive unscented Kalman filter (AUKF) were employed for SOC estimation. By comparing the experimental results, it can be shown that the application of AUKF algorithm in ELM model can improve the SOC estimation performance of lithium-ion batteries. In Ref. [15], a SOC estimation method based on neural fuzzy system and subtraction clustering is proposed. Simulation experiments are conducted by using an advanced car simulator, comparing it with back-propagation neural network and Elman neural network. In the SOC estimation model, the collected data is used to train and test for 10 different driving cycles. The experimental results show that the model has sufficient accuracy and performs better than the neural network and Elman neural network. A novel approach employing a deep feed-forward neural network (DNN) is utilized for the estimation of battery state of charge (SOC), which is generated by applying driving cycle load to lithium-ion batteries at various ambient temperatures, so that the batteries are exposed to variable dynamics [16]. It is verified on many different data sets, which effectively verifies its effectiveness. This paper presents a SOC estimation model of lithium-ion battery based on deep learning neural network. The structure of the paper is arranged as follows. In the second section, different SOC estimation models of lithium-ion batteries based on deep learning neural network are introduced. The third part includes the experimental simulation and result analysis of different models, and finally draws the conclusion of the paper.

II. SOC Estimation Model Based on Deep Learning Neural Network

A. Feed-forward Neural Network

Feed-forward Neural Network (FNN) is a fundamental model in neural networks, commonly referred to as the Multi-Layer Perceptron (MLP). The model consists of multiple neurons that are connected in a hierarchical structure, including input layers, hidden layers and output layers. FNN comprises of a single-layer feed-forward neural network and a multi-layer feed-forward neural network. Single-layer feed-forward neural networks with only one output layer are the simplest artificial neural networks. In the output layer, each node's value is determined by directly multiplying the input value with its respective weight. In other words, the results of the output layer can be obtained only by a simple linear combination of input values. This neural network is suitable for some simple tasks, such as binary classification or linear regression. See Eq. (1) and Eq. (2), and take out one of its elements for discussion. $x=[x_1,x_2,...,x_n]^T$ is the input feature vector, w_{ji} is the connection right from x_i to y_i , and the output $y_i(j=1,2,...,m)$ is the result of the classification, which is based on different characteristics.

$$s_j = \sum_{i=1}^{n} w_{ji} x_i - \theta_j \qquad (1)$$

$$y_i = f\left(s_j\right) = \begin{cases} 1, s_j \geq 0 \\ 0, s_j < 0 \end{cases} \qquad (2)$$

Multi-layer FNN is composed of three layers: the input layer, one or more hidden layers, and the output layer. In this neural network, the input value is first input through the input layer, then processed by nonlinear activation function through each hidden layer, and finally output to the output layer to get the final result. By using multiple hidden layers, the network can learn more complex features and achieve more complex tasks. Because there are multiple connections between hidden layers, neural network can learn nonlinear mapping, which is the biggest difference from single-layer FNN.

$$s_i^{(q)} = \sum_{j=0}^{n_{q-1}} w_{ij}^{(q)} x_j^{(q-1)}, \left(x_0^{(q-1)} = \theta_i^{(q)}, w_{i0}^{(q-1)} = -1\right) \qquad (3)$$

$$x_i^{(q)} = f\left(s_i^{(q)}\right) = \begin{cases} 1, s_i^{(q)} \geq 0 \\ -1, s_i^{(q)} < 0 \end{cases} \qquad (4)$$

where, $i=1,2,...n_q$; $j=1,2,...,n_{q-1}$; $q=1,2,...,Q$. Each layer represents a single-layer feed-forward neural network (FNN) that forms an $\mathrm{n_{q-1}}$ dimensional hyperplane at the q layer for linear classification of the input patterns at that layer. Through the combination of multiple layers, more complex input pattern classification can be realized eventually. Fig. 1 shows the structure of a multi-layer FNN. FNN have the ability to deal with a large number of nonlinear data and has excellent performance in image processing, natural language processing and other fields.

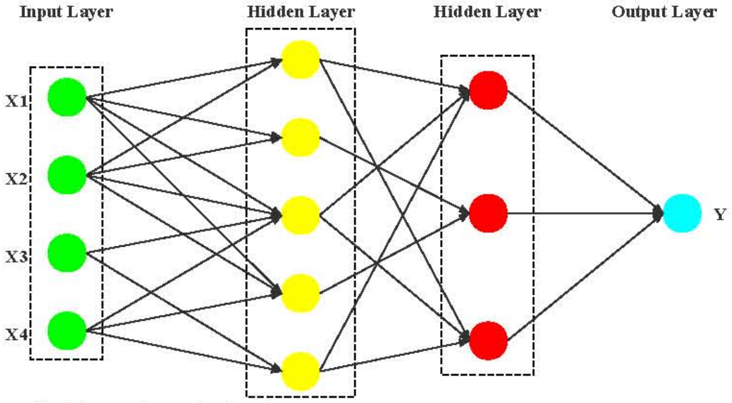


Fig. 1 Structure diagram of multi-layer feed-forward neural network.

The fundamental concept behind training FNN involves feeding a vast amount of familiar data into the neural network and iteratively adjusting its output to align with the real data, thereby enhancing the precision of prediction outcomes. Among them, three core algorithms are Back-Propagation, Gradient Descent and Error Back-Propagation. When the network encounters new data, it can classify or predict the data according to the established relationship. FNN with linear output layer and at least one layer containing nonlinear activation function can realize function approximation with arbitrary accuracy by augmenting the quantity of hidden units. In addition, the neural network can flexibly change the number of input and output neurons as needed to adapt to different joint space variables and task space variables. This improves the flexibility and applicability of the model, so that any position and posture can be modeled as task space variables.

B. Convolutional Neural Network

Convolutional neural network (CNN) is a deep learning technology used in image recognition, object detection, natural language processing and other fields. The primary characteristic of CNN involves the utilization of convolution operation to derive input data features, surpassing the conventional approach of manual feature extraction in terms of efficiency. Convolution operation is an effective way to scan the input image through convolution kernel, extract the feature regions of interest, and aggregate these features into higher- dimensional feature representations. In CNN, each neuron only processes part of the input data, that is, the data in the local receptive field, this effectively decreases the count of trainable parameters and enhances the training process's efficiency. The concrete implementation of CNN usually includes multiple components such as convolution layer, pooling layer and fully connected layer. Alternating convolution layer and pooling layer can continuously extract feature and downsample data. Hence, progressively diminish the size of feature map and minimize the computational intricacy of the model. The fully connected layer is responsible for the classification or regression of the extracted high-level features.

A convolutional neural network consists of several layers including convolution layers, pooling layers and fully connected layers. The core of CNN is convolution layer, and it can extract features from input data. The convolution layer obtains a new feature map by performing element-wise convolution operation on the convolution filter and each region of the input image. This process can be understood as extracting features in the input image, such as edges, textures and so on. At work, the convolution kernel periodically scans the input features and processes them through multiplication and summation of matrix elements and superposition of offsets.

$$\begin{aligned} Z^{l+1}(i,j) &= \left[Z^{l} \otimes \omega^{l+1}\right](i,j)+b \\ &= \sum_{k=1}^{K_l} \sum_{x=1}^{f} \sum_{y=1}^{f} \left[Z_K^{l}(s_0 i+x, s_0 j+y)\omega_k^{l+1}(x,y)+b\right] \end{aligned} \tag{5}$$

$$(i,j) \in \{0,1,\ldots,L_{l+1}\} \tag{6}$$

$$L_{l+1} = \frac{L_1+2p-f}{s_0}+1 \tag{7}$$

In fact, the summation part in Eq. (5), Eq. (6) and Eq. (7) can solve the first cross correlation, and the deviation is $\mathbf{b}$, the convolution input and output of the $l+1$st layer are Z_l and Z_{l+1}, and the size of Z_{l+1} is L_{l+1}. Suppose the dimensions of the feature map are identical in length and width. The pixels of the corresponding feature map are represented by $Z(i,j)$, and the feature map is characterized by a channel number of K. The dimensions of the convolutional kernel, the stride length for convolution, and the count of padding layers are f, s_0 and p, respectively.

The primary purpose of the pool layer is to decrease the input dimension of the subsequent layer and enhance the feature's invariance to position. Maximum pool and average pool are common operations in the process of pooling. The maximum pooled output is the maximum value in each input area, so that the key features of the image can be preserved. The output of average pooling is the average value of each input area, and over-fitting is prevented by reducing the amount of calculation. L_p pooling is a pooling model established at the level of the visual cortex, and its general form is shown in Eq. (8).

$$A_k^{l}(i,j) = \left[\sum_{x=1}^{f} \sum_{y=1}^{f} A_k^{l}(s_0 i+x, s_0 j+y)^{p}\right]^{\frac{1}{p}} \tag{8}$$

where, s_0 and (i,j) have the identical meaning as convolution layer, and p is a per-specified parameter. When the value of p is set to 1, L_p pooling employs average pooling by merging regions with the mean value. Conversely, as p tends towards infinity, L_p pooling adopts maximum pooling by selecting the highest value within the region. L_p pooling has two conceptual extensions, namely mixed pooling and random pooling. In order to make the non-maximum excitation signal enter the next structure, we need to randomly select a value in a random pool according to a specific probability distribution. The linear combination of the average pool and the maximum pool can be used to represent the mixed pool, and Eq. (9) is a concrete expression.

$$A_k^{l} = \lambda L_1\left(A_k^{l}\right)+L_{\infty}\left(A_k^{l}\right), \lambda \in [0,1] \tag{9}$$

The fully connected layer connects the output of the previous layer with each neuron of this layer to realize the classification task. The final layer in a neural network is typically the fully connected layer, which generates an output vector that represents the distribution of probabilities for different categories associated with the image. Loss function can be used to evaluate the prediction effect of a model. The cross-entropy loss function and mean square error loss function are widely used in different loss functions. The cross-entropy loss function is commonly applied in classification tasks to assess model performance by quantifying the difference between predicted and actual classes. The objective of utilizing this optimizer is to progressively minimize the value of the loss function through updating neural network parameters. Common optimizers are Random Gradient Descent (RGD) and Adam. In the training process, the optimizer will calculate the gradient of each parameter according to the value of the loss function, and revert it to every layer of the network, thus updating the value of each parameter. In practical application, CNN suitable for different scenarios can be constructed by adding, modifying and deleting different layers. A simple CNN structure diagram is shown in Fig. 2.

C. Long Short-Term Memory Neural Network

LSTM neural network, known as long short-term memory neural network, is widely utilized in deep learning networks to model and handle sequence data with high efficiency, particularly addressing prolonged dependencies. In conventional RNN models, the primary focus lies in addressing the issue of gradient vanishing or exploding while transmitting information, and has a very good effect when it needs long-term memory or pays attention to certain features. LSTM can be classified as a distinct type of recurrent neural network (RNN). The connection between RNN and LSTM is visually depicted in Fig 3.

The topological structure of LSTM is composed of a series of LSTM units. The input and output of each LSTM unit are similar to those of the traditional recurrent neural network unit. But distinct from the traditional RNN, each LSTM unit has three learnable gates, namely, the Input Gate, the Forget Gate and the Output Gate, which can effectively filter the information that may be irrelevant, thus improving the efficiency and accuracy of the model in processing long sequences. By controlling the opening and closing of the door, LSTM can selectively remember and discard historical information, avoid the gradient problem, and adapt to the needs of different sequence lengths and tasks. In practical application, LSTM is used in many fields, such as image description, inventory prediction, natural language processing, emotion classification, production model and so on. Among them, it is particularly prominent in natural language processing, such as machine translation, text generation and semantic analysis.

LSTM consists of memory cell, input gate, output gate and forgetting gate. An essential component of LSTM is a memory module that retains past data. The input gate has two parts, one is sigmoid function, which is used to control what information is input, and the other is tanh function, which is used to generate new candidate values. Forgetting gate controls the retention and forgetting of old information, and it also has two parts, one is sigmoid function, which is used to control what information is retained, and the other is tanh function, which is used to delete part of the original value. The output gate controls the output range and also has two parts, one is sigmoid function, which is used to control what information is output, and the other is tanh function, which is used to get the output value. The network structure of LSTM is illustrated in Fig. 4 through a schematic diagram.

The specific algorithm in Fig. 4 is shown in Eq. (10) to Eq. (15), where σ stands for sigmoid layer, and each data needs to pass sigmoid function, $\mathrm{W_f}$ stands for weight matrix, $\mathrm{b_f}$ stands for offset, and $[\mathrm{H_{t-1}}, \mathrm{X_t}]$ stands for splicing two matrices together.

$$F_t = \sigma\left(W_f \cdot [H_{t-1}, X_t] + b_f\right) \tag{10}$$

$$I_t = \sigma\left(W_i \cdot [H_{t-1}, X_t] + b_i\right) \tag{11}$$

$$C_t = tanh\left(W_c \cdot [H_{t-1}, X_t] + b_c\right) \tag{12}$$

$$C_t = F_t * C_{t-1} + I_t * C_t \tag{13}$$

$$O_t = \sigma\left(W_o \cdot [H_{t-1}, X_t] + b_o\right) \tag{14}$$

$$H_t = O_t * tanh(C_t) \tag{15}$$

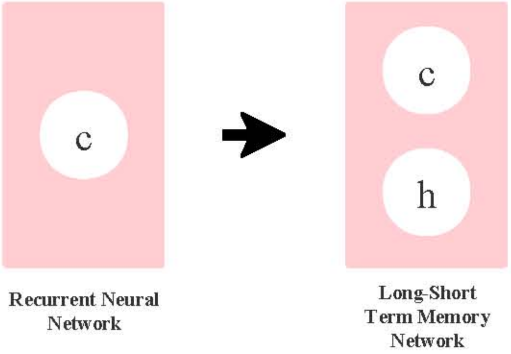


Fig. 3 Relationship between RNN and LSTM.

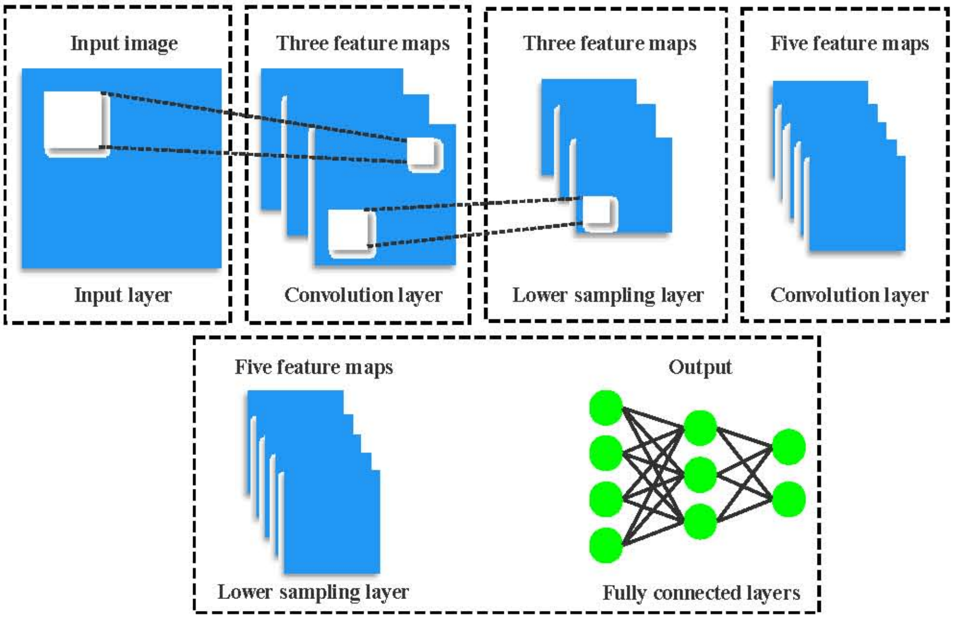


Fig. 2 Structure diagram of convolutional neural network.

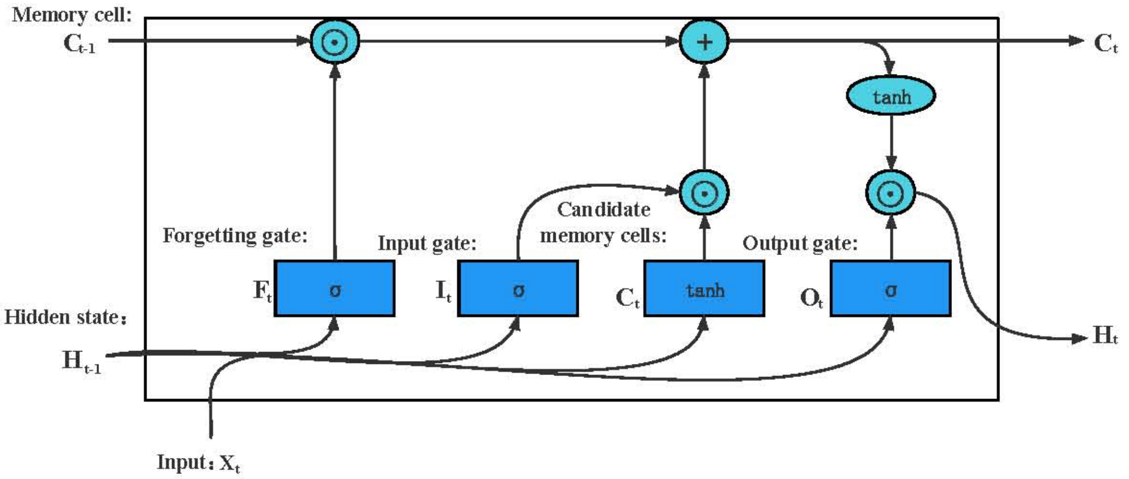


Fig. 4 Structure diagram of LSTM basic network.

III. Simulation Experiments and Result Analysis

A. *Data Selection*

The verification experiment described in this paper uses the test data of Turnigy Graphene 5000mAh 65C lithium-ion battery, which was conducted by Dr. Phillip Kollmeyer of McMaster University in Canada. The test used a 5000mAh Turnigy battery to carry out several groups of experiments, including multiple charge-discharge cycles at different temperatures and four HPPC tests with different discharge and charge rates, and the tests were carried out under various SOCs. Through the identification of the test data, the RC parameters in the battery model in Table I are obtained. Among them, SOC columns from 0 to 1 represent different SOC values of the battery. These parameters include ohmic resistance (R0), electrochemical polarization resistance (R1), concentration polarization resistance (R2), electrochemical polarization resistance capacitance (C1) and concentration polarization capacitance (C2).

B. *Analysis of Simulation Results under -10°C*

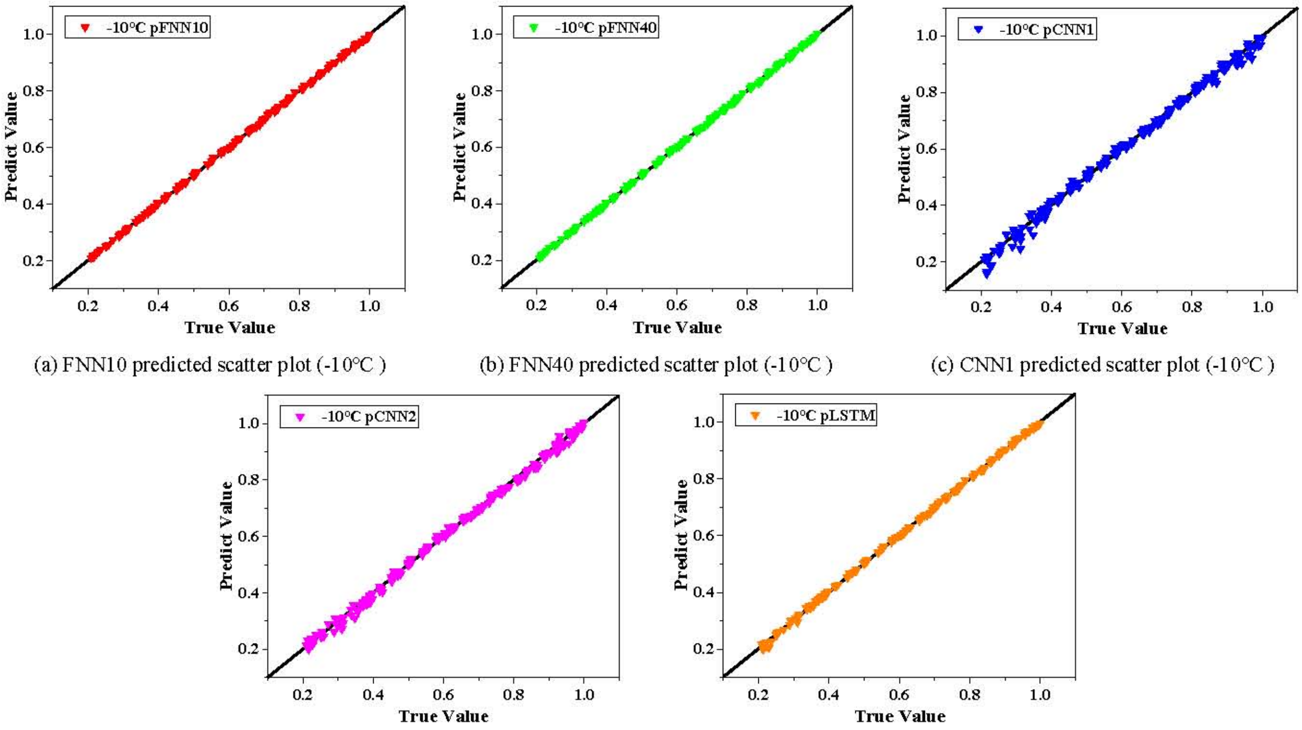
When the temperature is -10°C, a variety of neural network models were used to predict the state estimation of battery remaining power, and the corresponding prediction diagram are drawn. Among them, several classical models, such as feed-forward neural networks (FNN10 and FNN40), convolutional neural networks (CNN1 and CNN2), and long short-term memory neural networks (LSTM), were used to carry out experiments, and their prediction results were analyzed.

Fig. 5 (a) to Fig. 5 (e) are visual demonstrations of model prediction for lithium-ion battery SOC, in which the abscissa represents the true value of lithium-ion battery SOC and the ordinate represents the predicted value obtained by using different models. The straight line y=x in the figure represents that the predicted value coincides precisely with the actual value, while the 225 data points in the figure represent the results obtained by predicting the SOC of lithium-ion batteries through different models. It is evident from Fig. 5 (a) and Fig. 5 (b) that at the temperature of -10°C, the prediction results of FNN10 and FNN40 have very high similarity compared with the real value. Compared with FNN10, FNN40' s prediction result is relatively more stable, and it can accurately reflect the change of prediction value in the whole prediction range. Comparing Fig. 5 (c) and Fig. 5 (d), it's easy to see that the predicted result of CNN1 fluctuates greatly and is not accurate when the SOC range is 0.2 ~ 0.4, but it fluctuates slightly and performs relatively well when the SOC range is 0.4 ~ 1.0. By expanding the number of filters in CNN2, it can be clearly found that the fluctuation of the prediction results is obviously improved when the SOC range is 0.2 ~ 0.4. The predicted results align closely with the true value of y=x when the SOC range is 0.4 ~ 0.6. Based on the findings presented in Fig. 5 (e), it is evident that the LSTM model accurately predicts the SOC of lithium-ion batteries even at a temperature as low as -10°C. However, it should be pointed out that this is only the observation of the data set and the results obtained by the training model, which does not mean that LSTM can show similar high accuracy in all cases. In the practical application process, it is necessary to analyze and adjust the characteristics of the predicted data set more carefully to improve the accuracy of the model prediction.

Then, the prediction curves of five neural network models are compared, namely FNN10, FNN40, CNN1, CNN2 and LSTM. By putting the prediction curves of these models on the same picture and displaying them locally, we can compare the prediction effects of each model more intuitively. Of course, when comparing models, we need to make rigorous revision and evaluation to ensure the credibility of the results.

TABLE I. RC PARAMETERS IN BATTERY MODEL

SOC	R0/mΩ	R1/mΩ	R2/mΩ	C1/F	C2/F
0	1.0001	0.4991	0.4956	20127	201696
0.1	12.8431	0.4979	0.4844	20529	207017
0.3	15.7735	0.4955	0.5034	20165	199004
0.3	10.5689	0.5064	0.4963	19555	202674
0.5	12.8425	0.5048	0.4970	19779	200848
0.6	10.1624	0.5046	0.4988	19592	202566
0.7	10.7265	0.4993	0.5011	20058	198330
0.8	10.1553	0.5057	0.4994	19775	200313
0.9	10.5078	0.4986	0.5004	19945	199863
1	10.4174	0.4977	0.5011	20219	201614



(a) FNN10 predicted scatter plot (-10°C) (b) FNN40 predicted scatter plot (-10°C) (c) CNN1 predicted scatter plot (-10°C)

(d) CNN2 predicted scatter plot (-10°C) (e) LSTM predicted scatter plot (-10°C)

Fig. 5 Predicted scatter plot (-10°C).

The results of specific comparison can be seen in Fig. 6. From the error curves depicted in Fig. 7 (a) to Fig. 7 (e), it can be observed that, overall, the models exhibit errors ranging from -0.08 to 0.04. Specifically, with the increase of the number of hidden neural units in FNN model, the jitter amplitude of FNN40 error curve decreases. For convolutional neural networks, after the number of filters increases, the stability of the error curve of CNN2 is improved, while CNN1 still has significant jitter. The average error of LSTM is between -0.03 and 0.01, and it shows relatively good stability, but it is slightly worse than FNN and more stable than CNN.

After calculating the evaluation indexes of each model (see Table II for evaluation indexes), we make a comprehensive analysis by combining the forecast scatter diagram, forecast curve diagram and local enlarged diagram of forecast curve of each estimation model. At -10°C, among the five estimation models, FNN40 shows the most accurate prediction result, followed by FNN10, LSTM, CNN2 and CNN1. By observing the goodness of fit of different models, we can conclude that at -10°C, FNN10, FNN40, CNN1, CNN2 and LSTM can all effectively estimate the SOC of lithium-ion batteries.

C. Analysis of Simulation Results under 0°C

The data in Fig. 8(a) and Fig. 8(b) show that the predicted results of FNN10 and FNN40 are very similar to the true value at 0°C, respectively.

However, in the range of SOC value of 0.2-0.4, the prediction results of FNN10 fluctuate slightly, while FNN40 with hidden neural units hardly fluctuates. In contrast, the forecast result of FNN40 is more stable. By comparing Fig. 8(c) and Fig. 8(d), it can be noted that the accuracy of CNN1's predictions is significantly affected by the SOC range of 0.2-0.5, leading to substantial fluctuations in its results, but it performs well and fluctuates little in other SOC ranges. In contrast, the fluctuation of prediction results of CNN2 after the number of filters is expanded is obviously improved when the SOC is 0.2-0.5. By observing the results in Fig. 8(e), it can be found that the LSTM is stable in predicting the SOC of lithium-ion batteries, with little fluctuation, and can still maintain high accuracy even at 0°C.

The prediction curves of five neural network models are presented in Fig. 9. By observing the error curves depicted in Fig. 10(a) to Fig. 10(e), it can be observed that all models exhibit errors ranging from -0.08 to 0.09. Judging from the jitter amplitude of the error curve, FNN40 is more stable than FNN10. When the Simple Point value is in the range of 150-200, there is great jitter in CNN1, and the error curve of CNN2 is more stable after increasing the number of filters. The average error of LSTM is between -0.02 and 0.02, and it shows relatively good stability.

We get the prediction results of five estimation models, and then compare and analyze them, it can be found that FNN40 is the most accurate model at 0°C, followed by FNN10, LSTM, CNN2 and CNN1. At the same time, by observing the evaluation indexes of different models, it can be inferred that the SOC of lithium-ion batteries can be accurately estimated by the aforementioned five models. At a temperature of 0°C, the evaluation criteria for each model are presented in Table III.

TABLE II. EVALUATION INDEX OF EACH MODEL (-10°C)

Model	pFNN10	pFNN40	pCNN1	pCNN2	pLSTM
MAE	0.0022	0.0019	0.2627	0.2625	0.2626
RMSE	0.0029	0.0025	0.3138	0.3138	0.3137
R^2	0.9998	0.9999	0.9954	0.9976	0.9995

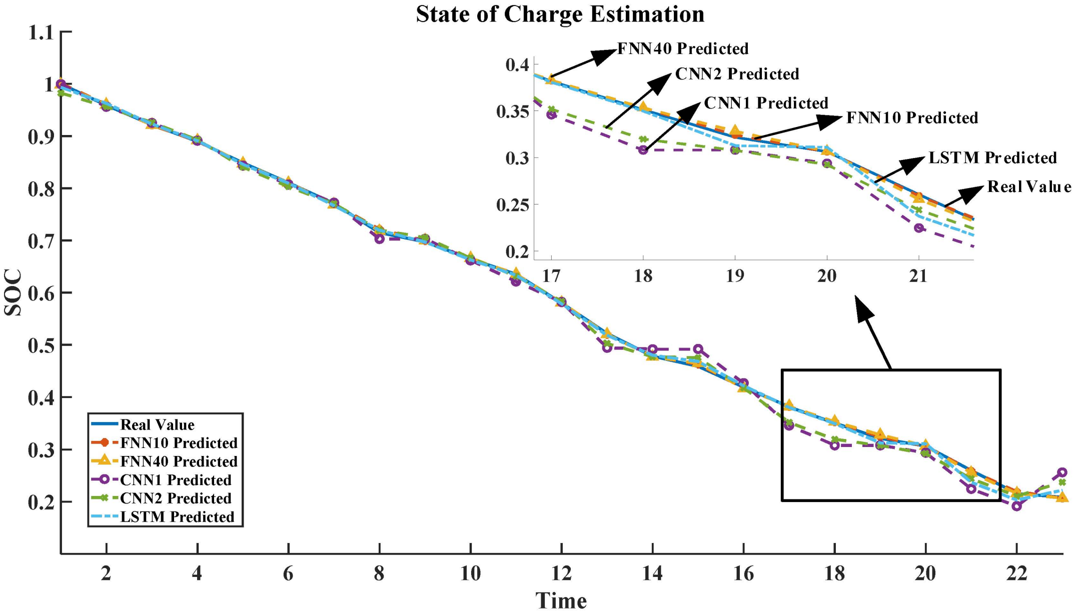
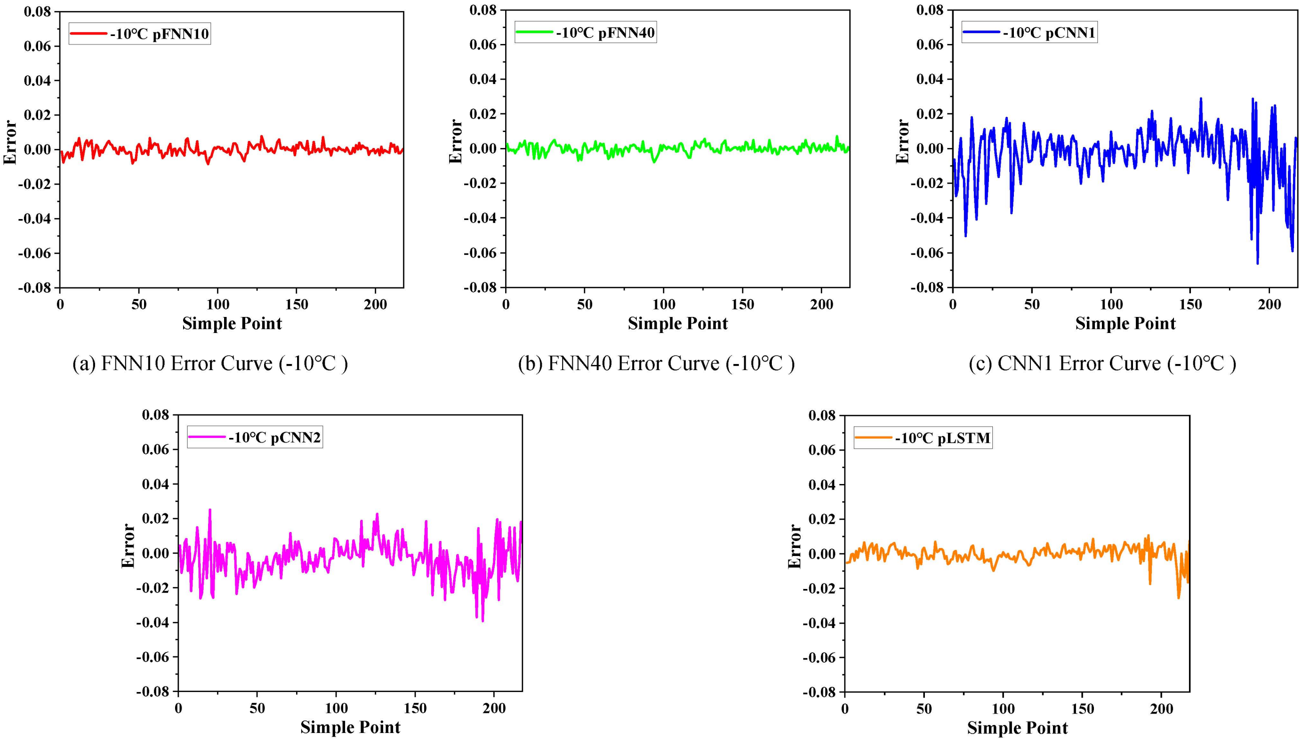


Fig. 6 Prediction Curves and Partial enlarged view (-10°C).



(a) FNN10 Error Curve (-10°C)

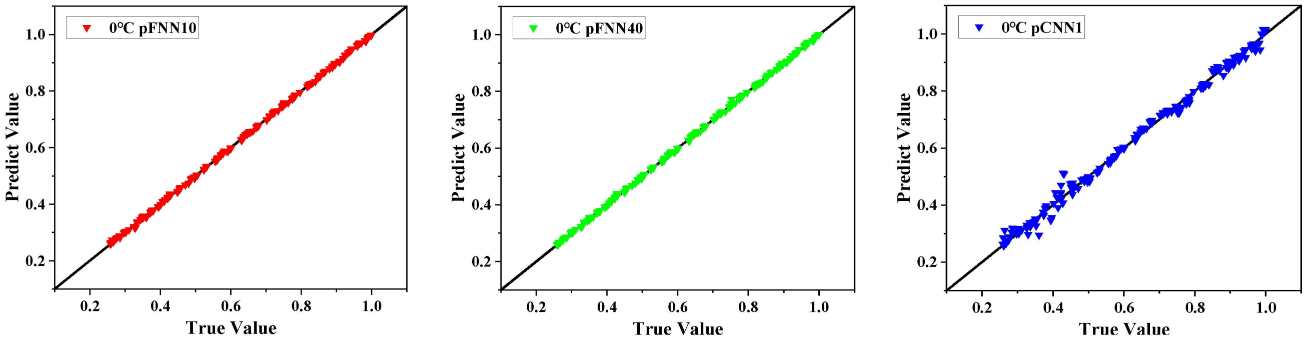
(b) FNN40 Error Curve (-10°C)

(c) CNN1 Error Curve (-10°C)

(d) CNN2 Error Curve (-10°C)

(e) LSTM Error Curve (-10°C)

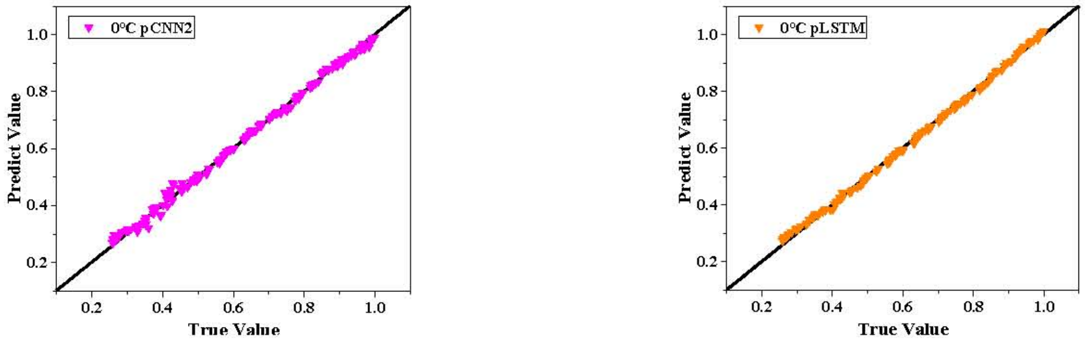
Fig. 7 Error Curve (-10°C).



(a) FNN10 predicted scatter plot (0°C)

(b) FNN40 predicted scatter plot(0°C)

(c) CNN1 predicted scatter plot (0°C)



(d) CNN2 predicted scatter plot (0°C)

(e) LSTM predicted scatter plot (0°C)

Fig. 8 Predicted scatter plot (0°C).

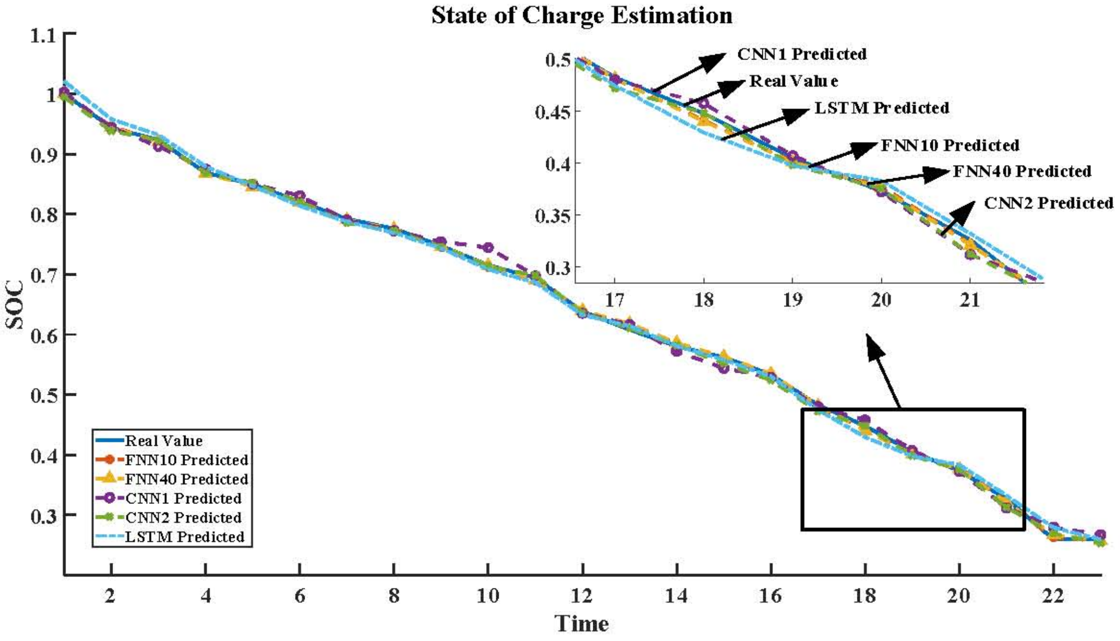
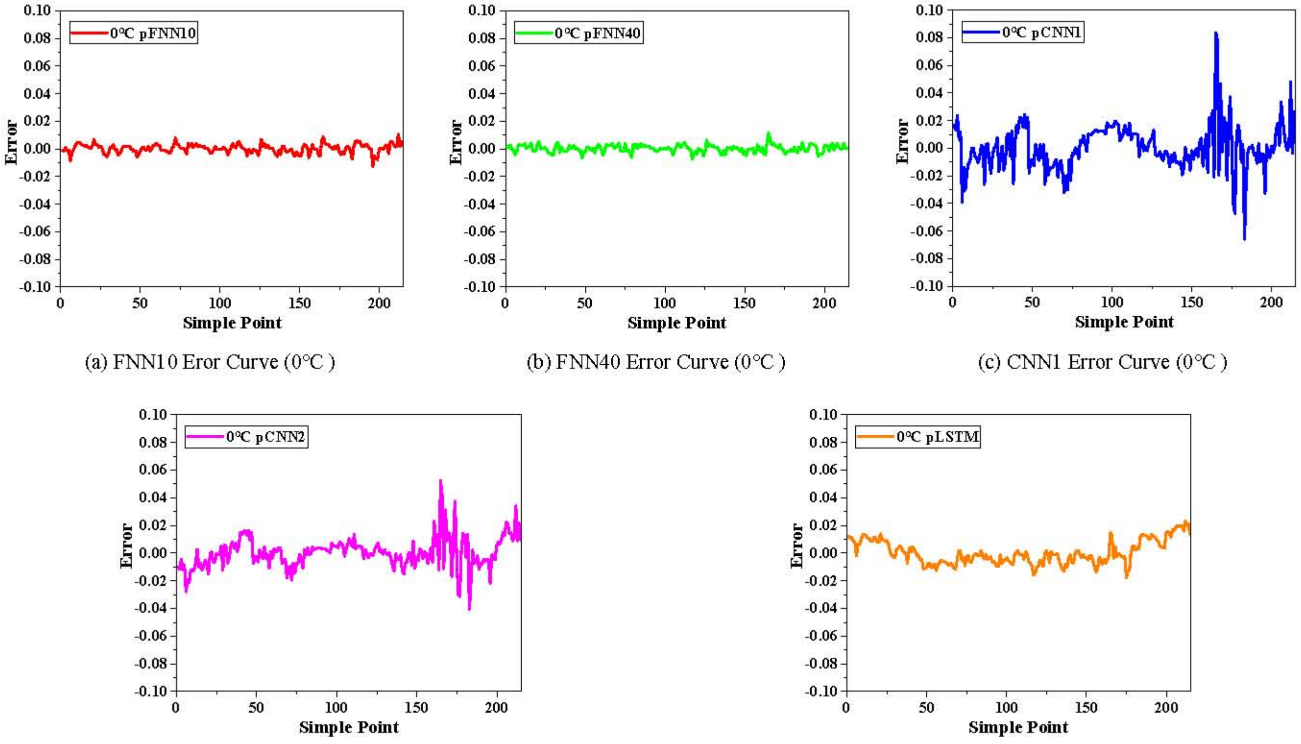


Fig. 9 Prediction Curves and Partial enlarged view (0°C).



(a) FNN10 Eror Curve (0°C)

(b) FNN40 Error Curve (0°C)

(c) CNN1 Error Curve (0°C)

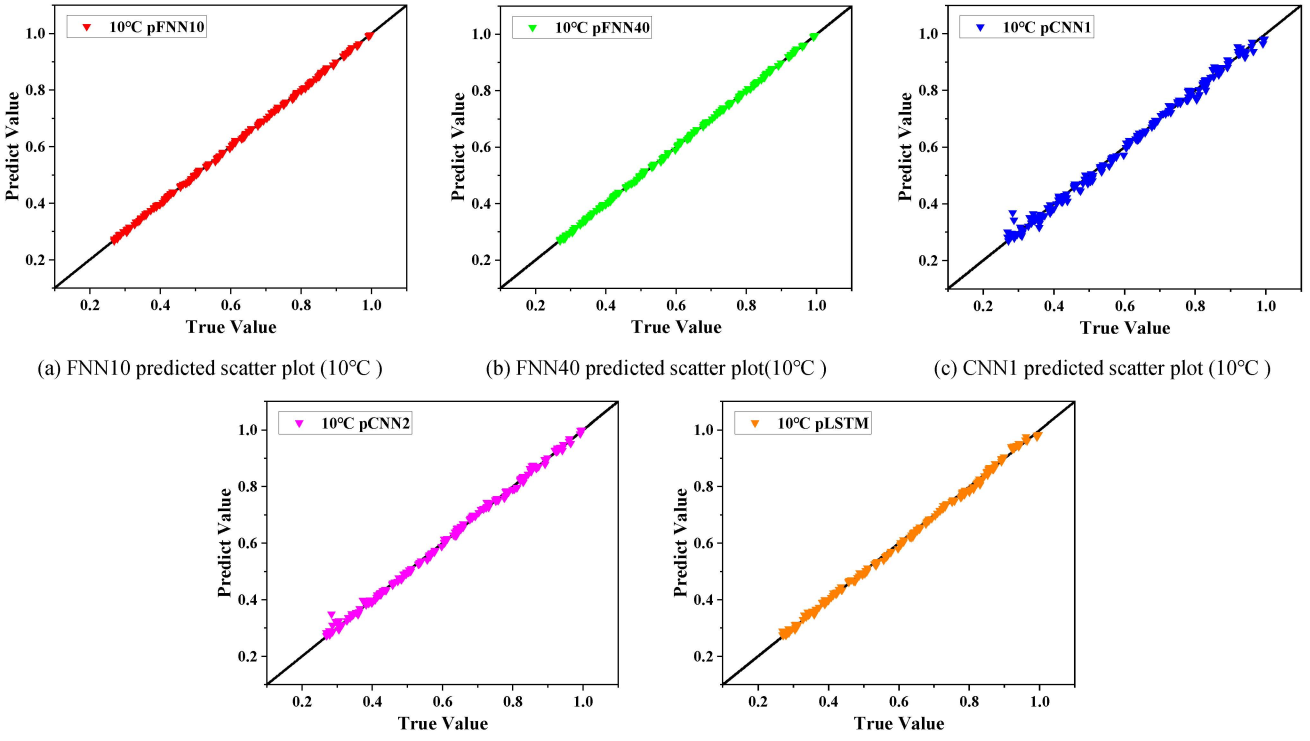
(d) CNN2 Error Curve (0°C)

(e) LSTM Error Curve (0°C)

Fig. 10 Error Curve (0°C).

D. Analysis of Simulation Results under 10℃

By looking at the data of Fig. 11(a) and Fig. 11(b), it can clearly see that the predicted results of FNN10 and FNN40 are very similar to the real values at 10℃. In the forecast scatter chart of FNN10, the fluctuation of forecast results can be almost ignored, while FNN40 is more stable and the forecast results are more dense. At a temperature of 10℃, based on the analysis of Fig. 11(c) and Fig. 11(d), it can be inferred that CNN1 exhibits minimal fluctuations within the SOC range, while the predicted data displays dispersion. After the number of filters is enlarged, the prediction data of CNN2 is more compact, and the whole SOC range is basically consistent with the real value. It can be found from Fig. 13(e), when the ambient temperature is 10℃, the predicted results of LSTM for the SOC of lithium-ion batteries fluctuate within the SOC value range of 0.8~1.0, but they still show high accuracy on the whole. For the evaluation of the estimation model at the ambient temperature of 10℃, the curves and local enlarged graphs of the prediction results of five neural network models, FNN10, FNN40, CNN1, CNN2 and LSTM, are drawn, as shown in Fig. 12. The error curves of each model at 10℃ are shown in Fig. 13(a) to Fig. 13(e), and the errors of all models are between -0.04 and 0.09. With Fig. 13(a) and Fig. 13(b), it can be seen that the error curve of FNN40 has the smallest fluctuation among the five curves, and it is slightly stable compared with FNN10 after adding hidden nerve units. When the value of Simple Point is in the range of 175-225, the fluctuation of CNN1 is large, while the stability of CNN2 is obviously improved. The error of LSTM is between -0.03 and 0.03 shown in Table IV.



(a) FNN10 predicted scatter plot (10℃)

(b) FNN40 predicted scatter plot(10℃)

(c) CNN1 predicted scatter plot (10℃)

(d) CNN2 predicted scatter plot (10℃)

(e) LSTM predicted scatter plot (10℃)

Fig. 11 Predicted scatter plot (10℃).

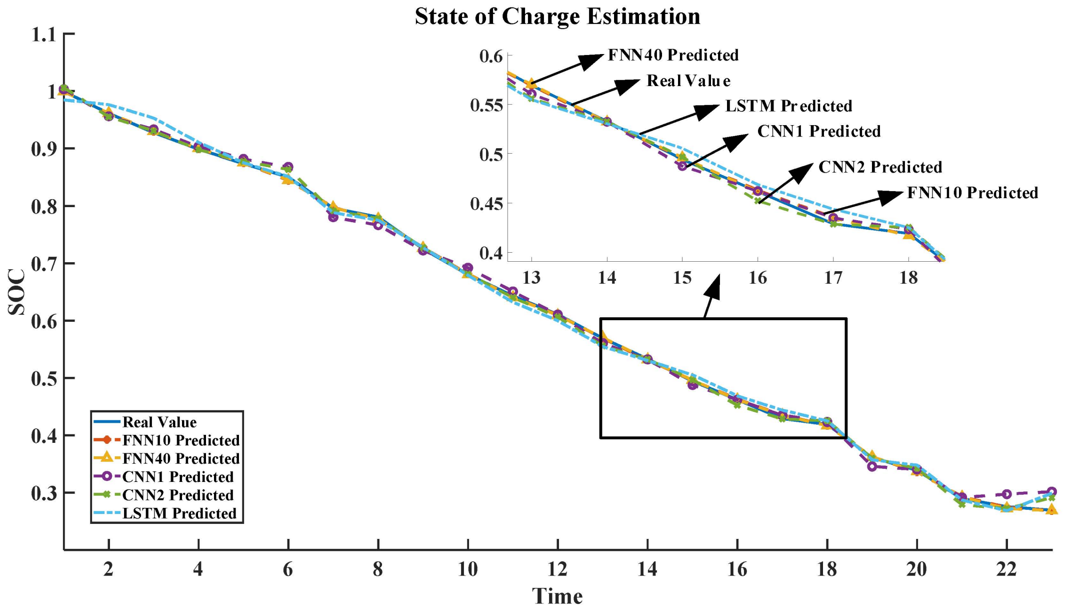
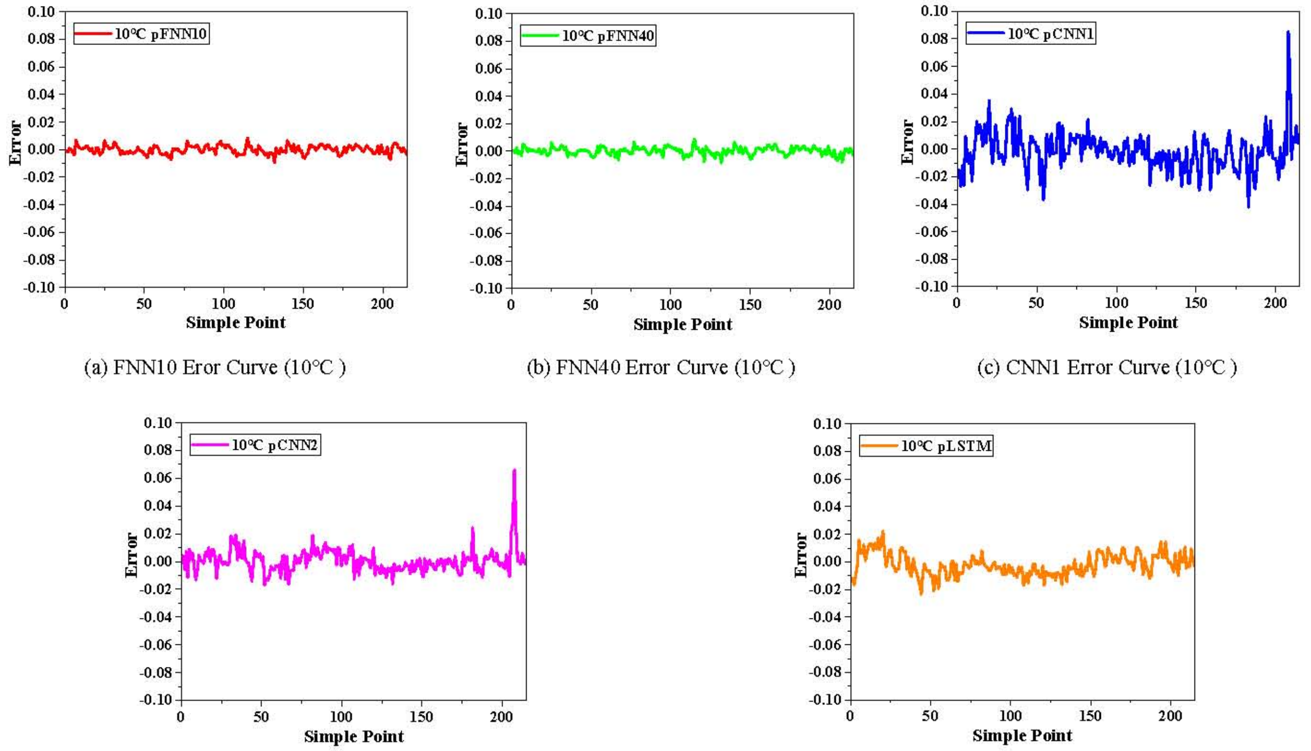


Fig. 12 Prediction Curve and Partial enlarged view (10℃).



(a) FNN10 Eror Curve (10°C)

(b) FNN40 Error Curve (10°C)

(c) CNN1 Error Curve (10°C)

(d) CNN2 Error Curve (10°C)

(e) LSTM Error Curve (10°C)

Fig. 13 Error Curve (10°C).

TABLE III. EVALUATION INDEX OF EACH MODEL (0°C)

Model	pFNN10	pFNN40	pCNN1	pCNN2	pLSTM
MAE	0.0024	0.0021	0.2590	0.2584	0.2590
RMSE	0.0032	0.0029	0.3101	0.3095	0.3098
R^2	0.9998	0.9998	0.9945	0.9976	0.9986

TABLE IV. EVALUATION INDEX OF EACH MODEL (10°C)

Model	pFNN10	pFNN40	pCNN1	pCNN2	pLSTM
MAE	0.0024	0.0022	0.2461	0.2461	0.2455
RMSE	0.0029	0.0028	0.2953	0.2951	0.2940
R^2	0.9998	0.9998	0.9942	0.9983	0.9985

IV. CONCLUSION

Based on the experimental data of Turnigy Graphene 5000mAh 65C lithium-ion battery at different temperatures, this paper uses feed-forward neural networks (FNN10 and FNN40), convolutional neural networks (CNN1 and CNN2) and long short-term memory neural networks (LSTM) to build a prediction model of lithium-ion battery SOC, and concludes that these five neural networks can effectively predict them. At the temperatures of -10°C, 0°C and 10°C, five models all showed high accuracy and precision. In the experiments at different ambient temperatures, FNN, CNN and LSTM models are relatively stable, and LSTM model can better capture the information of time series data. Comprehensive experimental results show that FNN, CNN and LSTM neural networks have broad application prospects in SOC prediction of lithium-ion batteries. For practical application, a suitable prediction model can be chosen according to the scene, so as to improve the energy utilization rate and system stability of lithium-ion batteries and reduce security risks.

REFERENCES

[1] W. Waag, C. Fleischer, and D. U. Sauer, “Critical Review of The Methods for Monitoring of Lithium-ion Batteries in Electric and Hybrid Vehicles”, *J. Power Sources*, vol. 258, pp. 321-339, 2014.

[2] R. Ahmed, M. El Sayed, I. Arasaratnam, J. Tjong, and S. Habibi, “Reduced-order Electrochemical Model Parameters Identification and SOC Estimation for Healthy and Aged Li-ion Batteries. Part I: Parameterization Model Development for Healthy Batteryies”, *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 3, pp. 659-677, 2014.

[3] X. Hu, and F. Sun, “Fuzzy Clustering Based Multi-model Support Vector Regression State of Charge Estimator for Lithium-ion Battery of Electric Vehicle”, *Intelligent Human Machine Systems and Cybernetics, IEEE Computer Society*, pp. 392-396, 2009.

[4] X. Hu, S. E. Li, and Y. Yang, “Advanced Machine Learning Approach for Lithium-ion Battery State Estimation in Electric Vehicles”, *IEEE Transactions on Transportation Electrification*, vol. 2, no.2, pp. 140-149, 2016.

[5] I. B. Ciocoiu, “RBF Networks Training Using a Dual Extended Kalman Filter,” *Neurocomputing*, vol. 48, no. 1-4, pp. 609-622, 2002.

[6] Hansen T, and C. J. Wang, “Support Vector Based Battery State of Charge Estimator”, *Journal of Power Sources,* vol. 141, no. 2, pp. 351-358, 2005.

[7] R. Giglioli, P. Pelacchi, M. Raugi, and G. Zini, “A State of Charge Observer for Lead-acid Batteries”, *Energia Elettrica*, vol. 65, no. 1, pp. 27-33, 1998.

[8] M. Charkhgard, and M. Farrokhi, “State-of-Charge Estimation for Lithium-ion Batteries Using Neural Networks and EKF”, *IEEE Transactions on Industrial Electronics*, vol. 57, no. 12, pp. 178-4187, 2010.

[9] J. R. Zhang, J. Zhang, T. M. Lok, and M. R. Lyu, “A Hybrid Particle Swarm Optimization-back-propagation Algorithm for Feedforward Neural Network Training”, *Applied Mathematics & Computation,* vol. 185, no. 2, pp. 1026-1037, 2007.

[10] J. Hu, J. Zhang, C. Zhang, and J. Wang, “A New Deep Neural Network Based on A Stack of Single-hidden-layer Feedforward Neural Networks with Randomly Fixed Hidden Neurons”, *Neurocomputing*, vol. 171, pp. 63-72, 2016.

[11] A. Gurunathan, and B. Krishnan, “Detection and Diagnosis of Brain Tumors Using Deep Learning Convolutional Neural Networks”, *International Journal of Imaging Systems and Technology*, vol. 31, no. 3, pp. 1174-1184, 2021.

[12] D. T. Anh, A. H. Tanim, D. P. Kushwaha, Q. B. Pham, and V. H. Bui, “Deep Learning Long Short-term Memory Combined with

Discrete Element Method for Porosity Prediction in Gravel-bed Rivers", *International Journal of Sediment Research*, vol. 38, no. 1, pp. 128-140, 2023.

[13] S. Tong, J. H. Lacap, and J. W. Park, "Battery State of Charge Estimation Using A Load-classifying Neural Network", *Journal of Energy Storage*, vol. 7, pp. 236-243, 2016.

[14] J. Du, Z. Liu, and Y. Wang, "State of Charge Estimation for Li-ion Battery Based on Model from Extreme Learning Machine", *Control Engineering Practice*, vol. 7, pp. 11-19, 2014.

[15] T. Zahid, K. Xu, W. Li, C. Li, and H. Li, "State of Charge Estimation for Electric Vehicle Power Battery Using Advanced Machine Learning Algorithm under Diversified Drive Cycles", *Energy*, vol. 162, no. 1, pp. 871-882, 2018.

[16] C. Ephrem, P. J. Kollmeyer, P. Matthias, and E. Ali, "State-of-Charge Estimation of Li-ion Batteries Using Deep Neural Networks: a Machine Learning Approach", *Journal of Power Sources*, vol. 400, pp. 242-255, 2018.