International Core Journal of Engineering Volume 5 Issue 10, 2019
ISSN: 2414-1895 DOI: 10.6919/ICJE.201909_5(10).0008

LOESS Optimized SVR Used to Estimate the Battery State of Charge

Yicheng Zhou [a], Yi Guo [b], Yijian Liu [c], Yibo Zhao [d]

School of Shanghai Maritime University, Shanghai 201306, China

[a]fyzx_135@126.com, [b]yiguo@shmtu.edu.cn, [c]yijian@shmtu.edu.cn, [d]yibo@shmtu.edu.cn

Abstract

This paper combines Locally Estimated Scatterplot Smoothing (LOESS) and Support Vector Regression (SVR) to estimate the State of Charge (SOC) of lithium-ion batteries. The experimental data was obtained from ADVISOR, and the UDDS conditions in which the virtual car was running in the city were simulated. Over the entire test set, the Root Mean Square Error (RMSE) of the final model is 0.776%, which is well below the 5% requirement for practical applications. This indicates that the accuracy of the model meets the requirements for the estimation of the state of charge of the power battery.

Keywords

Electric vehicle; state-of-charge(SOC); SVM.

1. Introduction

The state of charge (SOC) of lithium-ion batteries is an important parameter in the battery management system (BMS) [1]. By accurately estimating the SOC value, BMS can fully utilize the capacity and performance of the lithium-ion battery to prevent overcharging and discharging of the battery, effectively extending the service life of the battery and reducing the cost of the vehicle[2-3]. Battery SOC estimation current challenges are: ①You can not directly and accurately measure the SOC of lithium-ion batteries, you can only be indirectly estimated and predicted by current, voltage and other electrical parameters, and these indirect parameters have a nonlinear relationship with SOC; ②The actual operating conditions are complex, and the battery SOC is also affected by the use conditions, battery aging, load changes, etc.[4-5]; ③Different applications require lithium-ion battery series and parallel to increase power, however, each battery has its difference in discharge rate and leakage coefficient leads to inconsistency in the SOC of each battery after use; ④BMS storage space and computing resources are also quite limited. These all make the SOC estimation of lithium ion batteries become a tough problem.

The SOC estimation algorithms of battery commonly used can be divided into the following three categories[6]: ①Estimation method combining Open Circuit Voltage method (OCV) with ampere-time integration method[7]. By measuring the OCV of the battery, the initial SOC of the battery is obtained, and during use, the amount of electricity flowing out of the battery is integrated and accumulated, thereby obtaining the remaining power in the battery. The advantage is that the algorithm is simple and easy to implement, and the requirements for BMS resources are not high. However, there are also cases where the accuracy of the current sensor is high, the accumulated error cannot be eliminated, and the change in the rated capacity of the battery with the operating conditions and aging cannot be handled. Also it's more suitable for lead-acid batteries; ②based on battery equivalent model estimates. Such as extended Kalman filter (EKF)[7-9], sliding mode observer (SMO)[24] and nonlinear observer (NSO)[26]. These methods use closed-loop estimation algorithm, which has high estimation accuracy and good robustness. However, it also needs to rely on high-

ISSN: 2414-1895 DOI: 10.6919/ICJE.201909_5(10).0008

precision battery model, which is not only difficult to obtain, but also reduces the versatility of the method, and also requires a large amount of BMS computing resources. ③Machine learning based prediction methods, along with the rise of artificial intelligence, have also been added to the application[47]. For example, extreme learning machine, support vector machine (SVM), artificial neural network (ANN)[41] and so on. This is a popular method in recent years, and its estimation accuracy and robustness are better, with strong nonlinear approximation ability[10], and the model versatility can be improved by expanding the sample capacity. However, there are still cases where the model learning time is long and the BMS resources are occupied.¬

In view of the shortcomings of the above three methods, this paper combines LSTM-RNN and LS-SVM to estimate the SOC, which is abbreviated as Neuro-SVM in this paper. The model can accurately measure various indirect information, such as voltage, current and battery temperature, measured by the battery to the state of charge of the battery, and effectively avoids the cumbersome of such as Kalman filter, synovial observer, etc. In the parameter estimation process, the optimal model can be obtained by simply debugging the number of hidden layer neurons, the batch size, the number of iterations, the number of LSTM nuclei, C and gamma during the training process. In this paper, an electric vehicle simulation platform will be built on the electric vehicle simulation software ADVISOR, and the corresponding electrical parameters will be collected to verify the model SOC estimation results.

2. Neuro-SVM Model

2.1 LSTM-RNN

Traditional neural networks, such as BP neural networks, convolutional neural networks (CNN), or artificial neural networks (ANN), have an implicit premise that inputs and outputs are independent [11-12]. However, this is often not the case: the SOC of a lithium iron phosphate battery at a certain moment is closely related to the previous state. One of the advantages of Recurrent Neural Networks (RNN) is that they are able to pass previous information to the current moment. So instead of just mapping the input to the output, the RNN contains a mapping function loop of the input from the previous time step to the output. Its standard model is shown in the figure:

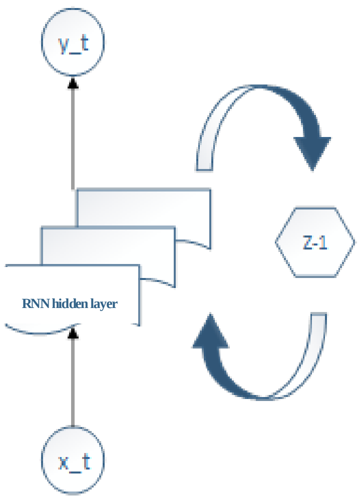

Figure 1. RNN basic structure

Figure 1 shows the basic RNN architecture with delays. In this configuration, the input vectors are fed into the RNN one at a time at a sampling interval t. This structure can take advantage of all of the previously available input information, rather than just using a fixed number of input vectors.

Although RNN can effectively process nonlinear time series, as time goes on, RNN may cause a series of problems, such as gradient disappearance and gradient explosion, which leads to the loss of

ISSN: 2414-1895 DOI: 10.6919/ICJE.201909_5(10).0008

effective information or the introduction of invalid information. The result of the estimation that caused the error. The LSTM circulating neural network used in this paper replaces the traditional RNN nucleus with the LSTM nucleus, which has long-term memory ability [18], which more successfully solves the problem of predicting and estimating the long-term large amount of data. The LSTM cell nucleus is shown in Fig. 2.

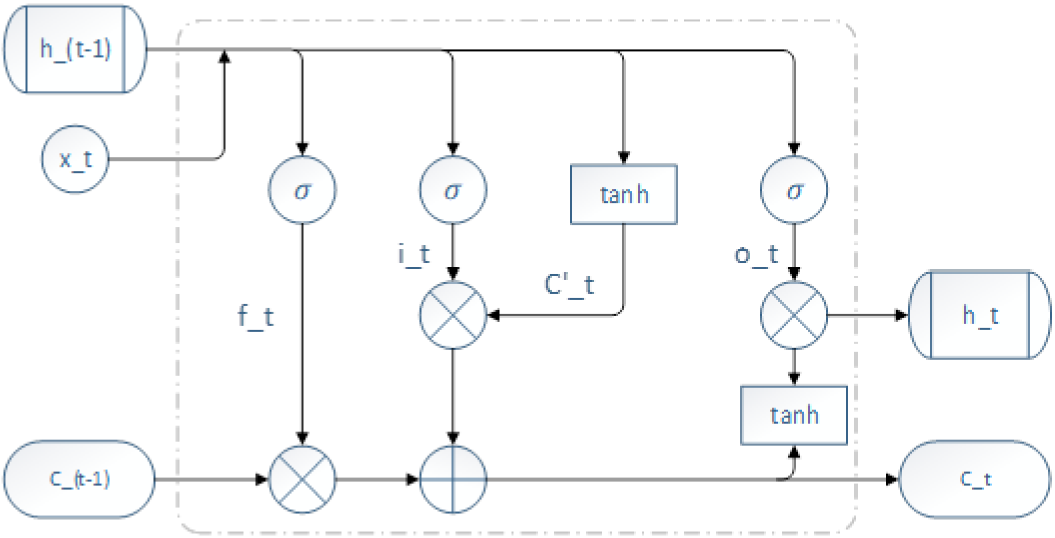

Figure 2. LSTM nuclei

The LSTM-RNN model models nonlinear dynamic systems by mapping input sequences to output sequences. First, the input vector of the model is x(t)=[V(t),I(t),T(t)] , where V(t), I(t), T(t) are the voltage and current of the battery, respectively. And the temperature are respectively packed into small units at the sampling interval t, and the state h(t-1) and the state vector x_t of the previous sample are hidden by the LSTM unit. This input is passed through the tanh activation function; the second step passes the combined input through the input gate, which is a sigmoid function layer that controls the input stream by making the input gate weight close to 0 or 1. The next step in the data flow into the cell is to discard the unnecessary input by t0 and the forget gate of the internal state. The input data lag s(t-1) is added to the recursive function. Finally, the output state is revoked by the tanh function, which is controlled by the output gate. In the LSTM memory unit, the input, output, and forget gates allow the LSTM to forget or write new information to the memory unit.

The mathematical equation for the storage unit is as follows:

$$\mathrm{f_t} = \sigma(\mathrm{w}_{xf}\mathrm{x_t} + \mathrm{w_{hf}}\mathrm{h_{t-1}} + \mathrm{b}_f) \tag{1}$$

$$\mathrm{i}_t = \sigma(\mathrm{w}_{xi}\mathrm{x}_t + \mathrm{w_{hi}}\mathrm{h_{t-1}} + \mathrm{b_i}) \tag{2}$$

$$\mathrm{C'_t} = \tanh\,(w_{xc}x_t + w_{hc}h_{t-1} + b_c) \tag{3}$$

$$C_t = \mathrm{f}_t \cdot C_{t-1} + i_t \cdot \mathrm{C'_t} \tag{4}$$

$$\mathrm{o}_t = \sigma(\mathrm{w_{xo}}x_t + w_{ho}h_{t-1} + b_o) \tag{5}$$

$$\mathrm{h}_t = o_t \tanh\,(C_t) \tag{6}$$

Where o, i, f and c are the output, input, forget the gate and memory unit of the network. Although there is no droupout to train the model without affecting the hidden layer, when testing the neural network, you need to turn off the droupout, and all hidden neuron output will affect the testing process. In this paper, a droupout is used during training of the LSTM network model.

ISSN: 2414-1895 DOI: 10.6919/ICJE.201909_5(10).0008

2.2 SVM

The concept of Support Vector Machine (SVM) was originally based on the principle of structural risk minimization [19]. It uses the values of different classes as the training set, and the training model is used to obtain the hyperplane, which is the least misjudged. In case you can distinguish between different classes. The goal of this classifier is to maximize the distance between the hyperplane and the nearest data point of each class, the support vector [12].

Suppose the training set with input data $x_i \in R^n$ is $S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_l, y_l)\}$, and the corresponding two-class label is $y_i \in \{-1, +1\}$. According to Vapnik's original formula, the SVM classifier is [24]:

$$D(x) = w^T \Psi(x) + b \tag{7}$$

Where $\Psi(x)$ is a nonlinear function that maps the input space x to a feature space whose possible dimension is infinite; the coefficient w^T is an m-dimensional weight vector; the threshold b is a scalar [17].

$$y_i(w^T \Psi(x_i) + b) \geq 1, i = 1, \cdots, l \tag{8}$$

In order to maximize the interval between the two classes, that is, to obtain the optimal hyperplane [19], the following optimization problems need to be defined:

$$\min_{\boldsymbol{w},\boldsymbol{b}} \boldsymbol{J}(\boldsymbol{w}, \boldsymbol{b}) = \frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{w}^{\boldsymbol{t}} \boldsymbol{w} \tag{9}$$

Introduce the relaxation variable ξ_i to equation (7) as follows:

$$\begin{gathered} y_i(w^T \Psi(x_i) + b) \geq 1 - \xi_i, i = 1, \cdots, l \\ \xi_i \geq 0, i = 1, \cdots, l \end{gathered} \tag{10}$$

Therefore, in order to obtain the optimal hyperplane, it means that the following optimization problems need to be solved:

$$\min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi}} \boldsymbol{J}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}) = \frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{w} + \boldsymbol{\gamma} \frac{\mathbf{1}}{\mathbf{2}} \sum_{\boldsymbol{k}=\mathbf{1}}^{\boldsymbol{l}} \boldsymbol{\xi}_{\boldsymbol{i}} \tag{11}$$

Where γ is a parameter that weighs the maximum interval and the minimum classification error. Its corresponding Lagrangian function is:

$$\begin{aligned} &\boldsymbol{L}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &= \boldsymbol{J}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}) \\ &- \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{l}} \boldsymbol{\alpha}_{\boldsymbol{i}} \{\boldsymbol{y}_{\boldsymbol{i}} [\boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{\Psi}(\boldsymbol{x}_{\boldsymbol{i}}) + \boldsymbol{b}] - \mathbf{1} + \boldsymbol{\xi}_{\boldsymbol{i}}\} - \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{l}} \boldsymbol{\beta}_{\boldsymbol{i}} \boldsymbol{\xi}_{\boldsymbol{i}} \end{aligned} \tag{12}$$

The minimum value for w, b, ξ needs to be solved, and the maximum value for the non-negative Lagrangian multiplier α, β.

$$\min_{\omega,b,\xi} \max_{\alpha,\beta} L(w, b, \xi; \alpha, \beta) \tag{13}$$

The parameters in the formula must satisfy the following conditions:

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{l} \alpha_i y_i \Psi(x_i) \tag{14}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{15}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.201909_5(10).0008

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \alpha_i + \beta_i = \gamma, i = 1, \cdots, l \tag{16}$$

Substituting (11) into Lagrangian and considering equations (12) and (13), the following dual problem [19] is obtained:

$$\boldsymbol{Q}(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \boldsymbol{\alpha_i} - \frac{\mathbf{1}}{\mathbf{2}} \sum_{i=1}^{l} \sum_{j=1}^{l} \boldsymbol{\alpha_i \alpha_j y_i y_j \Psi(x_i)^T \Psi(x_j)} \tag{17}$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0, 0 \le \alpha_i \le \gamma$$

2.2.1 Least squares support vector machine (LS-SVM)

Compared with SVMS, the optimization problem of Least Squares Support Vector Machine (LS-SVM) is [19]:

$$\min_{\boldsymbol{w,b,e}} \boldsymbol{J}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{e}) = \frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{w^T w} + \boldsymbol{\gamma} \frac{\mathbf{1}}{\mathbf{2}} \sum_{k=1}^{l} \boldsymbol{e_i^2} \tag{18}$$

The conditions are:

$$\mathrm{y_i}(w^T \Psi(x_i) + b) \ge 1 - e_\mathrm{i}, i = 1, \cdots, l \tag{19}$$

The corresponding Lagrangian function is:

$$\boldsymbol{L}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{e}; \boldsymbol{\alpha}) = \boldsymbol{J}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{e}) - \sum_{i=1}^{l} \boldsymbol{\alpha_i} \{\boldsymbol{y_i}[\boldsymbol{w^T \Psi(x_i)} + \boldsymbol{b}] - \mathbf{1} + \boldsymbol{e_i}\} \tag{20}$$

Where α_i is a Lagrangian multiplier that satisfies the sample corresponding to $\alpha_i \neq 0$, which is the support vector.

The optimal conditions are as follows:

$$\frac{\partial \mathbf{L}}{\partial \mathbf{w}} = \mathbf{0} \Rightarrow \boldsymbol{w} = \sum_{i=1}^{l} \boldsymbol{\alpha_i y_i} \boldsymbol{\Psi}(\boldsymbol{x_i}) \tag{21}$$

$$\frac{\partial \mathbf{L}}{\partial \boldsymbol{b}} = \mathbf{0} \Rightarrow \sum_{i=1}^{l} \boldsymbol{\alpha_i y_i} = \mathbf{0} \tag{22}$$

$$\frac{\partial \mathbf{L}}{\partial \boldsymbol{e_i}} = \mathbf{0} \Rightarrow \boldsymbol{\alpha_i} = \boldsymbol{\gamma e_i}, \qquad \boldsymbol{i} = \mathbf{1}, \cdots, \boldsymbol{l} \tag{23}$$

The matrix form of equations (19)-(21) is:

$$\begin{bmatrix} \boldsymbol{\Omega} & \boldsymbol{Y} \\ \boldsymbol{Y^t} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} \tag{24}$$

Where Ω, Y and 1 are respectively

$$\boldsymbol{\Omega_{ij}} = \boldsymbol{y_i y_j} \boldsymbol{\Psi}(\boldsymbol{x_i})^{\boldsymbol{T}} \boldsymbol{\Psi}(\boldsymbol{x_j}) + \frac{\boldsymbol{\delta_{ij}}}{\boldsymbol{\gamma}} \tag{25}$$

$$\mathrm{Y} = [\mathrm{y}_1, \cdots, y_l]^T$$

$$\boldsymbol{\delta_{ij}} = \begin{cases} \mathbf{1} & \boldsymbol{i} = \boldsymbol{j} \\ \mathbf{0} & \boldsymbol{i} \neq \boldsymbol{j} \end{cases}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.201909_5(10).0008

2.2.2 Kernel function

For linearly separable sample vectors, the SVM will find the hyperplane from the maximum Euclidean distance of the support vector. If the linear boundary is not suitable, the SVM can map the input vector from the input space to the high-dimensional feature space [17]. By choosing a nonlinear mapping as a priori, the SVM constructs the best separated hyperplane in this high dimensional space. The idea of the kernel function is to enable the above operations to be performed in the input space rather than the potentially high dimensional feature space. Therefore, there is no need to evaluate the inner product in the feature space, which provides a way to solve the dimensional curse. Reproducing Nuclear Hilbert Space Theory (RKHS) (Wahba. 1990: Aronszajn, 1950: Girosi. 1997: Heckman. 1997) shows the inner product in the feature space and the equivalent kernel in the input space.

$$\mathrm{K}(\mathrm{x},\mathrm{x_i}) = \Psi(x)^T\Psi(x_i) \tag{26}$$

SVM can have different kernel functions. In this paper, Gaussian radial basis function (GRBF) is used to implement SVM kernel, and its form is given by: RKHS) (Wahba.1990: Aronszajn, 1950: Girosi. 1997: Heckman. 1997) Indicates the inner product in the feature space and the equivalent kernel in the input space

$$\mathbf{K}(\mathbf{x},\mathbf{x_i}) = \mathbf{exp}\left\{-\frac{\|\boldsymbol{x}-\boldsymbol{x_i}\|_2^2}{\boldsymbol{\sigma}^2}\right\} \tag{27}$$

SVM is a solution to the neural network overfitting problem. Using the kernel technique of RBF, a clustering method is used to select the center point [12]. Each of these support vectors contributes to the local Gaussian function.

3. Training of the Neuro-SVM model

3.1 Acquisition of experimental data

The experimental data in this paper uses ADVISOR automotive simulation software for simulation. ADVISOR (Advanced Vehicle Sim-ulator) is an advanced vehicle simulation software developed by the US Renewable Energy Laboratory under the Matlab/Simulink software environment [18]. This paper adopts ADVISOR's backward simulation method, which is a simulation method for calculating the performance of various components in a vehicle under the premise that the vehicle can meet the required driving trajectory of the existing road cycle (including vehicle driving speed, road gradient and vehicle dynamic quality)[19].

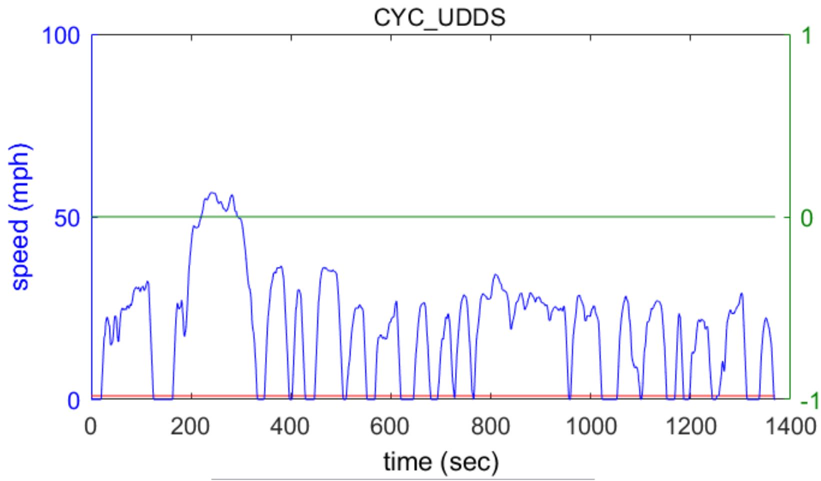

Figure 2. UDDS operating conditions, speed versus time curve

In ADVISOR, test routes that are widely used in electric vehicle performance testing are: ECE operating conditions, UDDS operating conditions, HWFET operating conditions, etc[19]. The UDDS operating conditions used in this paper are the urban road cycling conditions developed by the US

ISSN: 2414-1895 DOI: 10.6919/ICJE.201909_5(10).0008

Environmental Protection Agency to test the cyclic performance of various performances of vehicles traveling under urban roads [17]. Its speed versus time curve is shown in Figure 2.

This paper refers to the relevant technical parameters of a domestic electric vehicle, and builds a simulation platform on ADVISOR. The main technical parameters of the simulated electric vehicle are shown in Table 1 [31].

Table 1. Technical parameters of electric vehicle

Parameter Name	Parameter value
Dimensions /mm	4500x1705x1495
Wheelbase /mm	2605
Track (front/rear)/mm	1470/1460
Vehicle quality /kg	1490
Full load quality /kg	1895
Front and rear tire specifications	195/60R15(0.308)
Rolling radius /mm	298
Tire rolling resistance coefficient	0.01
Frontal area /$\mathbf{m^2}$	2.3
Wind resistance coefficientCd	0.3
Driveline mechanical efficiency	0.9

Under the UDDS condition, one round of driving from SOC=100% to SOC=0%, the total driving distance is 14.49km, and the total driving speed is 2329s, and the maximum driving speed is 91.25km/h. The lithium ion battery used in the simulated electric vehicle is lithium iron phosphate. The total simulation time is 1367s, the sampling step is 0.1s, and the actual sampling has obtained 13691 data points.

This paper samples the electric vehicle battery with the following characteristics: charge and discharge current (Figure 4), battery temperature (Figure 5), battery voltage (Figure 6), battery output power (Figure 7), battery pack available power (Figure 8). The sign (laebls) is the battery's SoC (Figure 3).

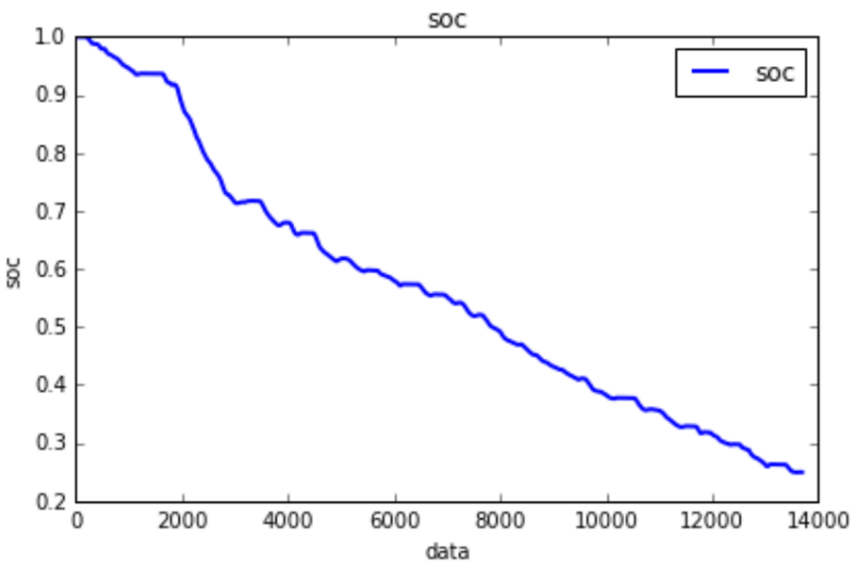

Figure 3. Battery SOC

ISSN: 2414-1895
DOI: 10.6919/ICJE.201909_5(10).0008

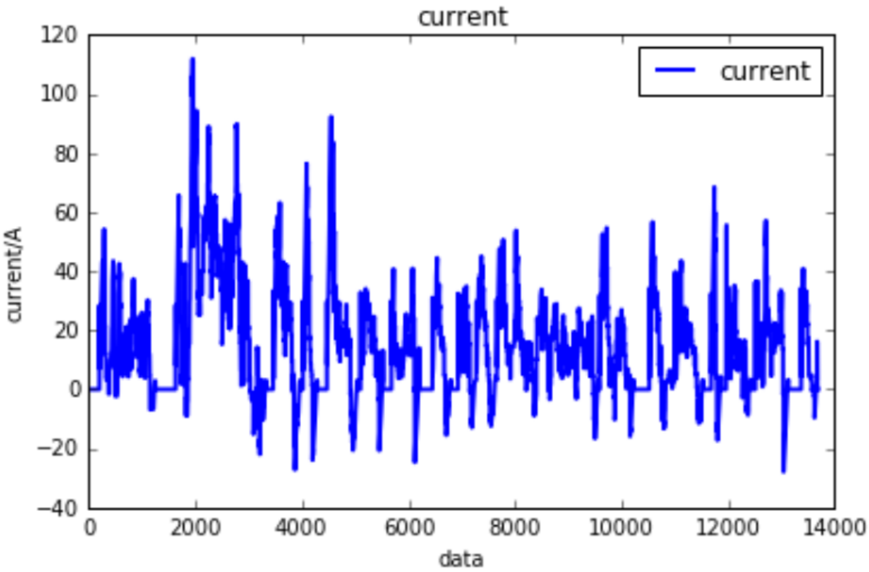

Figure 4. battery current

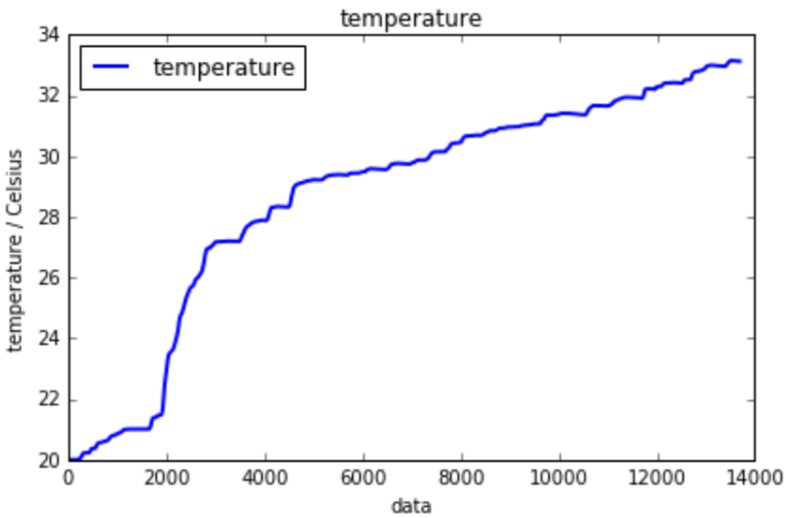

Figure 5. battery temperature

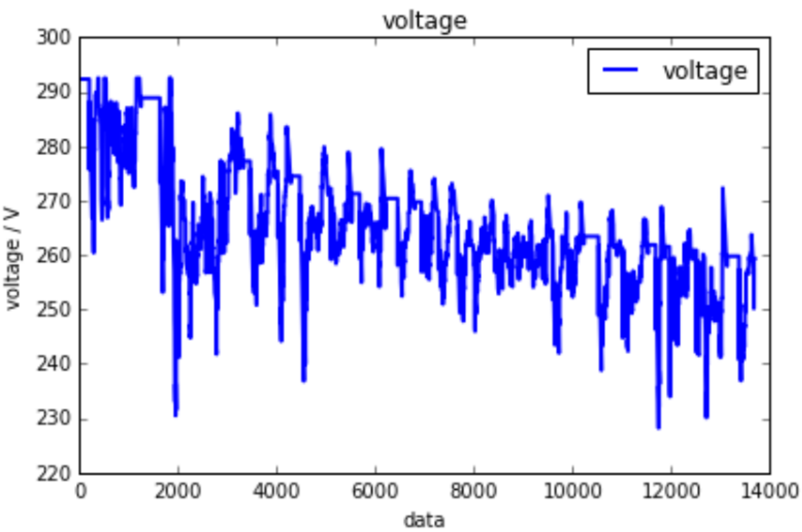

Figure 6. battery voltage

ISSN: 2414-1895
DOI: 10.6919/ICJE.201909_5(10).0008

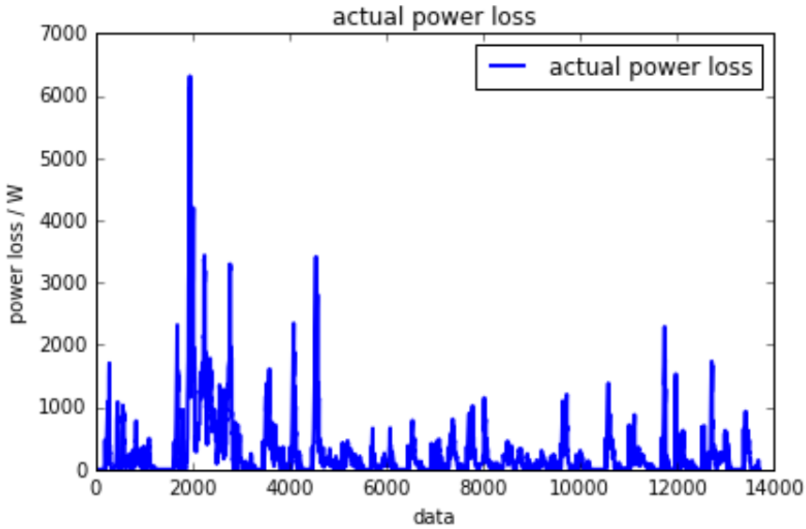

Figure 7. Battery pack actual power

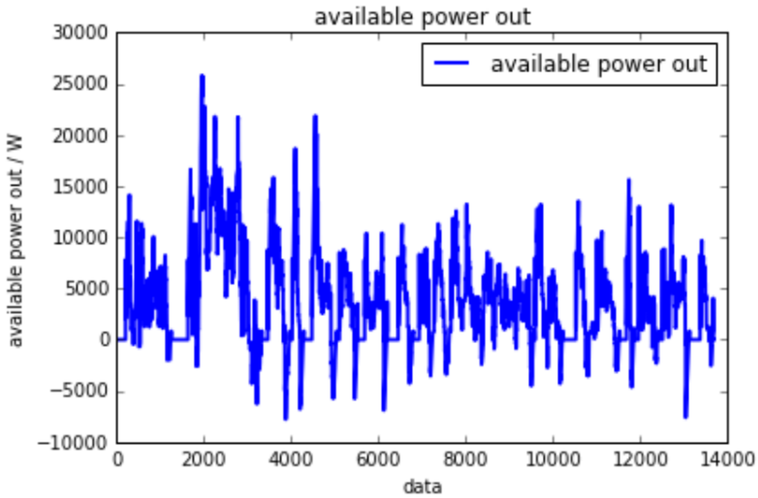

Figure 8. Battery SOC battery pack available power

3.2 data preprocessing

Before modeling, the paper firstly performs data preprocessing: the original data is normalized and the feature selection of SBS can not only make the modeling effect of Neuro-SVM better, but also reduce the computational complexity and model complexity of the model. After data preprocessing, the original training data is reconstructed into a supervised learning problem of continuous input and output pairs.

3.2.1 Feature scaling

Because the range of characteristic values of the sampled data is very different: the battery temperature is only a few dozen, and the battery voltage can reach several hundred, and the power can even be tens of thousands. If the feature is not scaled, the power pre-coefficient may be extremely small, and the pre-temperature coefficient may be extremely large, which invisibly increases the weight of the battery temperature on the soc estimation result. So this article uses Python's scikit-learn machine learning library to feature feature scaling and standardization. By feature scaling, the feature values are scaled to the same interval, avoiding the value of the feature affecting the model weight. Since the eigenvalues of the data need not be specified in the [0, 1] interval, this paper uses normalization to scale, which is defined as equation (30).

$$\mathrm{x}_{\mathrm{std}}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x} \tag{28}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.201909_5(10).0008

Where μ_x is the mean of a characteristic column of the sample, and σ_x is its standard deviation. After standardization, the value of the feature column will be in a standard normal distribution with a mean of 0 and a variance of 1.

3.2.2 Feature selection

In this paper, the Sequential Backward Selection (SBS) algorithm is used for dimensionality reduction. SBS is a greedy search algorithm that compresses the original d-dimensional feature space into a smaller dimensional k-dimensional feature subspace, thereby eliminating irrelevant features and noise, and retaining only the information most relevant to the prediction result, thereby improving Calculate efficiency and reduce model generalization error [20]. In this paper, one or several features are deleted from the feature set in turn, so that after the feature is deleted, the performance loss caused by the model is minimized until the dimension of the new feature subspace satisfies the specified number of dimensions. Finally, the three external input characteristics most relevant to the SOC estimation are selected by SBS: battery current, battery voltage, and battery temperature.

3.2.3 Feature selection

There are two indicators commonly used to measure the accuracy of model output estimates: Root Mean Square Error (RMSE) and Maximum Error (Max Error) [20]. The formula of RMSE is shown in equation (31).

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}(\hat{y} - y_i)^2} \tag{29}$$

Where $\hat{y}$ is the predicted value, y_i is the true value of the ith sample, and n is the total number of samples included in the calculation.

4. Experimental verification and result analysis

After data preprocessing, all simulation data are randomly divided into three sub data sets according to 8:1:1: training set, verification set, and test set. The training set is used to train the Neuro-SVM model. The verification set is used to select the most suitable model parameters. The test set is used to simulate the possible situation of the model in reality and verify the validity of the model.

Table 2. Neuro-SVM hyperparameters

LSTM neuron number	Activation function	Batch size /Bs	Hidden layer	C	gamma	Maximum error	RMSE
20	Sigmoid	30	-	1000	0.001	0.199666761	0.008251492
50	linear	75	4	50	1	0.144777416	0.026886295
100	tanh	16	8	10	0.01	0.063832017	0.007759814
100	tanh	20	2	1	1	0.064515105	0.007264439

The final parameters are adjusted as follows: LSTM-RNN: LSTM neurons = 100, activation function = tanh, batch size = 20/Bs; LS-SVM: C = 1.0, kernel = rbf, gamma = 1.0.

On the test set, the RMSE was 0.7264% and the maximum error was 0.06451. Far less than 5% of the actual application requirements. As shown in Figure 10 and Figure 11. This indicates that the accuracy of the Neuro-SVM model meets the requirements for the estimation of the state of charge of the power battery.

ISSN: 2414-1895
DOI: 10.6919/ICJE.201909_5(10).0008

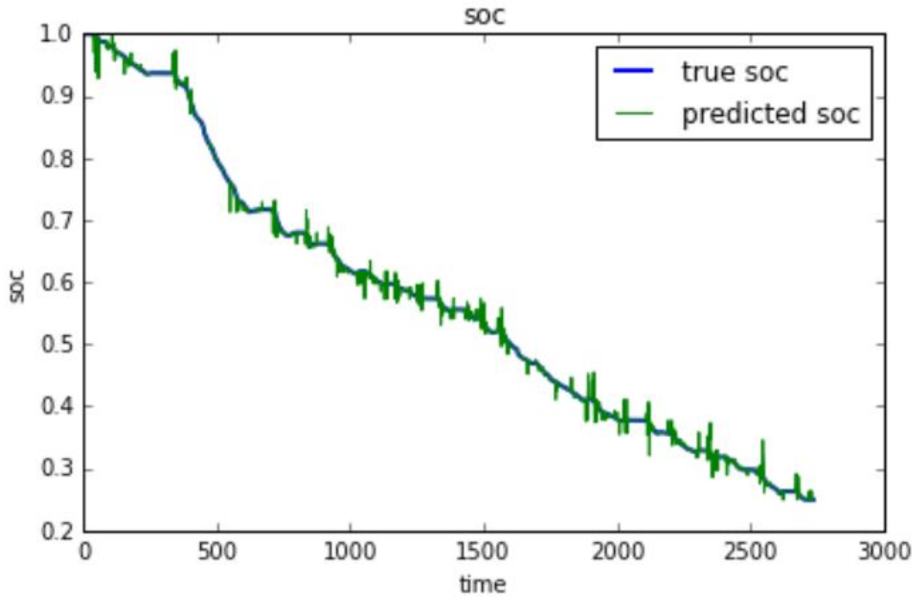

Figure 9. Comparison of battery SOC true and predicted values

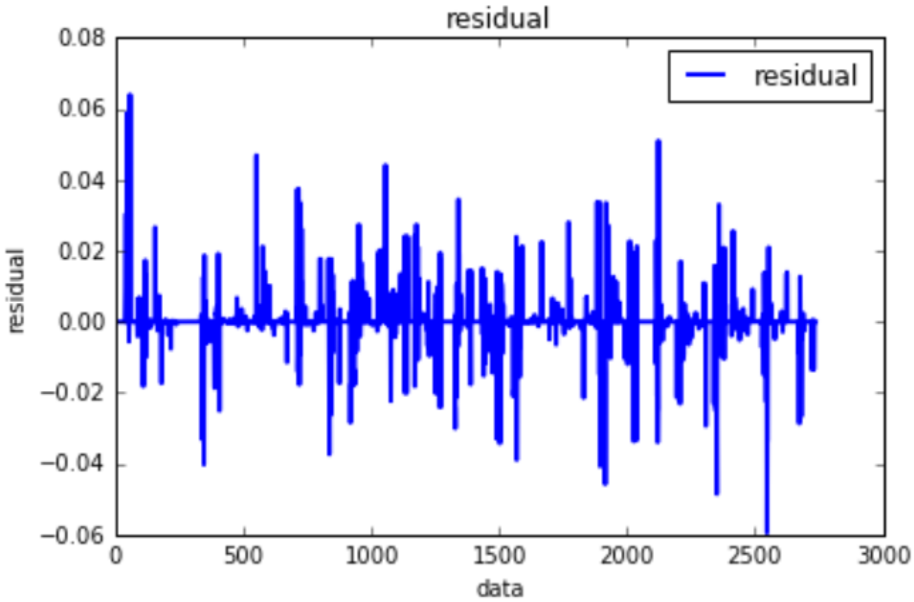

Figure 10. Error between the actual and predicted values of the battery SOC

5. Conclusion

This paper combines the cyclic neural network (RNN) and support vector machine (SVM) to model the state of charge (SOC) of lithium iron phosphate (LiFePO4) battery. The experimental data of the simulated car under UDDS conditions obtained by ADVISOR verified the validity of the Neuro-SVM model. After data preprocessing, the lithium-ion battery current, voltage and temperature are taken as inputs, and the SOC is used as an output to establish a three-input-output Neuro-SVM model. The model's predicted SOC root mean square error (RMSE) is 0.726%, which is much lower than the actual application requirement of 5%. This indicates that the accuracy of the Neuro-SVM model meets the requirements for the estimation of the state of charge of the power battery.

References

[1] Li A, Hu X, Li T, et al. Research on the Prediction Method of Power Battery SOC Based on Deep Learning[C]// IEEE Third International Conference on Data Science in Cyberspace. IEEE Computer Society, 2018:673-679.

ISSN: 2414-1895 DOI: 10.6919/ICJE.201909_5(10).0008

[2] Huang G B, Zhou H, Ding X, et al. Extreme learning machine for regression and multiclass classification[J]. IEEE Trans Syst Man Cybern B Cybern, 2012, 42(2):513-529.

[3] Charkhgard M, Farrokhi M. State-of-Charge Estimation for Lithium-Ion Batteries Using Neural Networks and EKF[J]. IEEE Transactions on Industrial Electronics, 2010, 57(12):4178-4187.

[4] Meng J, Luo G, Gao F. State-of-charge estimation for lithium-ion battery using AUKF and LSSVM[C]// Transportation Electrification Asia-Pacific. IEEE, 2014:1-6.

[5] Maharjan L, Inoue S, Akagi H, et al. State-of-Charge (SOC)-Balancing Control of a Battery Energy Storage System Based on a Cascade PWM Converter[J]. IEEE Transactions on Power Electronics, 2009, 24(6):1628-1636.

[6] Vasebi A, Partovibakhsh M, Bathaee S M T. A novel combined battery model for state-of-charge estimation in lead-acid batteries based on extended Kalman filter for hybrid electric vehicle applications[J]. Journal of Power Sources, 2007, 174(1):30-40.

[7] Liu Hao. Research on SOC estimation method for lithium ion battery for electric vehicles based on EKF [D]. Beijing Jiaotong University, 2010.

[8] Feng Guang. Modeling and Simulation of SOC Estimation of Lithium Ion Battery Based on EKF [D]. Wuhan University of Technology, 2013.

[9] Zhao Tianyi. Research on state estimation method of lithium ion battery based on improved Kalman filter[D]. Harbin Institute of Technology, 2016.

[10] Sun Yue. Research on State of Charge Estimation Algorithm Based on Lithium Ion Battery[D]. Guizhou University, 2017.

[11] Chan S C, Zhou Y, Lau W Y. APPROXIMATE QR-BASED ALGORITHMS FOR RECURSIVE NONLINEAR LEAST SQUARES ESTIAMTION[J]. 2005:4333-4336 Vol. 5.

[12] Hussein A A. Capacity fade estimation in electric vehicles Li-ion batteries using artificial neural networks[C]// Energy Conversion Congress and Exposition. IEEE, 2013:677-681.

[13] MA You-Liang, CHEN Quan-Shi, QI Zhan-Ning. Definition and Detection Method of Battery SOC for Electric Vehicles[J]. Journal of Tsinghua University(Science and Technology), 2001, 41(11): 95-97.

[14] Lin Chengtao, Wang Junping, Chen Quanshi. Principle and Application of SOC Estimation Method for Electric Vehicles[J]. Batteries, 2004, 34(5): 376-378.

[15] JIANG You煊[1, Li Junxiu. Research Progress in SOC Estimation Method for Lithium Ion Batteries[J]. Power Technology, 2018(2).

[16] Du J, Liu Z, Chen C, et al. Li-ion battery SOC estimation using EKF based on a model proposed by extreme learning machine[M]. 2012.

[17] Hu Liping. Research on SOC estimation algorithm for power lithium-ion battery based on support vector machine[D]. Hubei University of Technology, 2014.

[18] WAAG W，FLEISCHER C，VIEJO C B，et al. Critical review of the methods for monitoring of lithium-ion battery in electric and hybrid vehicles[J]. Journal of Power Sources，2014，258：321-339.10.1016/j.jpowsour.2014.02.064.

[19] Antón J C Á, Nieto P J G, Juez F J D C, et al. Battery state-of-charge estimator using the SVM technique[J]. IEEE Transactions on Power Electronics, 2013, 37(9):6244-6253.

[20] Hu J N, Hu J J, Lin H B, et al. State-of-charge estimation for battery management system using optimized support vector machine for regression[J]. Journal of Power Sources, 2014, 269(3):682-693.

[21] Lin C, Mu H, Xiong R, et al. A novel multi-model probability battery state of charge estimation approach for electric vehicles using H-infinity algorithm[J]. Applied Energy, 2016, 166:76-83.

[22] Mu H, Xiong R, Sun F. A Novel Multi-model Probability Based Battery State-of-charge Fusion Estimation Approach ☆[J]. Energy Procedia, 2016, 88:840-846.

[23] Kang L W, Zhao X, Ma J. A new neural network model for the state-of-charge estimation in the battery degradation process[J]. Applied Energy, 2014, 121(5):20-27.

[24] Zhao Youqun, Zhou Xiaofeng, Liu Yingjie. SOC Estimation of Lithium Batteries Based on Extended Kalman Particle Filter Algorithm[J]. China Mechanical Engineering, 2015, 26(3): 394-397.

ISSN: 2414-1895 DOI: 10.6919/ICJE.201909_5(10).0008

[25] Sun Dong, Chen Xikun. Estimation of State of Charge of Lithium Batteries Based on Discrete Sliding Mode Observer[J]. Proceedings of the CSEE, 2015, 35(1):185-191.

[26] He Yao, Qin Shaoxun, Liu Xintian, et al. SOC Estimation of Power Battery Based on Fractional Order Unscented Particle Filter[J]. Automotive Technology, 2018(5).

[27] Sepasi S, Ghorbani R, Liaw B Y. Improved extended Kalman filter for state of charge estimation of battery pack[J]. Journal of Power Sources, 2014, 255(6):368-376.

[28] Xu Jie. Accurate estimation of SOC of power battery pack based on Kalman filter[D]. Hangzhou University of Electronic Science and Technology, 2009.

[29] Dai Haifeng, Wei Xuezhe, Sun Zechang. Estimation of State of Charge of Lithium Ion Power Battery for Fuel Cell Vehicle Based on Extended Kalman Filter[J]. Journal of Mechanical Engineering, 2007, 43(2): 92-95.

[30] Xu Jie. Accurate estimation of SOC of power battery pack based on Kalman filter[D]. Hangzhou University of Electronic Technology, 2009.

[31] Domenico D D, Fiengo G, Stefanopoulou A. Lithium-ion battery state of charge estimation with a Kalman Filter based on a electrochemical model[C]// IEEE International Conference on Control Applications. IEEE, 2008:702-707.

[32] Zhu Yuan, Han Xiaodong, Tian Guangyu. Research on SOC Prediction Technology for Electric Vehicle Power Battery[J]. Power Technology, 2000, 24(3): 153-156.

[33] Gong Xuegeng, Qi Platinum, Liu Youbing, et al. Power Battery Model and SOC Estimation Strategy for Electric Vehicles[J]. Power Technology, 2004, 28(10): 633-636.

[34] Gong Xuegeng, Qi Platinum. Modeling and Analysis of Battery Equilibrium Control for Electric Vehicles[J]. Batteries, 2005, 35(1): 37-38.

[35] Wang Hongwei, Liu Jun, Xiao Haiqing, et al. Comparative Analysis of Related Standards for Lithium Ion Power Battery at Home and Abroad[J]. Electronic Components & Materials, 2012, 31(10): 89-92.

[36] Wang Hongwei, Deng Shuang, Xiao Haiqing, et al. Current status of domestic lithium-ion battery for electric vehicles[J]. Electronic Components & Materials, 2012, 31(6): 84-86.

[37] Zhang Zhongyi, Zhai Jiayu, Yang Lin, et al. Battery Management System for Hybrid Electric Vehicles[J]. Mechanical & Electrical Engineering Technology, 2006, 35(1): 61-64.

[38] Xiong R, Cao J, Yu Q, et al. Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles[J]. IEEE Access, 2018, 6(99):1832-1843.

[39] Feng Jin, FENGJin. BP neural network estimation of SOC training data selection for lithium-ion batteries[J]. Power Technology, 2016, 40(2): 283-286.

[40] Wu Haidong, Ren Xiaoming, Na Wei, et al. Improving the SOC of lithium-ion battery by using improved time method combined with neural network[J]. Batteries, 2016, 46(1).

[41] Dai Haifeng, Wei Xuezhe, Sun Zechang. Estimation of State of Charge of Lithium Ion Power Battery for Fuel Cell Vehicle Based on Extended Kalman Filter Algorithm[J]. Journal of Mechanical Engineering, 2007, 43(2): 92-95.

[42] Shang Yunlong, Zhang Chenghui, Cui Naxin, et al. Estimation of state of charge of lithium-ion battery based on fuzzy neural network optimized extended Kalman filter[J]. Control Theory and Applications, 2016, 33(2): 212-220.

[43] Dai Haifeng, Sun Zechang, Wei Xuezhe. Estimation of Internal State of Lithium Ion Power Battery for Electric Vehicles Using Dual Kalman Filter Algorithm[J]. Journal of Mechanical Engineering, 2009, 45(6): 95-101.

[44] Tian Y, Li D, Tian J, et al. An optimal nonlinear observer for state-of-charge estimation of lithium-ion batteries[C]// IEEE Conference on Industrial Electronics and Applications. IEEE, 2018:37-41.

[45] Cai C, Du D, Liu Z, et al. State-of-charge (SOC) estimation of high power Ni-MH rechargeable battery with artificial neural network[M]. 2002.

[46] Zheng D, Wang H, An J, et al. Real-time estimation of battery state of charge with metabolic grey model and LabVIEW platform[J]. IEEE Access, PP(99):1-1.

[47] Jeong J, Jeong S, Kim C, et al. A 42nJ/conversion on-demand state-of-charge indicator for miniature IoT Li-ion batteries[C]// Design Automation Conference. IEEE, 2018:C206-C207.

ISSN: 2414-1895 DOI: 10.6919/ICJE.201909_5(10).0008

[48] Yan Youzhen, Chen Yuanzhang. A review of the main strategies of machine learning[J]. Journal of Computer Applications, 2004, 21(7): 4-10.

[49] Chen Kai, Zhu Wei. Overview of Machine Learning and Related Algorithms[J]. Journal of Statistics and Information, 2007, 22(5): 105-112.

[50] He Qing, Li Ning, Luo Wenjuan, et al. Overview of machine learning algorithms under big data [C]// China Computer Society Artificial Intelligence Conference. 2013.

[51] Gao Yang, Chen Shifu, Lu Xin. A Review of Intensive Learning Research[J]. Acta Automatica Sinica, 2004, 30(1): 86-100.