Applied Energy 242 (2019) 1259–1273

Contents lists available at ScienceDirect

Applied Energy

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

Big data driven lithium-ion battery modeling method based on SDAE-ELM algorithm and data pre-processing technology

Shuangqi Li, Hongwen He*, Jianwei Li*

National Engineering Laboratory for Electric Vehicles and Collaborative Innovation Center of Electric Vehicles in Beijing, Beijing Institute of Technology, Beijing 100081, China

HIGHLIGHTS

- A battery temperature-dependent model is developed based on SDAE-ELM algorithm.
- A new battery big data processing and quality assessment approach is proposed.
- A new training method for SDAE-ELM model is proposed and proved effectively.
- A C-BMS is established and with which a cloud-vehicle joint working mode is built.

ARTICLE INFO

Keywords:
Electric vehicles
Battery energy storage
Temperature-dependent model
Battery management system
Big data
Deep learning

ABSTRACT

As one of the bottleneck technologies of electric vehicles (EVs), the battery hosts complex and hardly observable internal chemical reactions. Therefore, a precise mathematical model is crucial for the battery management system (BMS) to ensure the secure and stable operation of the battery in a multi-variable environment. First, a Cloud-based BMS (C-BMS) is established based on a database containing complete battery status information. Next, a data cleaning method based on machine learning is applied to the big data of batteries. Meanwhile, to improve the model stability under dynamic conditions, an F-divergence-based data distribution quality assessment method and a sampling-based data preprocess method is designed. Then, a lithium-ion battery temperature-dependent model is built based on Stacked Denoising Autoencoders- Extreme Learning Machine (SDAE-ELM) algorithm, and a new training method combined with data preprocessing is also proposed to improve the model accuracy. Finally, to improve reliability, a conjunction working mode between the C-BMS and the BMS in vehicles (V-BMS) is also proposed, providing as an applied case of the model. Using the battery data extracted from electric buses, the effectiveness and accuracy of the model are validated. The error of the estimated battery terminal voltage is within 2%, and the error of the estimated State of Charge (SoC) is within 3%.

1. Introduction

The battery is one of the vital components of electric vehicles (EVs). As an important instrument for battery management, a typical battery management system (BMS) in EVs is shown in Fig. 1, the main function of which is to measure, model and manage thousands of battery cells in the vehicle [1] and thus improve the reliability of the battery pack. Accurate battery modeling is a prerequisite for stable and safe operation of BMS [2], which influences the battery durability and safety management directly.

In recent years, the development of a flexible, self-reconfigurable and reliable BMS has become one of the most crucial technologies for EVs [3]. The existing research on the lithium-ion battery and its management system mainly focuses on parameter identification [4], State of

Abbreviations: EVs, electric vehicles; BMS, battery management system; C-BMS, cloud-based battery management system; V-BMS, BMS in vehicles; SoC, State of Charge; CAN, Controller Area Network; UDDS, Urban Dynamometer Driving Schedule; HPPC, Hybrid Pulse Power Characteristic; DST, Dynamic stress test; AE, Automatic Encoder; DAE, Denoising Autoencoder; SDAE, Stacked Denoising Autoencoders; SDAE-ELM, Stacked Denoising Autoencoders – Extreme Learning Machine; SVR, Support Vector Regression; ELM, Extreme Learning Machine; BP, Back Propagation; SMOTE, Synthetic Minority Oversampling Technique; PCA, Principal Component Analysis; KL divergence, Kullback-Leibler divergence; TIM, Temperature-independent model; TDM, Temperature dependent model; MAPE, Mean Absolute Percent Error; Std, Standard Deviation

* Corresponding authors.
E-mail addresses: hwhebit@bit.edu.cn (H. He), jianwei.li@bath.edu (J. Li).

https://doi.org/10.1016/j.apenergy.2019.03.154
Received 13 December 2018; Received in revised form 12 March 2019; Accepted 13 March 2019

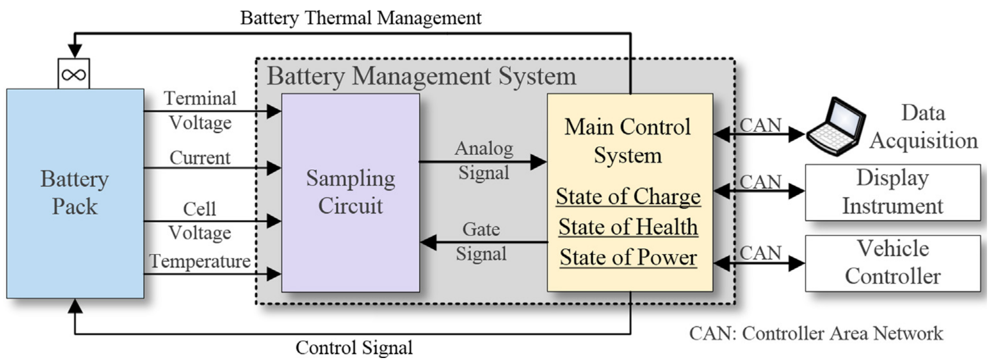

Fig. 1. The battery management system in vehicles.

Charge (SoC) estimation [5], and fault detection [6] based on the equivalent circuit models and electrochemical models. But there are two fundamental issues and critical challenges still lying in the modeling and management of batteries: (1) The effects of temperature are rarely taken into consideration in the battery modeling, which may result in poor model accuracy. (2) The limited data volume and calculating ability available in the BMS results in poor adaptability of the battery model in dynamic conditions.

As temperature is a key factor in battery modeling, establishing a model including temperature variables is becoming increasingly important [7]. The physicochemical properties of the battery are generally temperature-dependent and have a significant effect on the life and reliability of the battery [8]. To improve the model accuracy, reference [9] proposed an SoC estimation method based on an unscented Kalman filter, and the SoC is modified by temperature compensation coefficient. The model parameters were expressed as a nonlinear function of temperature in reference [10], but a large volume of experimental data are required for modeling, so that the accuracy of the model in practice is unsatisfactory. In reference [11], the fuzzy logic method was used in battery modeling, and the influence of temperature on the parameters is described by fuzzy rules. However, the relationship between the parameters of the equivalent circuit model and the temperature is extremely complicated and difficult to parameterize. The above methods are not able to adequately reflect the effect of temperature on the battery.

On the other hand, the data used for battery modeling in existing BMS are generally derived from a single vehicle, where the data volume and calculating ability is limited. But the chemical reactions in the battery are affected by multiple factors, such as temperature, State Of Health [12] and dynamic conditions [13]. It is difficult to accurately model the battery in the case of incomplete data. Additionally, when experimenting and testing the battery model, the majority of the data come from standard test conditions, such as Urban Dynamometer Driving Schedule (UDDS), Hybrid Pulse Power Characteristic (HPPC), and Dynamic Stress Test (DST) [14,15], which are quite different from the dynamic conditions during normal driving.

The neural network algorithms have a strong nonlinear mapping ability, which can automatically learn useful knowledge from the data without an accurate mathematical model [16]. Hence, in recent years, neural network algorithms have been widely used in battery modeling. A neural network based approach for lithium-ion battery modeling and state estimation was proposed in reference [17]. The temperature was taken into consideration while modeling the battery without any mathematical model, and the experimental results highlighted the high modeling accuracy. With the support of a large volume of data, a battery model based on a neural network algorithm is able to fully reflect the influence of temperature.

Although neural network algorithms perform well when establishing the temperature-dependent battery model, most of them can only be regarded as shallow structure algorithms, with limited approximation ability to tackle complex functions in the cases of finite computational units. However, the battery data own the characteristics of high dimensionality and large quantity, which causes great difficulties for the neural network training, especially under complicated dynamic conditions as aforementioned. A black box model for the battery based on Back Propagation (BP) neural network algorithm was established in reference [18]. It is able to accurately represent the highly nonlinear mapping between the input and output of the battery, but a tremendous amount of data are needed to improve the model accuracy and adaptability. Fortunately, with the advent of "big data", the amount of data is no longer a problem in battery modeling [19]. Theoretically, the accuracy and adaptability of the black box model can be greatly improved. However, as the data volume increases, the training process becomes more complicated, and there is a serious overfitting problem in the case of poor data quality [20]. To improve the accuracy of the battery model, the Support Vector Regression (SVR) algorithm was used in reference [21] and the experiment results showed that the estimated SoC error is within 4%. However, as the amount of training data increases, the training time increases exponentially.

Compared to the neural network algorithm, the deep learning algorithm has the ability to effectively simulate highly nonlinear mapping between the input and output [22]. Therefore, it is able to accurately model the battery under multi-variable environment and dynamic conditions. At present, deep learning algorithms are widely used in many fields, for example, electric load forecasting [23], traffic speed prediction [24], and fault diagnosis systems [25].

Based on the discussion above, in order to establish a temperature-dependent battery model that can adapt to the dynamic conditions and a multi-variable environment, this paper proposes a big data driven battery modeling method based on Stacked Denoising Autoencoders-Extreme Learning Machine (SDAE-ELM) algorithm. The key contributions of this paper are as follows:

1) New technique: To overcome the difficulty of limited data volume and calculating ability available in BMS in vehicles (V-BMS), we propose, for the first time, the architecture of Cloud-based BMS (C-BMS) and explore the conjunction working mode combining C-BMS with V-BMS.
2) New problems: Although the machine learning algorithm has been widely used in battery modeling, our experiment indicates that it is prone to over-fitting. We found that over-fitting is mainly caused by three factors: poor data quality, poor data distribution quality, and inappropriate training method. Specifically, this problem is more serious when modeling the battery under a multi-variable environment and dynamic conditions.
3) New method: To realize the proposed technique for battery management and to solve the key problems listed in (2), this paper proposes the following four methods: First, to improve data quality, we propose a machine learning algorithm-based data cleaning method for EV's big data. Then, to relieve the uneven data

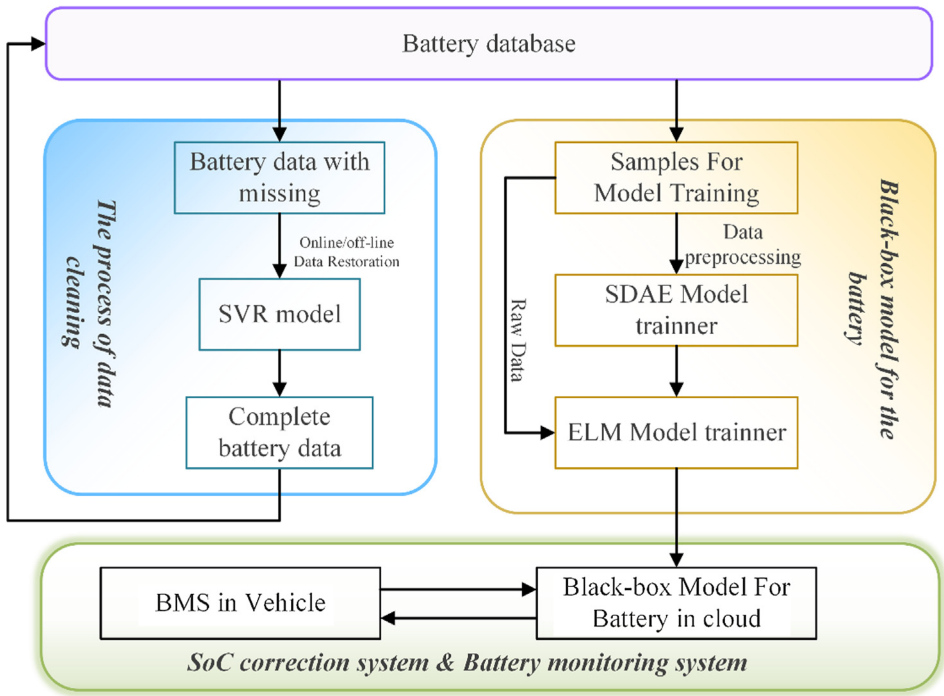

Fig. 2. The structure of this paper.

distribution and improve the battery model adaptability in a multi-variable environment and dynamic conditions, we propose a novel evaluation method for uneven data distribution and a novel data preprocessing method based on the F-divergence algorithm and sampling algorithm. Third, we make the first attempt to apply the SDAE-ELM algorithm to battery modeling issues. The idea is to fully excavate the hidden features in battery data. Finally, to build a more precise and robust model, we propose a new training method that combines the data preprocessing with SDAE-ELM algorithm.

The structure of this paper is shown in Fig. 2, and the remainder of the paper is organized as follows: Section 2 briefly introduces the C-BMS proposed in the paper. Section 3.1 proposes a machine learning-based data cleaning method. Section 3.2 explains the proposed data uneven distribution evaluating method and the data preprocessing method. Section 3.3 proposes a battery modeling method based on SDAE-ELM algorithm and explains the proposed conjunction working mode between C-BMS and V-BMS. Section 4.1 explains the data used in our work, as well as the details of the modeling parameters. Results and discussions are shown in Sections 4.2–4.4. Conclusions and future studies are listed in part 5.

2. The Cloud-based battery management system

Cloud data centers have high storage capacities and calculating abilities advantages. Therefore, this paper proposes a cloud-based battery management system. The C-BMS is shown in Fig. 3. For a single electric vehicle, a C-BMS with a data transmission module is established on the basis of the V-BMS, and the battery data are uploaded to the cloud in real-time for further analysis. The cloud receives and stores the data of all EVs that share the same specification, and a multi-dimensional, multistate, multifactor complex data space is provided for the modeling and fault diagnosis. Combined with an artificial intelligence algorithm, the established model can comprehensively reflect the influence of temperature, aging, dynamic conditions and so on, improving the accuracy of battery state estimations and optimizing battery management strategies.

Because the amount of data that can be used for battery modeling in a single electric vehicle is insufficient, the V-BMS can hardly work stably under a multivariate environment and dynamic conditions. Meanwhile, the lithium-ion battery internal state changes under impact and vibration, which have a negative impact on the operation of the V-BMS. As such, in this section, a conjunction working mode between C-BMS and V-BMS is also proposed to make better use of big data stored in the cloud. First, a black box model is established for the battery in the cloud, the SoC and terminal voltage are used as the model's output, respectively. Second, the SoC estimator in the cloud works in conjunction with the SoC estimator in the vehicle to improve the accuracy of SoC estimation. Meanwhile, the terminal voltage estimator monitors the battery in real-time and provides a data foundation for fault diagnosis.

3. Methodology

3.1. Data cleaning

3.1.1. The Support Vector Regression algorithm

Support Vector Regression (SVR) is a machine learning algorithm that adheres to the minimizing structural risks principle [26]. A kernel function is used to map data into high-dimensional space, where a line that fits all data points can be found more easily. The special loss function of SVR allows the model to have an acceptable error for the training data, and meanwhile, the slack variables are also used in SVR, which prevent the model from over-fitting. Because the SVR algorithm has such advantages, it performs better than the Back Propagation neural network, especially when the data are polluted by noise or bad points [27].

3.1.2. Missing data filling and outlier data correction

Battery data are characterized by large volume, low-value density, error. Therefore, data cleaning is an important step before battery modeling. Most statistics-based data preprocessing methods rely on the time-dependence characteristic in data or systems, but unfortunately, battery is a highly nonlinear system with strong dynamic characteristics, so the traditional statistical-based data preprocessing methods barely work effectively. Therefore, a data cleaning method for EV's big

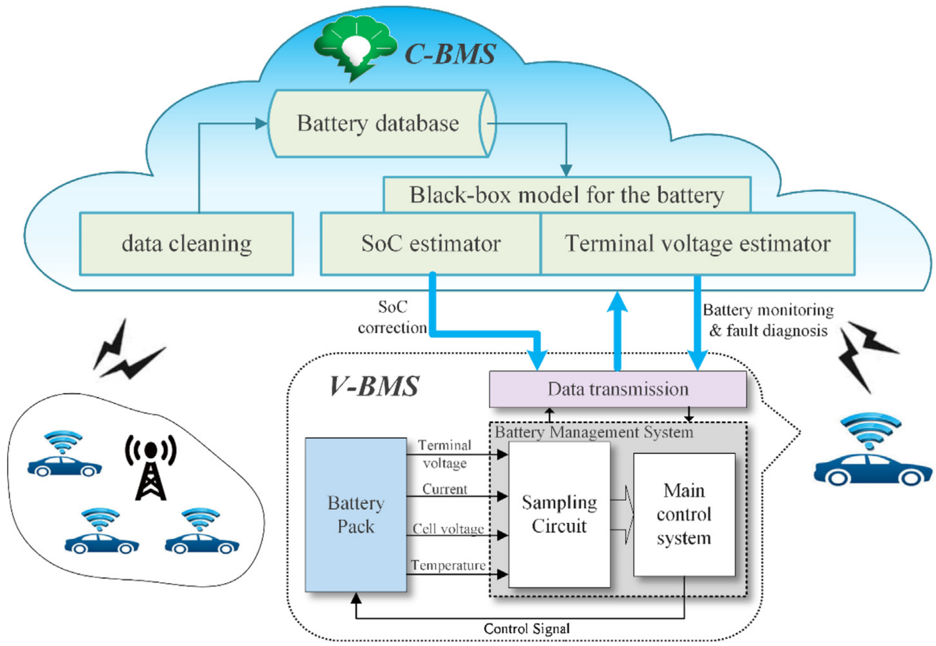

Fig. 3. The cloud-based battery management system (C-BMS).

data based on a machine learning algorithm is proposed in this section.

Aiming at filling the missing data: The terminal voltage, current (I), SoC, and temperature, at least 3 of them are required to characterize the state of the battery. However, the terminal voltage, current or temperature missing will inevitably result in SoC missing. With the absence of two at a time, the missing data cannot be filled. So the missing SoC data are filled in accordance with the time-dependence firstly. After that, a complete battery state is built, and the SVR algorithm is used to fill in the other missing data (The current, the terminal voltage or the temperature).

Aiming at correcting the outlier data: As battery data are inevitably affected by noise, sensor error, and BMS error when acquiring and transmitting, it is important to propose a method to identify the erroneous data. At present, the mainstream methods are based on statistics, but it may cause inaccurate identification of normal samples and result in information loss. Thus, this paper proposes a method to identify the outlier data based on SVR algorithm, which will improve the efficiency of data mining.

The data cleaning process is shown in Fig. 4, and the steps are as follows:

Step 1. Large amounts of data are collected for the same specification battery and a database is built in the cloud.

Step 2. The SVR model is trained by the data in the database, which contains data for different temperature and working conditions. During training, the current and terminal voltage are used as the outputs of SVR, respectively.

Step 3. A conclusion can be drawn from the time-dependence of battery data. The terminal voltage and the current change vary greatly with time, which is related to the working condition of the vehicle, while the temperature and SoC hardly change. Therefore, the missing SoC and temperature are filled in accordance with the time-dependence. The formulas based on online data and offline data are as follows:

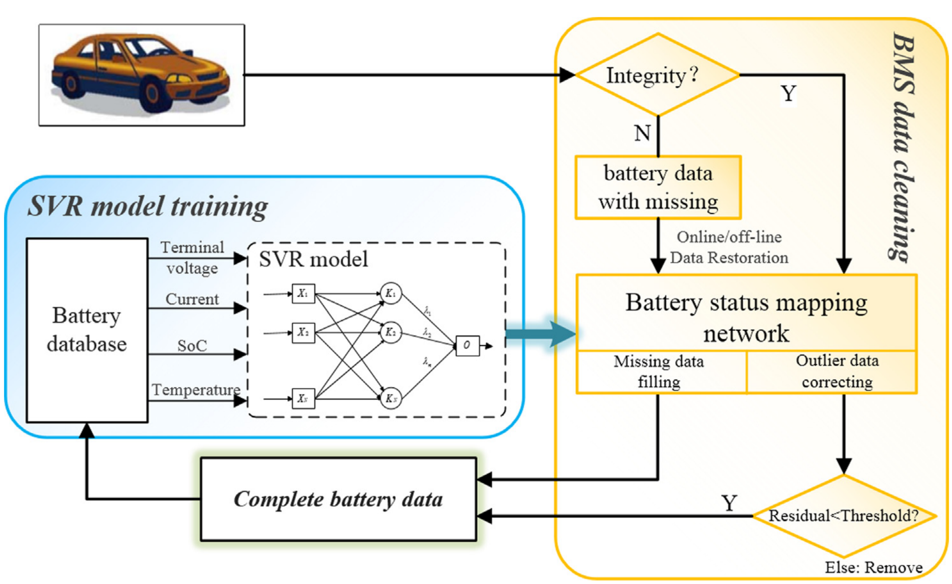

Fig. 4. The flowchart of data cleaning.

$$\begin{aligned} &online: SoC(k+1) = SoC(k) \\ &off-line: SoC(k) = \tfrac{SoC(k-1)+SoC(k+1)}{2} \end{aligned} \tag{1}$$

where k represents the current sampling point and k + 1 represents the next sampling point.

Step 4. After the missing SoC is filled, a complete battery state is built. The SVR algorithm is used to fill in other missing data (the current or terminal voltage). Then, the data are sent to the database for further mining.

Step 5. Based on the SVR model in step 2, fitting the following function and recording the residuals:

$$U = f(SoC, I, T) \tag{2}$$

By monitoring the residuals of the SVR model, the outlier data can be identified and removed from the data set to improve data quality.

3.2. Data preprocessing method for big data

3.2.1. The distribution characteristics of battery big data

An unbalanced data distribution usually exists when training the classifier, and is known as class-imbalance [28]. Ideally, the amounts of positive and negative samples are equal or similar, but if the amount of positive samples is much larger than the amount of negative samples, the class-imbalance will affect the training process [29].

As for function fitting, class-imbalance may also occur, which would affect all optimization methods based on minimizing the reconstruction error; we call this "uneven data distribution" in this paper. During training, as the trainer provides too few penalties for the data in the segment with a few data points, the training is biased, which will inevitably result in a large error for a certain fraction of data.

Data presented in this paper are collected during normal driving conditions. The data distribution is shown in Fig. 5, where most of the data accumulated in the small or medium current range. This may cause a data uneven distribution problem and will affect the battery modeling process, resulting in poor model stability under dynamic conditions. Therefore, it is necessary to propose a method to evaluate and correct the data uneven distribution.

3.2.2. The F-divergence algorithm

The trainer performs best when the data are evenly distributed, but the actual distribution is different from the ideal one, so the method for evaluating the distribution similarity can be used to evaluate the degree of data uneven distribution. There are many methods to evaluate distribution similarity, for example, the Kullback-Leibler divergence (KL-divergence) method [30], and the goodness-of-fit test method [31]. The disadvantage of most methods, however, is that the results are unbounded, which means the similarity of different data sets toward the same distribution cannot be described by comparing the magnitude of the results. Therefore, the Hollinger distance in the F-divergence algorithm is used to evaluate the distribution similarity in this section, the value of which is between 0 and 1.

For the training data, every variable $(X_1, X_2 ... X_m)$ can be divided into N groups in its range, where the value of N is related to the data's practical significance and is usually in the range of 3–6. For X_i, after it is divided into N groups, the amount of data in group i is called the segment frequency (k_i). The probability density can be calculated as:

$$P(X_i) = \begin{cases} \frac{k_1}{n} & x_1 < x \leqslant x_2 \\ \frac{k_2}{n} & x_2 < x \leqslant x_3 \\ \cdots & x_i < x \leqslant x_{i+1} \\ \frac{k_n}{n} & x_n < x \leqslant x_{n+1} \end{cases} \tag{3}$$

Assuming that the data are evenly distributed under ideal conditions, the probability density of a uniform distribution can be depicted as follows:

$$q_i = \frac{\Delta x}{x} \tag{4}$$

In statistics, the F-divergence algorithm is used to evaluate the difference between two probability densities. The F-divergence algorithm for a discrete variable can also be depicted as:

$$H(P, Q) = \sum_{x \in X} P(x) f\left(\frac{P(x)}{Q(x)}\right) \tag{5}$$

where f represents the Hellinger distance function, that is:

$$f(t) = (\sqrt{t} - 1)^2 \tag{6}$$

Substituting formula (6) into the formula (5), the obtained similarity evaluation function is bounded, and it is normalized for the range 0–1 for a simplified calculation. Then, formula (5) becomes:

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2} \tag{7}$$

where P and Q represent the actual and ideal data distributions, respectively. k represents the amount of segments, and p and q are the probability densities.

3.2.3. The sampling algorithm

The sampling method is generally used to improve the data distribution quality, and it can be divided into two types: undersampling, which eases the uneven distribution by discarding some samples; and oversampling, which alleviates the uneven distribution by generating new samples. Both are used to mitigate the data uneven distribution problem in this paper.

For the classifier, the SMOTE algorithm is used as an oversampling method. It is, however, not appropriate for regression. Therefore, Gaussian White Noise is added to the data to generate new samples.

3.3. Battery modeling method based on the deep learning algorithm

The chemical reactions in the Lithium-ion battery are very complex, and it is difficult to monitor directly, especially when considering the effects of temperature. As such, a black-box model is established in this

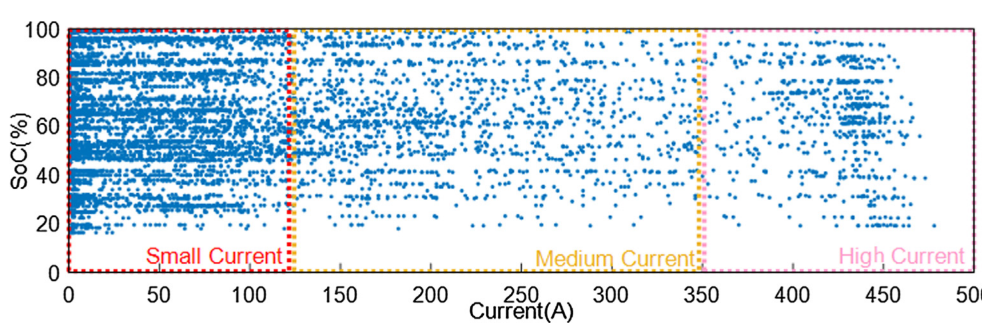

Fig. 5. The distribution of data.

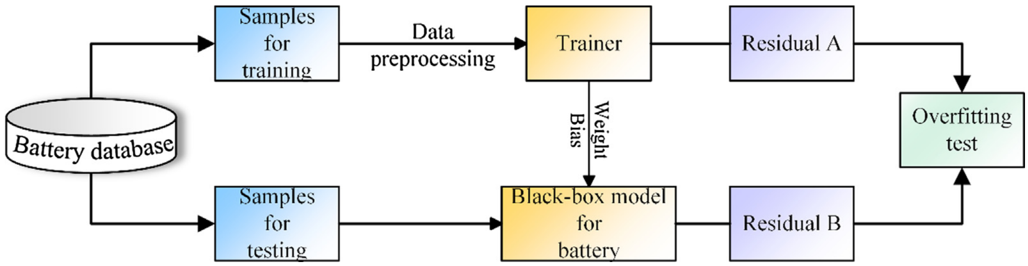

Fig. 6. Black-box model for the battery.

section. The modeling process is shown in Fig. 6.

The database is established based on the C-BMS system mentioned in Section 2, and the data preprocessing method is shown in Section 3.2. During the training, the residual A and residual B are recorded. When residual A is within the threshold, the parameters of the network (weights and bias) are assigned to the black-box model. Residual B is used to determine whether the model is over-fitting.

3.3.1. The Stacked Denoising Autoencoders (SDAE) algorithm

SDAE is a semi-supervised learning model [32]. The Automatic Encoder (AE) is the basic unit of SDAE [33], and it is a neural network with single hidden layer. The SDAE is formed by stacking multiple layers of AE from the bottom up. And by reconstructing the input data from the bottom up, the data deep structural features are continuously extracted while the effective information is retained, and the re-expressed data are easier for further mining [34].

At present, the AE algorithm is mainly used as a Principal Component Analysis (PCA) model [34]. However, it can also be used to mine the structural features and enrich the information of the data when sparsity restrictions are added to the hidden layer [35]. The relative entropy is used as sparsity restrictions in this section, and it is as depicted below:

$$KL(\rho||\hat{\rho}) = \rho\log\frac{\rho}{\hat{\rho}} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}} \tag{8}$$

where ρ represents the sparse target value and $\hat{\rho}$ represents the average activation rate of neurons, and in the case of$\rho = \hat{\rho}$, the relative entropy function value is 0.

The loss function with sparsity restrictions is:

$$J_{sparse}(W, b) = J(W, b) + \beta\cdot KL(\rho||\hat{\rho}) \tag{9}$$

where W and b are the weights and bias of the network, respectively, $J_{sparse}(W, b)$ is the loss function.

The Denoising Autoencoder (DAE) trains the AE with the polluted data. Therefore, the DAE model not only filters out the noise but also extracts structural features of the data. In this paper, Gaussian White Noise is added to the bottom layer of SDAE to avoid excessive damage to input data. For the hidden layer, the noise intensity cannot be controlled as the neuron value is rather small under sparsity restrictions, so noise is added by randomly assigning 0 for neurons. The basic unit of the SDAE model is shown in Fig. 7.

3.3.2. The Extreme learning Machine algorithm

The Extreme Learning Machine (ELM) is a type of fast learning algorithm and is a neural network with a single hidden layer [36]. Different from the BP neural network, the weights and biases of the hidden layer in the ELM are determined randomly, and the output layer has only weights but no bias. During training, the weights of the output layer are not solved by the gradient-based algorithm, but are transformed into a linear system for solving [37].

The ELM's first layer function is to randomly map the original input into the high-dimensional space, where a line that fits all data points can be more easily found with low error. When ELM is used alone as a predictor or classifier, the first layer is indispensable, but the role is fungible when ELM is combined with other algorithms. As such, in this section, only the second layer of ELM is used.

Given that the amount of neurons in the hidden layer is L, the hidden layer input is X, the weights between the hidden layer and the output layer are β, and assuming that there are N samples, then the output can be depicted as:

$$\sum_{i=1}^{L}\beta_i x_{ij} = o_j \tag{10}$$

The purpose of the trainer is to find the specific β to minimize the output error, depicted as:

$$\min_{\beta}||X\beta - Tu|| \tag{11}$$

where Tu represents the expected output of the model.

The above equation can be converted into finding the least-squares solution for a linear system, and in this way, the complex iteration process can be omitted and the weight matrix can be directly obtained as follows:

$$\hat{\beta} = X^{+}Tu \tag{12}$$

where X^{+} is the generalized inverse matrix of$X$.

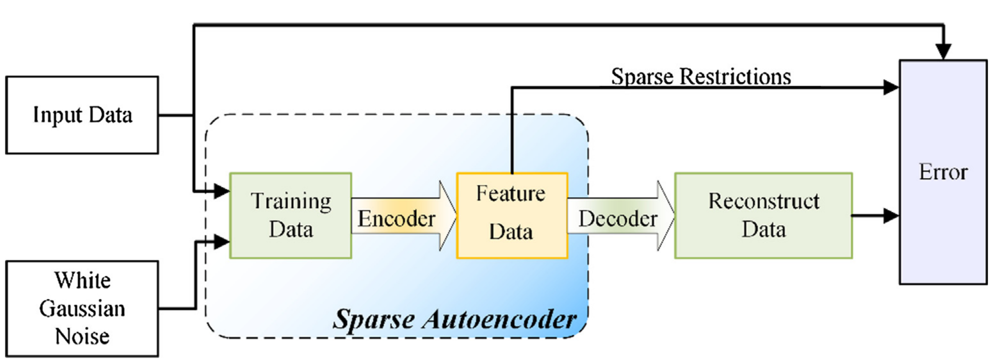

Fig. 7. The basic unit of the SDAE model.

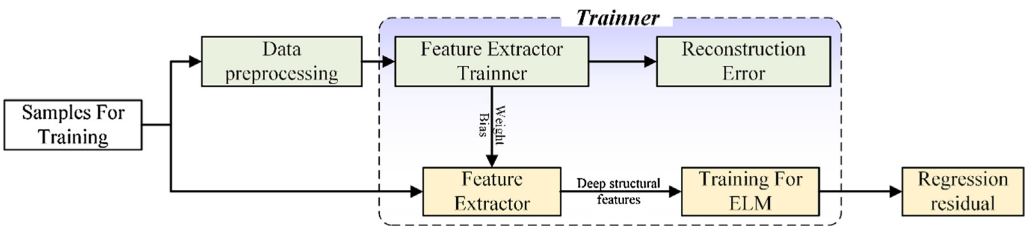

Fig. 8. The new training method for the SDAE-ELM model.

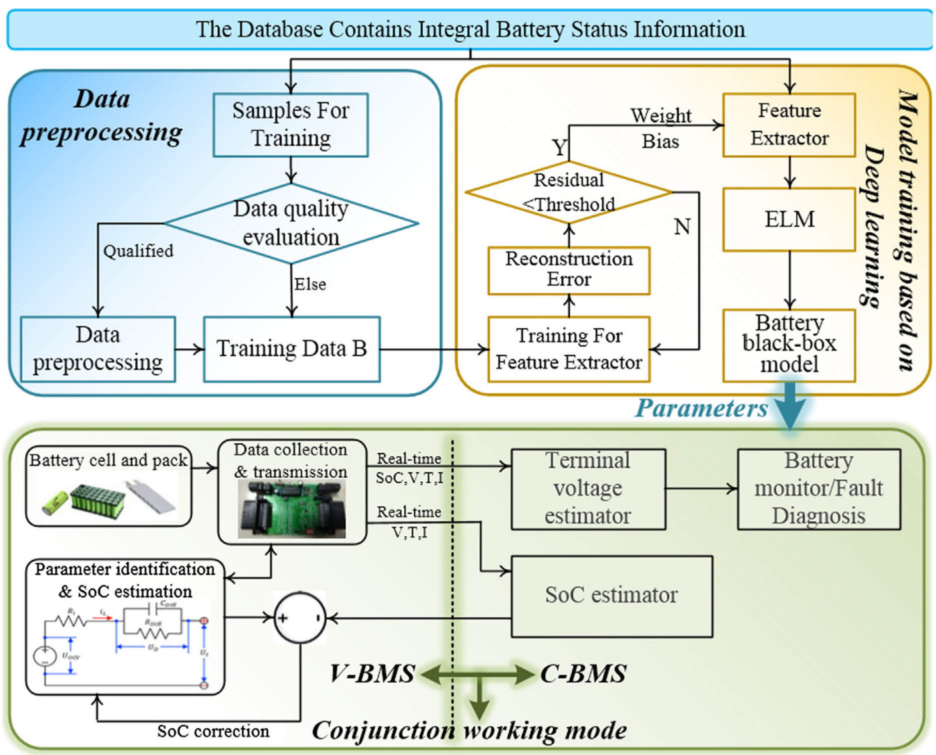

Fig. 9. The conjunction working mode between C-BMS and V-BMS.

3.3.3. The new training method for SDAE-ELM model

SDAE-ELM is a commonly used algorithm in deep learning. The SDAE model is trained unsupervised from the bottom-up by unlabeled data, and it is used as a feature extractor. The output of the SDAE model is used as the ELM's input; afterward, the ELM is trained directly by labeled data and is used as the output layer of the SDAE-ELM model. However, for the data uneven distribution problem in the big data environment, the SDAE-ELM model training method mentioned above does not perform well, so a new training method combined with data preprocessing is proposed in this section.

The training process is illustrated in Fig. 8. The SDAE trainer performs best when the data are evenly distributed. Thus, SDAE was trained by the preprocessed data. For ELM, since its weights are directly obtained by the pseudo-inverse method, it will be less influenced by the data uneven distribution problem, but its error increases significantly as the data quality decreases, so ELM is trained by raw data in this section.

3.3.4. Model development

To make better use of the cloud-based battery model, based on the C-BMS mentioned in Section 2 and the battery modeling method mentioned in Sections 3.2–3.3, a conjunction working mode between C-BMS and V-BMS is proposed in this section. As shown in Fig. 9, the system mainly consists of two parts: An online-data-driven lithium-ion battery monitoring method and a big data driven adaptive SoC estimation method.

The online-data-driven lithium-ion battery monitoring method consists of four steps:

Step 1. Based on the C-BMS, a database that contains integral battery status information is established to provide a good data foundation for the battery modeling process. Additionally, in order to monitor the EV's battery online, the battery data are uploaded in real-time.

Step 2. Based on the data distribution evaluating and preprocessing method presented in Section 3.2, the data are preprocessed to improve the distribution quality.

Step 3. The battery is modeled with SDAE-ELM algorithm. The battery terminal voltage is used as the output to establish a terminal voltage estimator. To establish a temperature-dependent battery model that can work stably in a multi-variable environment, the temperature is also used as an input. In other words, the purpose of the neural network is to fit the function:

$$U = f(SoC, I, T) \tag{13}$$

Step 4. The battery state could be monitored by comparing the model residual for the training data and the real-time data. If the residual increases rapidly, the battery is in an abnormal state, and the battery fault information is sent to the BMS in time to prevent further battery damage. In addition, the error of the model can be further used to mine the fault information of the battery and its management system and provide a data foundation for fault diagnosis.

The big data driven adaptive SoC estimation method also consists of four steps:

Steps 1 to 3 are the same as those listed above for the battery

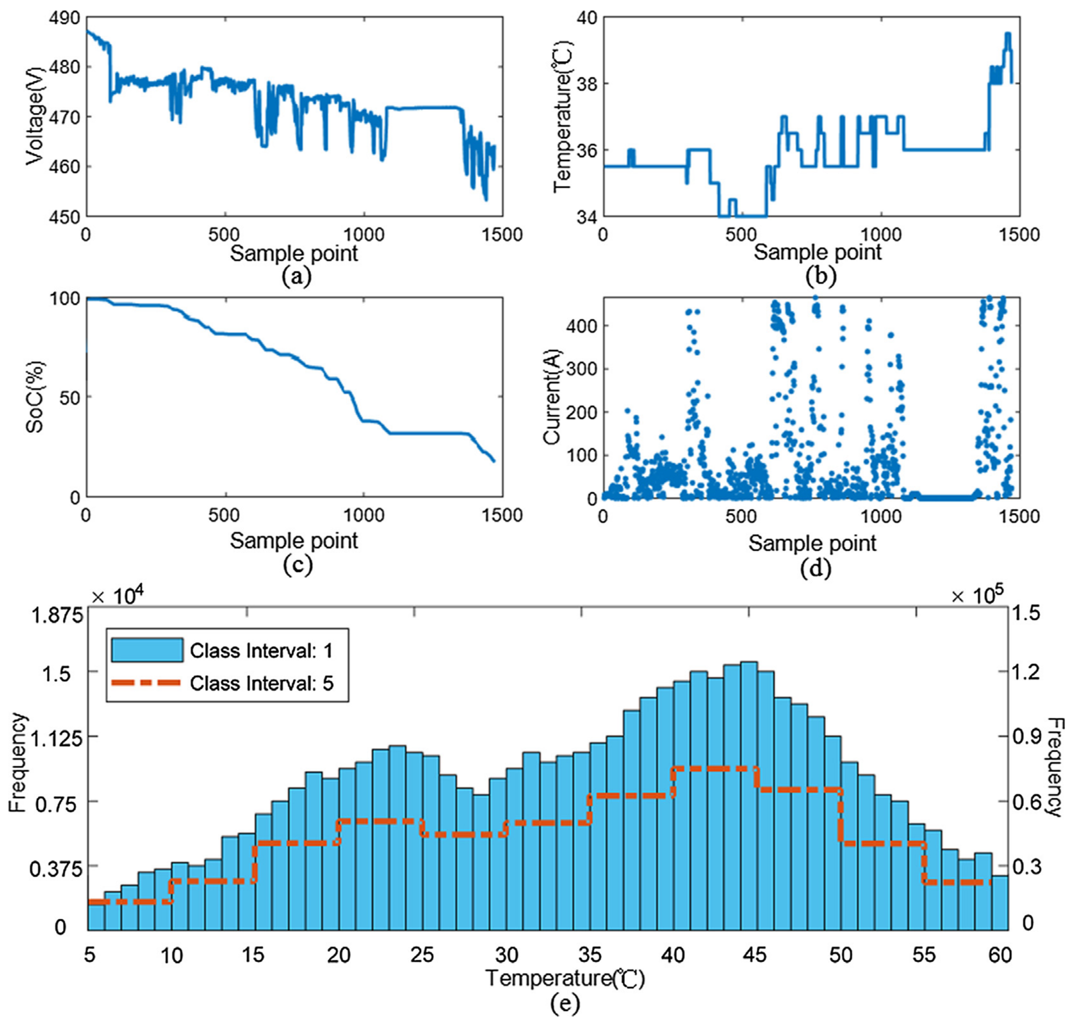

Fig. 10. The data used in this paper. (a) Voltage data in one discharge cycle. (b) Temperature data in one discharge cycle. (c) SoC data in one discharge cycle. (d) Current data in one discharge cycle. (e) Temperature distribution in the whole data set.

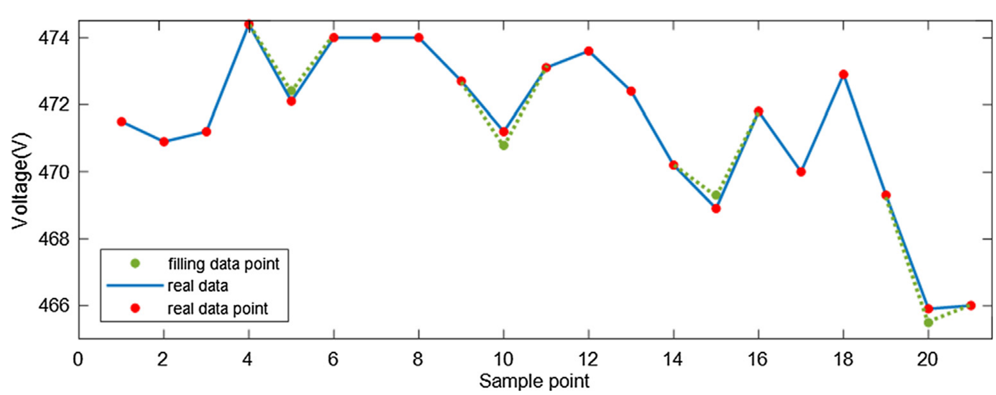

Fig. 11. The result of missing data filling based on SVR algorithm.

Table 1
The principle of current division based on operating conditions.

Operating conditions	Parked	Normal driving	Starting& Accelerating
Current I (A)	$0 < I \le 5$ $5 < I \le 10$	$10 < I \le 100$ $100 < I \le 200$ $200 < I \le 300$	$300 < I$

monitoring method. However, the SoC of the battery is used as an output to establish an SoC estimator. In other words, the purpose of the neural network is to fit the function:

$$SoC = f(U, I, T) \tag{14}$$

The current, terminal voltage, SoC, and temperature parameters are sufficient to characterize the battery state. For the battery monitoring system mentioned above, the terminal voltage is the model output, which is greatly affected by the current and has a weak time-dependence. However, as for the SoC estimator, the model output is SoC,

Table 2
Data distribution before preprocessing. (H = 0.4833).

Operating conditions	Current I (A)	Number of samples	p_i	q_i
Stop	0 < I ≤ 5	193,750	0.3875	0.0107
	5 < I ≤ 10	26,550	0.0531	0.0107
	10 < I ≤ 100	155,100	0.3102	0.1915
Normal driving	100 < I ≤ 200	56,700	0.1134	0.2128
	200 < I ≤ 300	27,550	0.0551	0.2128
Accelerate	300 < I	40,400	0.0808	0.3617

Table 3
Data distribution after preprocessing. (H = 0.3042).

Operating conditions	Current I (A)	Number of samples	p_i	q_i
Stop	0 < I ≤ 5	54,559	0.1493	0.0107
	5 < I ≤ 10	44,838	0.1227	0.0107
	10 < I ≤ 100	87,338	0.239	0.1915
Normal driving	100 < I ≤ 200	63,878	0.1748	0.2128
	200 < I ≤ 300	46,556	0.1274	0.2128
Accelerate	300 < I	68,263	0.1868	0.3617

which has a strong time-dependence; that is, the SoC at the next sample point has a direct relationship to the previous point. Therefore, in order to improve the SoC estimator accuracy, the previous SoC value is also used as the model input. In other words, the following recursive relationship can be established in the SoC estimator:

$$SoC_t = f(U_t, I_t, T_t, SoC_{t-1}) \tag{15}$$

Step 4. The V-BMS monitors and directly controls the battery. The equivalent circuit model and the least squares algorithm or the Kalman filter algorithm are used to model the battery and estimate the SoC. At the same time, the C-BMS builds a big data driven SoC estimator that can work stably in a multi-variable environment and dynamic conditions in the cloud. However, due to the data exchange delay between the V-BMS and the C-BMS, the SoC estimator in the C-BMS cannot be directly used in battery management or vehicle energy management. Therefore, in this step, the cloud-based SoC estimator is used to work in conjunction with the V-BMS. The V-BMS uploads the battery data to the cloud by the data transmission module; meanwhile, the cloud-based SoC estimator accurately estimates the SoC. By calculating the difference in the estimation results between the cloud-based SoC estimator and V-BMS SoC estimator, an error-feedback signal is formed. Based on the error-feedback signal, on one hand, the V-BMS adaptively modifies the equivalent circuit model parameters, and on the other hand, it adaptively corrects the super-parameter in the Kalman filter.

4. Result and discussion

4.1. Simulation environment

To obtain experimental data for verifying the effectiveness of the battery modeling method, the C-BMS mentioned in Section 2 was installed in several electric buses in Beijing, China, and continuously collected battery data. Specifically, the temperature sensors are installed in each battery pack to get the temperature respectively, but for simplifying the model, the average temperature of all the battery pack is calculated and used to describe battery temperature. Approximately 500,000 sets of data under different temperature conditions were collected, which include the Terminal voltage, the SoC, the temperature and the current. Fig. 10(a–d) shows the data in one discharge cycle.

The temperature distribution of the data is mainly in the range of 5–60 °C, as shown in Fig. 10(e). The group frequency is calculated with a group distance of 1 °C and 5 °C respectively. Due to the geographical location, temperatures below 5 °C and above 60 °C are not registered.

The ELM, BP, and SDAE-ELM algorithm are used in our work, and these are the optimized parameters obtained for all the algorithm:

$$Training\ set\ size \in \{450000\} points\ of\ data$$
$$Validation\ set\ size \in \{50000\} points\ of\ data$$

$$Parameter\ of\ BP\ network$$
$$\begin{cases} neuron\ number \in \{3, 50, 1\} \\ Training\ batch\ size \in \{5000\} points\ of\ data \\ Training\ method \in \{\text{Bayesian regularization}\} \\ Neuron\ cell\ unit \in \{Artificial\ Neural\ Network\} \\ Loss\ function \in \{Root\ Mean\ Square\ Error\} \end{cases}$$

$$Parameter\ of\ \text{SDAE - ELM}$$
$$\begin{cases} SDAE\ neuron\ number \\ \quad \in \{3, 20, 20, 50, 50, 100, 100\} \\ \text{SDAE - ELM}\ neuron\ number \\ \quad \in \{3, 20, 20, 50, 50, 100, 100, 1\} \\ Training\ batch\ size \in \{5000\} points\ of\ data \\ Neuron\ cell\ unit \in \{SDAE, ELM\} \\ Training\ method \in \{Gradient\ descent, Pseudo \\ \quad -inverse\} \\ Loss\ function \in \{Root\ Mean\ Square\ Error\} \end{cases}$$

4.2. Data cleaning result

The data set in Section 4.1 is used for model testing. However, limited by the calculating ability of the simulation environment, only a fraction of the data are selected for model verification. Approximately

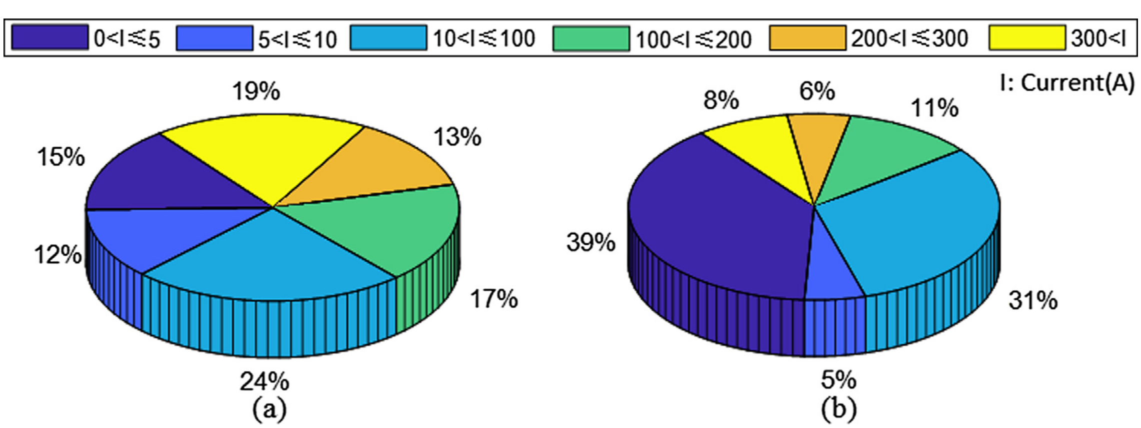

Fig. 12. Comparison of data distribution before and after data preprocessing. (a) Data distribution before preprocessing. (b) Data distribution after preprocessing.

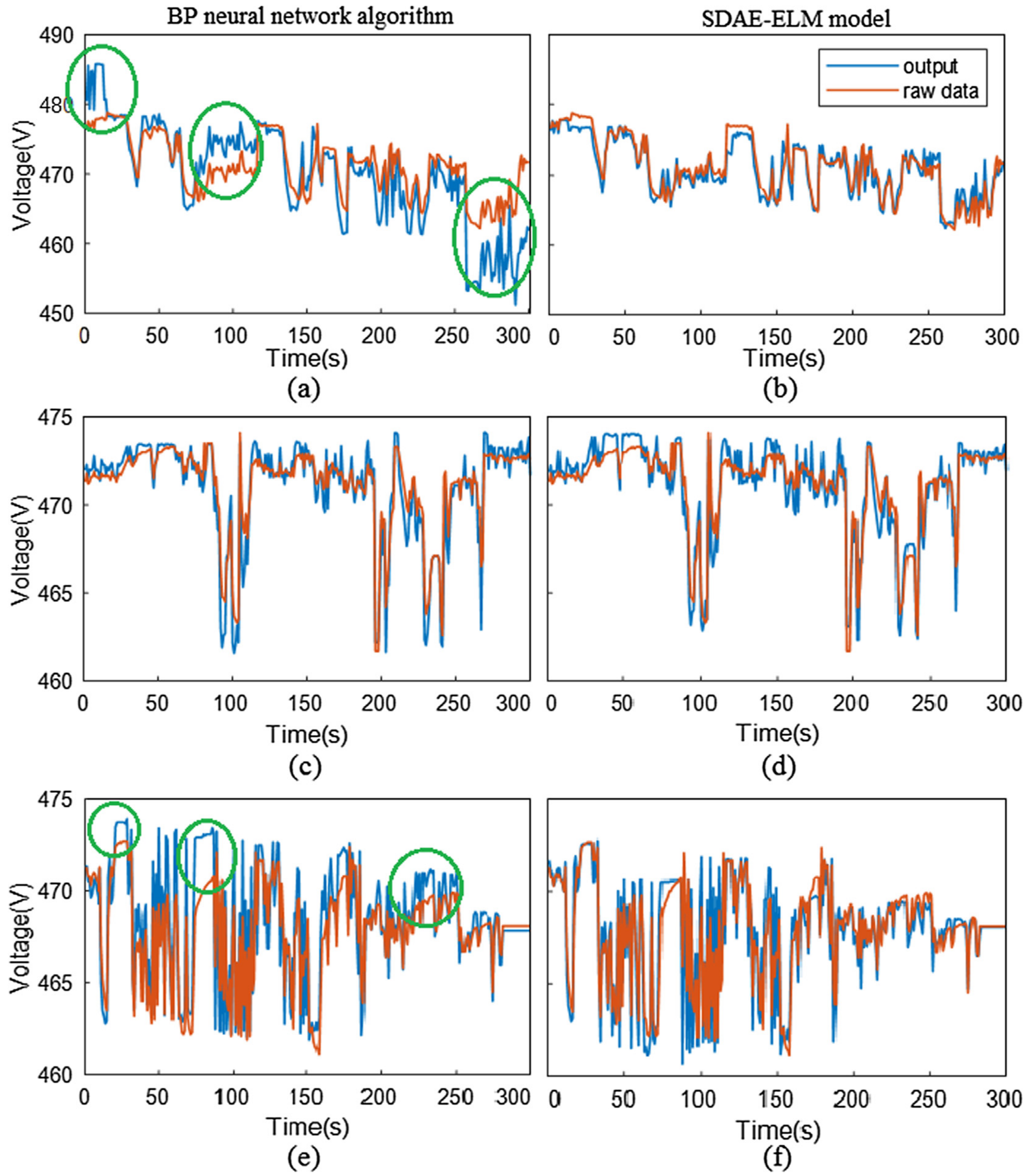

Fig. 13. The output of the black-box model for batteries at different SoC levels. (a) BP algorithm with SoC: 70–100%. (b) SDAE-ELM algorithm with SoC: 70–100%. (c) BP algorithm with SoC: 40–70%. (d) SDAE-ELM algorithm with SoC: 40–70%. (e) BP algorithm with SoC: 5–40%. (f) SDAE-ELM algorithm with SoC: 5–40%.

30,000 sets of data are used as training data and 5000 sets of data are used as test-data in this section.

In the experiment, the sampling frequency is 1 s, the data loss ratio is 1/5, and the missing variables are terminal voltage and SoC.

As shown in Fig. 11, in the experiment, due to SVR's higher robustness, the maximum relative error of the proposed SVR algorithm based missing data filling method is within 3%.

4.3. Data preprocessing result

There are three input variables in the battery model: SoC, temperature, and current. In this section, the current data uneven distribution problem is addressed. Before preprocessing, the current data should be divided into several groups. In this paper, according to the driver's urgency degree for accelerating and subjective demand for engine power, we divide the current data into three categories, namely "Parked", "Normal driving" and "Starting & Accelerating". According to the aforementioned rules, more detailed, the data can be divided into six groups, as shown in Table 1.

The data quality assessment and preprocessing methods are shown in Sections 3.2.2 and 3.2.3. Table 2 shows the amount of data in each group before preprocessing, and the unevenness degree is H = 0.4833. Table 3 shows the amount of data in each group after preprocessing, and the unevenness degree is H = 0.3042. Fig. 12 shows the comparison of data distribution before and after the data pre-processing. Apparently, the sampling algorithm can greatly improve the quality of the data distribution. However, undersampling results in information loss and oversampling results in serious over-fitting, so there is no strict requirement for the H value in this paper.

4.4. Battery modeling result

The battery is modeled based on the method described in Sections 3.2 and 3.3, and the experiment data from Section 4.1 are used to verify the model accuracy.

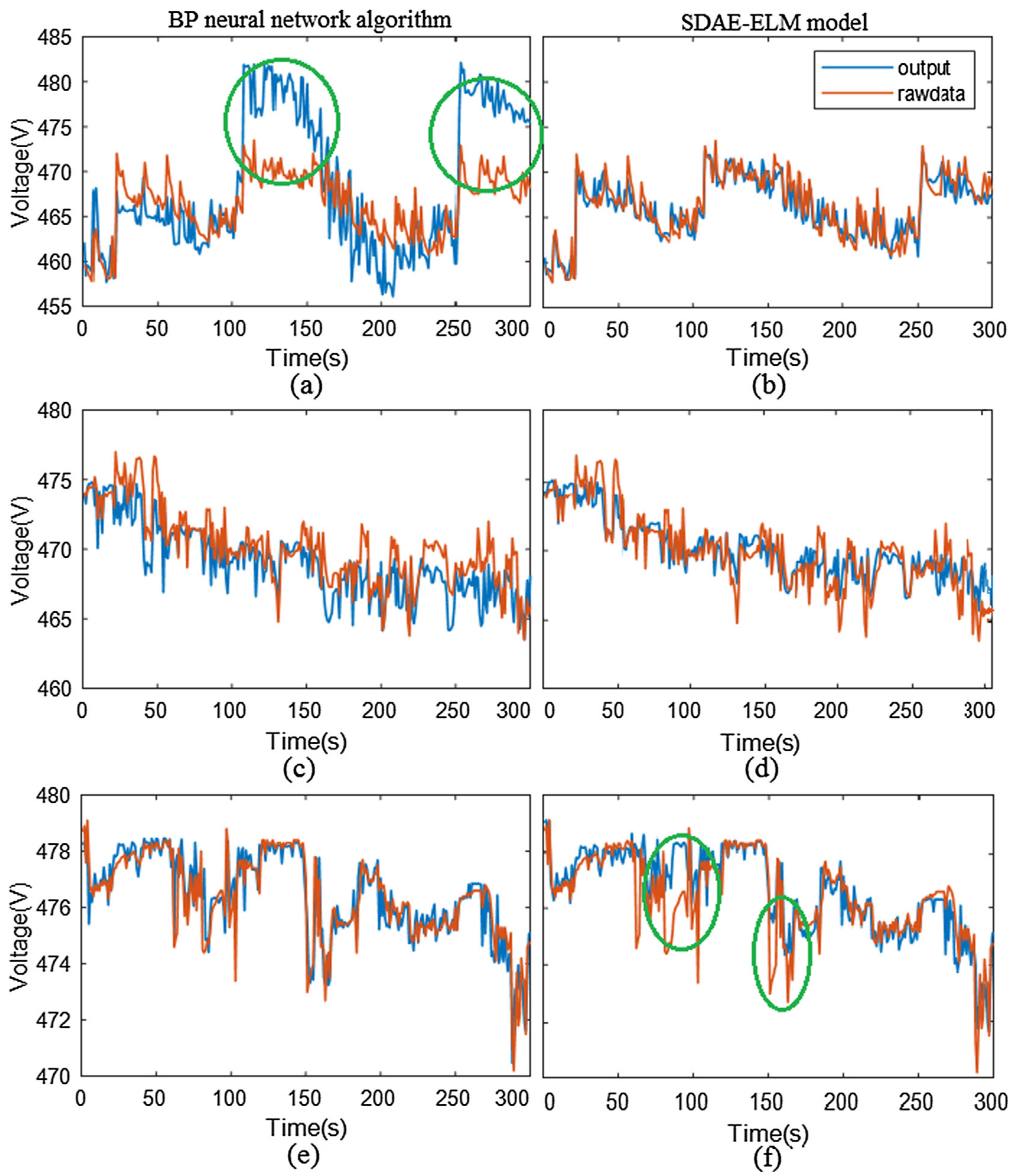

Fig. 14. The output of the black-box model for batteries in different current ranges. (a) BP algorithm with Current: 300–500 A. (b) SDAE-ELM algorithm with Current: 300–500 A. (c) BP algorithm with Current: 100–300 A. (d) SDAE-ELM algorithm with Current: 100–300 A. (e) BP algorithm with Current: 0–100 A. (f) SDAE-ELM algorithm with Current: 0–100 A.

Table 4
The model error under different temperature.

Temperature(°C)	TIM MAPE (%)	TDM MAPE (%)
$5 \le T < 10$	5.18	3.01
$10 \le T < 20$	3.54	2.54
$20 \le T < 30$	2.55	2.11
$30 \le T < 40$	2.58	2.05
$40 \le T < 50$	2.42	1.93
$50 \le T < 60$	2.77	2.46

4.4.1. The result of Online-data-driven lithium-ion battery monitor

The key of the online-data-driven lithium-ion battery monitoring method is to establish a terminal voltage estimator. Therefore, in this section, this paper establishes the battery terminal voltage estimator based on different algorithms and analyzes the influences of temperature, current, SoC and other factors on the model.

The different current/SoC intervals own different data density:

SoC intervals: During electric buses normal driving, limited by charging conditions, electric buses are less likely to be in high SoC (70–100%). Meanwhile, for battery life protection, there are few EVs travel in low SoC conditions. So most electric buses travel in medium SoC (40–70%).

Current intervals: The battery discharging current data are divided into three states, large current state (300–500 A), medium current state (100–300 A) and low current state (0–100 A). During electric buses normal driving, there are few starting or accelerating conditions, so the period that battery discharging in medium current state are far more than that in the other two states.

To analyze the effect of the model under different data density, the performance of BP neural network and SDAE-ELM algorithm in different current/SoC intervals (300–500 A, 100–300 A, 0–100 A; 70–100%, 40–70%, 5–40%) are chosen to be compared in our work, as shown in Fig. 13 and Fig. 14, respectively. Subfigure (a), (c) and (e) are

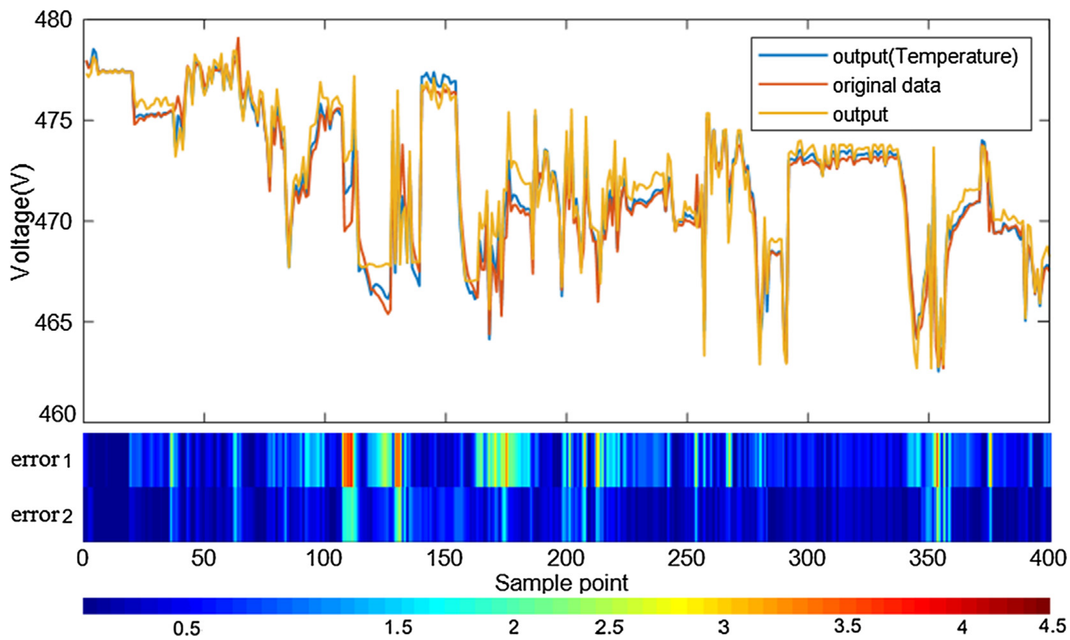

Fig. 15. The comparison between the temperature-dependent and temperature-independent model: (1) Temperature-independent model, (2) Temperature-dependent model.

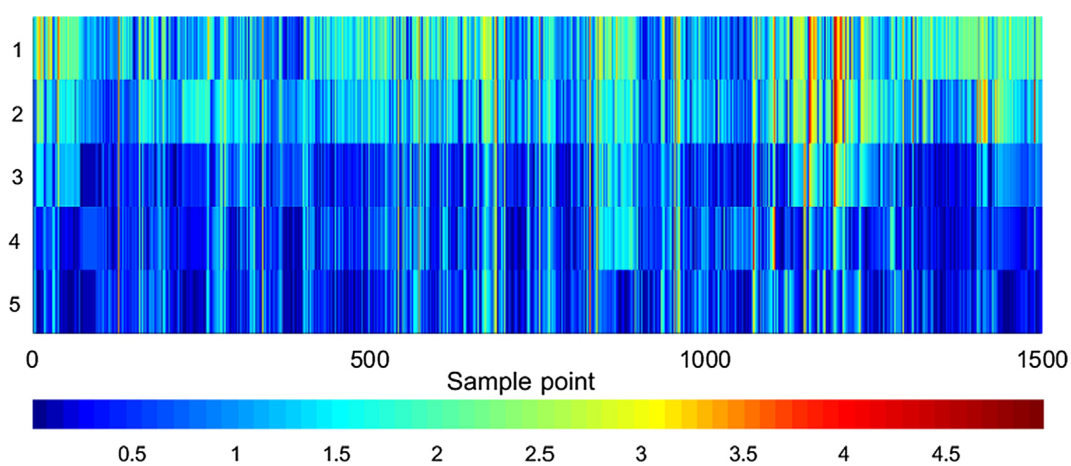

Fig. 16. The error of five different algorithms: (1) BP neural network, (2) ELM, (3) SDAE + ELM, (4) Data preprocessing + SDAE-ELM, (5) Data preprocessing combined with SDAE-ELM.

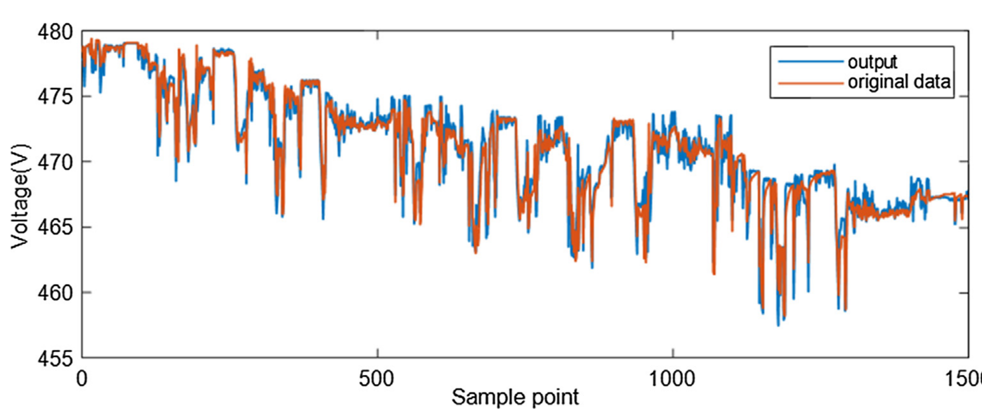

Fig. 17. The accuracy of the proposed model in a complete discharge cycle.

Table 5
The error of different algorithms.

Model	MAPE(%)	Std
BP neural network	4.45	2.3782
ELM	4.15	2.6414
SDAE + ELM	3.45	1.7283
Data pre-processing + SDAE-ELM	2.62	1.5452
Data pre-processing combine with SDAE-ELM	2.07	1.1394

the results of the BP neural network algorithm, and (b), (d) and (f) are the results of the SDAE-ELM model.

Fig. 13(a) and 13(b) are the estimation results when SoC is between 70% and 100%; it is apparent that the accuracy of the SDAE-ELM model is much higher than that of the BP neural network. In some intervals, the BP neural network model is even divergent. When SoC is between 5% and 40%, as shown in Fig. 13(e) and 13(f), the accuracy of the SDAE model is also slightly higher than in the BP neural network. Due to the lack of training data at high and low SoC levels, the BP neural network model is not adequately trained in both situations, which results in a larger error. The SDAE-ELM model can fully exploit the deep structural features in the data, which effectively improves the model accuracy. In Fig. 13(c) and 13(d), when SoC is between 40% and 70%, both models perform well, as they can be fully trained with large amounts of data.

Fig. 14(a) and 14(b) are the results for when the current is between 300 and 500 A. For the sake of that the SDAE-ELM model and the oversampling algorithm effectively boosted the model robustness, the SDAE-ELM model is also much more accurate than that of BP neural network. However, when the current is between 0 and 100 A in Fig. 14(e) and 14(f), since the undersampling algorithm would result in information loss, the SDAE-ELM model performed slightly worse than BP neural network. Fig. 14(c) and Fig. 14(d) show the performance of the model when the current is between 0 and 100 A, both models perform well with the support of large amounts of data. In other words, the modeling method proposed in this paper improves the model accuracy in the high current range by sacrificing it in the low current range. As long as its error in the low current range is acceptable, the SDAE-ELM model could be regarded as an effective method to replace the BP neural network.

All in all, the SDAE model and the new training method proposed in this paper obviously improve model accuracy and adaptability under dynamic conditions, especially in the case of high current and low SoC discharging.

To illustrate the effect of temperature, we compare the model performance under different temperatures in Table 4, the model accuracy is reflected by the Mean Absolute Percent Error (MAPE). The terminal voltage estimation error of the temperature-independent model (TIM) is generally higher than that of temperature-dependent model (TDM), especially when the temperature is low ($5 \le T < 10$) and high ($50 \le T < 60$). The reason is that the battery model is not able to reflect the change in temperature, but the battery electromotive force decreases with temperature. Temperature-dependent battery model is able to deeply explore the battery data under different temperature interval, so its accuracy is satisfactory in all temperature range.

Fig. 15 shows the detailed effect of temperature on battery modeling. The battery temperature is 10–15 °C, and it is apparent that when the temperature is used as an additional input, the model accuracy improves significantly. When the model input is only the SoC and the current, the terminal voltage estimation result is generally higher than the accurate value and the maximum error is up to 5%. When the temperature is used as an additional input, the model accuracy is significantly improved, and the maximum error is within 2.5%.

This paper also evaluated the results of several different modeling algorithms. The error of each is shown in Fig. 16, and the accuracy of the SDAE-ELM model in a complete discharge cycle is shown in Fig. 17.

As shown in Table 5, in which the accuracy of different algorithm is compared and evaluated by the MAPE and Standard Deviation (Std). The performances of the BP neural network and ELM are not satisfying, but in terms of computing speed, the efficiency of the ELM algorithm is much higher than that of the BP neural network. In contrast, the SDAE-ELM model can extract deep structural features in data, improving model accuracy and stability, and the improvement is much more remarkable when the SDAE-ELM model is trained by the preprocessed data. Additionally, the new training method combined with data preprocessing for the SDAE-ELM model proposed in this paper can effectively control the impact of noise on the model while improving the quality of the data distribution, which can significantly improve the model adaptability under dynamic conditions.

Based on the discussion above, the big data driven terminal voltage estimator established in this paper can work stably in a multi-variable environment and under dynamic conditions, which can effectively monitor the battery state and provide a good data foundation for fault diagnosis.

4.4.2. *The result of cloud-based SoC estimator*

The key of the big data driven adaptive SoC estimation method is to establish a cloud-based SoC estimator. Therefore, in this section, this paper establishes a big data driven SoC estimator based on the modeling method described in Section 3.3.4. The training process is the same as that for the terminal voltage estimator, so in order to simplify the article content, only the SDAE-ELM model and the new training method are used in this section.

Fig. 18 illustrates the results of cloud-based SoC estimator, wherein subfigure (a) demonstrates the result of SoC estimator with only current, terminal voltage, and temperature as inputs, and subfigure (b) demonstrates the result of SoC estimator with SoC_{t-1} as an additional input. It is apparent that both kinds of SoC estimators can work stably, but the model accuracy is significantly improved after SoC_{t-1} was

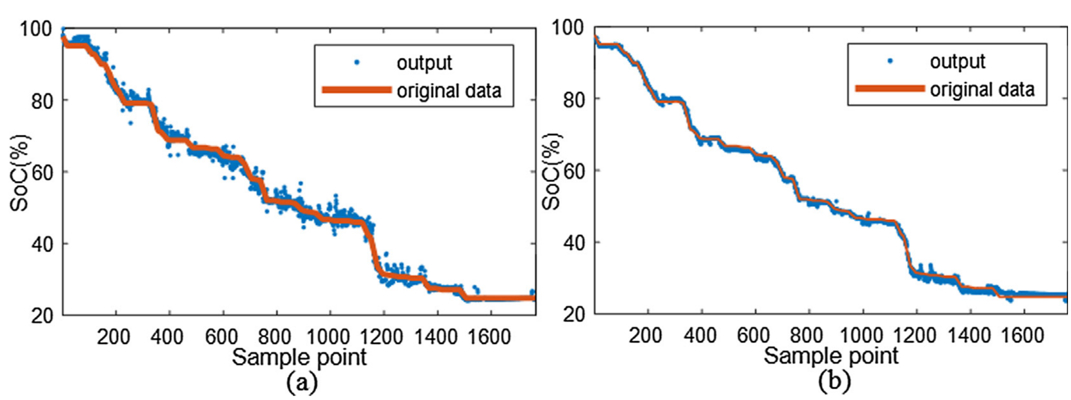

Fig. 18. The result of the cloud-based SoC estimator. (a) Model performance without SoC_{t-1} as input. (b) Model performance with SoC_{t-1} as input.

introduced as an additional input. When SoC_{t-1} is not used as an input, the mean absolute error is within 4%, but the maximum relative error is up to 7%. After SoC_{t-1} was introduced as model input, the mean absolute error is reduced to 1%, and the maximum relative error is also limited to within 3%.

The cloud-based SoC estimator can accurately estimate SoC in a multi-variable environment and under dynamic conditions, which provides a data foundation for adaptively modifying the parameters of an equivalent circuit model and a Kalman filter algorithm in V-BMS.

5. Conclusions and future work

This paper presents a big data driven lithium-ion battery model that has the ability to work stably in a multi-variable environment and dynamic conditions based on the deep learning algorithm.

(1) The proposed C-BMS is able to effectively deal with big data resources and can reduce the calculation burden of the V-BMS. It provides a multidimensional, multistate, multifactor complex data space for battery modeling and the fault diagnosis process. The conjunction working mode between C-BMS and V-BMS provides a platform for future research on intelligent management of batteries.
(2) By using the data cleaning method based on the machine learning algorithm, the maximum relative data filling errors are within 3%, which indicates great improvement for the overall quality of the data set. The proposed data quality assessment method based on the F-divergence algorithm can be effectively used for analyzing the data uneven distribution problem, and the data preprocessing method based on the sampling algorithm improves battery model accuracy and stability under dynamic conditions.
(3) By using the new training method combined with data preprocessing for the SDAE model proposed in this paper, the trained battery model achieves great accuracy and stability. The results show that the errors of the terminal voltage and the SoC estimators are within 2% and 3%, respectively. The model is able to achieve a high-precision simulation for the battery dynamic characteristics in a multi-variable environment and under dynamic conditions, and it provides a good data foundation for further fault diagnosis and adaptive SoC estimation.

This paper mainly focuses on the data collection method and cloud-based battery modeling method. Future work can be conducted on the following two aspects.

(1) Limited by experimental conditions, the established battery model has not been tested for wide ranges of temperatures which may comprise different electrochemical kinetics regimes. We have already begun to obtain more accurate and comprehensive data set by changing the experimental location and adding temperature sensors and we will verify the model through more sufficient data set in our future works.
(2) The data collection and storage process should be improved, the benefits gained from the data quality improvement are much greater than that of algorithms, and a high-quality data set is the basis for precise modeling.

Acknowledgments

This work is supported by the National Nature Science Foundation of China (No. U1864202) and the National Key R&D Program of China (No. 2017YFB0103802) in part.

References

[1] Zhang Y, Xiong R, He H, Shen W. Lithium-ion battery pack state of charge and state of energy estimation algorithms using a hardware-in-the-loop validation. Trans Power Electron 2017;32:4421–31.
[2] Lin C, Mu H, Xiong R, Shen W. A novel multi-model probability battery state of charge estimation approach for electric vehicles using H-infinity algorithm. Appl Energy 2016;166:76–83.
[3] He H, Zhang Y, Xiong R, Wang C. A novel Gaussian model based battery state estimation approach: state-of-energy. Appl Energy 2015;151:41–8.
[4] Rahimi-Eichi H, Baronti F, Chow M. Online adaptive parameter identification and state-of-charge coestimation for lithium-polymer battery cells. IEEE Trans Ind Electron 2013;61:2053–61.
[5] He H, Xiong R, Peng J. Real-time estimation of battery state-of-charge with unscented Kalman filter and RTOS μCOS-II platform. Appl Energy 2016;162:1410–8.
[6] Dai H, Wei X, Sun Z, Wang J, Gu W. Online cell SOC estimation of Li-ion battery packs using a dual time-scale Kalman filtering for EV applications. Appl Energy 2012;95:227–37.
[7] Sangwan V, Sharma A, Kumar R, Rathore A. Equivalent circuit model parameters estimation of Li-ion battery: C-rate, SOC and temperature effects. IEEE international conference on power electronics, drives and energy systems. 2017.
[8] Li J, Yang Q, Robinson F, Liang F, Zhang M, Yuan W. Design and test of a new droop control algorithm for a SMES/battery hybrid energy storage system. Appl Energy 2017;118:1110–22.
[9] Liu Y, He Z, Gao M, Li Y, Liu G. Dual estimation of lithium-ion battery internal resistance and SOC based on the UKF. International congress on image and signal processing. 2013. p. 1639–43.
[10] Yao H, Cao C, Liu X, Zheng X, Zeng G. Control. SOC estimation method for lithium battery based on variable temperature model. Electric Mach 2018;22:43–52.
[11] Du J, Liu Z, Wang Y, Wen C. A fuzzy logic-based model for Li-ion battery with SOC and temperature effect. IEEE international conference on control & automation. 2014. p. 1333–8.
[12] Li J, Xiong R, Mu H, Cornélusse B, Vanderbemden P, Ernst D, et al. Design and real-time test of a hybrid energy storage system in the microgrid with the benefit of improving the battery lifetime. Appl Energy 2018;218:470–8.
[13] Li J, Xiong R, Yang Q, Liang F, Zhang M, Yuan W. Design/test of a hybrid energy storage system for primary frequency control using a dynamic droop method in an isolated microgrid power system. Appl Energy 2016;201:257–69.
[14] Wang Y, Liu C, Pan R, Chen Z. Experimental data of lithium-ion battery and ultracapacitor under DST and UDDS profiles at room temperature. Data in Brief 2017;12:161–3.
[15] Sun B, Jiang J, Wang Z. SOC Estimation of Ni-MH Battery Pack based on Approved HPPC Test and EKF Algorithm for HEV. Trans Tech Publ 2012;403:4398–402.
[16] Benedetti M, Cesarotti V, Introna V, Serranti J. Energy consumption control automation using Artificial Neural Networks and adaptive algorithms: proposal of a new methodology and case study. Appl Energy 2016;165:60–71.
[17] Chaoui H, Ibe-Ekeocha CC, El Mejdoubi A, Oukaour A, Gualous H, Omar N. State of charge estimation of LiFePO 4 batteries with temperature variations using neural networks. Industrial Electronics (ISIE), 2016 IEEE 25th international symposium on. IEEE; 2016. p. 286–91.
[18] Kang L, Zhao X, Ma J. A new neural network model for the state-of-charge estimation in the battery degradation process. Appl Energy 2014;121:20–7.
[19] Zhao Y, Liu P, Wang Z, Zhang L, Hong J. Fault and defect diagnosis of battery for electric vehicles based on big data analysis methods. Appl Energy 2017;207:354–62.
[20] Lee C, Wu C. Collecting and mining big data for electric vehicle systems using battery modeling data. International conference on information technology – new generations. 2015. p. 626–31.
[21] Haq IN, Saputra RH, Edison F, Kurniadi D, Leksono E, Yuliarto B. State of charge (SoC) estimation of LiFePO4 battery module using support vector regression. Electric vehicular technology and industrial, mechanical, electrical and chemical engineering. 2016. p. 16–21.
[22] Li A, Hu X, Li T, Zhang H. Research on the prediction method of power battery SOC based on deep learning. IEEE third international conference on data science in cyberspace. 2018. p. 673–9.
[23] Mao M, Wang Y, Yue Y, Chang L. Multi-time scale forecast for schedulable capacity of EVs based on big data and machine learning. Energy conversion congress and exposition (ECCE), 2017 IEEE. IEEE; 2017. p. 1425–31.
[24] Wang J, Gu Q, Wu J, Liu G, Xiong Z. Traffic speed prediction and congestion source exploration: a deep learning method. IEEE international conference on data mining. 2017. p. 499–508.
[25] Wu J, He H, Peng J, Li Y, Li Z. Continuous reinforcement learning of energy management with deep Q network for a power split hybrid electric bus. Appl Energy 2018;222:799–811.
[26] Kisi O, Parmar KS. Application of least square support vector machine and multivariate adaptive regression spline models in long term prediction of river water pollution. J Hydrol 2016;534:104–12.
[27] Jain R, Smith K, Culligan P, Taylor J. Forecasting energy consumption of multi-family residential buildings using support vector regression: investigating the impact of temporal and spatial monitoring granularity on performance accuracy. Appl Energy 2014;123:168–78.
[28] Jiménez-Valverde A, Lobo J. Distributions. The ghost of unbalanced species distribution data in geographical model predictions. Diversity Distribut 2010;12:521–34.
[29] Han H, Wang W, Mao B. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. Lecture notes in computer science. 2005. p. 878–87.
[30] Belov DI, Armstrong RD. Distributions of the Kullback-Leibler divergence with applications. Brit J Math Statist Psychol 2011;64:291–309.
[31] Zhao Z, Qiang F, Miao LI, Lei S, Wang H. Blind estimation of Pseudo-random noise codes in NPLSC-DSSS signals based on goodness of fit test. J Electron Inform

Technol 2017;39:749–53.
[32] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 2010;11:3371–408.
[33] Wang H, Shi X, Yeung DY. Relational stacked denoising autoencoder for tag recommendation. Twenty-ninth AAAI conference on artificial intelligence. 2015. p. 3052–8.
[34] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 2010;11:3371–408.
[35] Sun W, Shao S, Zhao R, Yan R, Zhang X, Chen X. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. Measurement 2016;89:171–8.
[36] Huang G, Zhu Q, Siew C. Extreme learning machine: a new learning scheme of feedforward neural networks. IEEE international joint conference on neural networks, 2004 proceedings. 2005. p. 985–90.
[37] Huang G, Zhou H, Ding X, Zhang R. Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B Cybern 2012;42:513–29.