

Article

Lithium-Ion Battery SOH Estimation Based on XGBoost Algorithm with Accuracy Correction

Shuxiang Song, Chen Fei and Haiying Xia *

Department of Electronic Engineering, Guangxi Normal University, Guilin 541004, China;
 songshuxiang@mailbox.gxnu.edu.cn (S.S.); feichen.china@gmail.com (C.F.)

* Correspondence: xhy22@mailbox.gxnu.edu.cn

Received: 5 January 2020; Accepted: 10 February 2020; Published: 13 February 2020



Abstract: SOH (state of health) estimation is important for battery management. Since the electrochemical reaction inside LIBS (lithium-ion battery system) is extremely complex and the external working environment is uncertain, it is difficult to achieve accurate determination of SOH. To improve the accuracy of SOH estimation, we propose a SOH estimation method for lithium-ion battery based on XGBoost algorithm with accuracy correction. We extract several features, including average voltage, voltage difference, current difference, and temperature difference, to describe the aging process of batteries. Due to the higher prediction accuracy and generalization ability of ensemble learning algorithm, the XGBoost model is established to estimate the SOH of lithium-ion battery. Then, the estimation values are corrected by Markov chain. Compared with the methods by XGBoost, random forest, k-nearest neighbor algorithm (KNN), SVM, linear regression, our proposed method shows an accuracy improvement by 10% to 20%. Additionally, the errors of our method are also superior to the others in terms of the average absolute error, root mean square error, and root mean square error.

Keywords: lithium-ion battery; SOH estimation; XGBoost; accuracy correction

1. Introduction

Lithium-ion batteries are widely used in electric vehicles because of their high energy density and low self-discharge rate. Not only that, lithium-ion batteries are also widely used in high-tech products such as mobile phones and various portable information processing terminals, however, the service life of lithium-ion battery has limited its further promotion and development of electric vehicles. As the remaining life decreases, the overall performance of the battery decreases, which is characterized by capacity attenuation and internal resistance increase [1]. Therefore, accurate SOH estimation is of great importance [2–4].

SOH indicates the capacity of battery to store electric energy, and the percentage is generally used to indicate the health status of the battery. Here, SOH is defined as [5]:

$$\text{SOH} = \frac{\text{Capacity}_{\text{aged}}}{\text{Capacity}_{\text{rated}}} \times 100\%, \quad (1)$$

where $\text{Capacity}_{\text{aged}}$ represents the current capacity of the battery, and $\text{Capacity}_{\text{rated}}$ represents the rated capacity of the battery.

At present, many SOH estimation methods have been presented in the literature. These methods can be divided into three categories: Direct measurement methods, physical failure methods, and data driven methods. Direct measurement methods include coulomb counting based methods [6] and internal resistance-based methods [7]. This kind of method calculates the state of charge (SOC) of the

lithium-ion battery by integrating the currents of the charging and discharging processes, and then further estimates the SOH by the coulomb counting. The electric resistance method is a method for measuring the internal resistance of a battery by using the mixed pulse power characterization, and estimating the SOH of the lithium-ion battery through the internal resistance of the battery. These methods are easy to implement in BMS and are simple to calculate, but the estimation accuracy is susceptible to measurement noise, and has a high dependence on measurement techniques, so it is not suitable for SOH.

The physical failure method needs to consider both the external factors, such as temperature, voltage, current, and state of charge, and the internal variables, for example, electrolyte concentration. It estimates the SOH by summarizing the electrochemical aging behavior of the battery [8]. Sánchez et al. proposed a methodology for designing semi-physical fuzzy models based on the fuzzy logic theory [9]. Hoenig et al. proposed another physical failure model based on battery impedance and open circuit voltage (OCV) to establish a multivariate linear model [10]. Since the physical failure method has a huge computational burden, it is difficult to apply in real-time SOH estimation.

The data-driven approach achieves SOH estimation by utilizing the test data of battery performance (current, voltage, temperature, impedance, etc.) without considering the electrochemical reaction and aging mechanisms of the battery. Due to its superior nonlinear mapping capability, it is widely used in SOH estimation of lithium-ion batteries. The data-driven method mainly includes Gaussian process regression (GPR), genetic algorithm, support vector machine (SVM), and autoregressive integral moving average (ARIMA). The GPR model makes use of battery health characteristics to predict the SOH of lithium-ion batteries [11]. Since the genetic algorithm depends partly on the initial population selection, it is difficult to obtain accurate SOH estimation results [12]. The support vector machine (SVM) has good nonlinear regression performance and is more capable of solving nonlinear problems [13–15]. The voltage and temperature during the discharge process are used as the inputs of the SVM model to describe the discharge process, but the use of extensive training sets make it difficult to solve the quadratic programming (QP) problem [16]. Autoregressive integral moving average (ARIMA) can also be used to estimate SOH, but can only be predicted under specific charge and discharge conditions [17]. XGBoost attracts wide attention because of its excellent learning effect and high learning efficiency. It is actually an extensible tree-enhanced machine learning system proposed by Chen et al. [18].

Aiming at the limitations above, we propose a method of SOH estimation for lithium-ion batteries using XGBoost algorithm with accuracy correction. The XGBoost model is established to estimate the SOH of lithium-ion battery, which can effectively avoid the over-fitting problem. Then, the estimation accuracy is corrected by Markov chain to further improve the estimation accuracy. The main contributions of this paper include: (1) The XGBoost model is established to obtain SOH estimation results with robustness; (2) the Markov Chain is utilized to correct the estimation results obtained by the XGBoost model to further improve the estimation accuracy; (3) we conduct extensive experiments to demonstrate the improvement of the estimation accuracy.

The following organization of this paper is described as follows. In Section 2, we detail the construction of the proposed model of SOH estimation using XGBoost algorithm with accuracy correction. The comparison of our proposed method with five other methods in estimation accuracy is shown in Section 3 and the conclusions are drawn in Section 4.

2. The Proposed Method

As shown in Figure 1, the proposed method, named as MC-XGBoost, can be divided into three parts: Feature selection, XGBoost estimation, Markov correction. First, we use voltage difference, temperature difference, and average voltage as the XGBoost model inputs to describe the discharge process of lithium-ion battery. Then, the XGBoost estimation model is established to predict the SOH of the battery. Finally, the Markov chain is used to correct the estimation results to further improve the accuracy of SOH estimation.

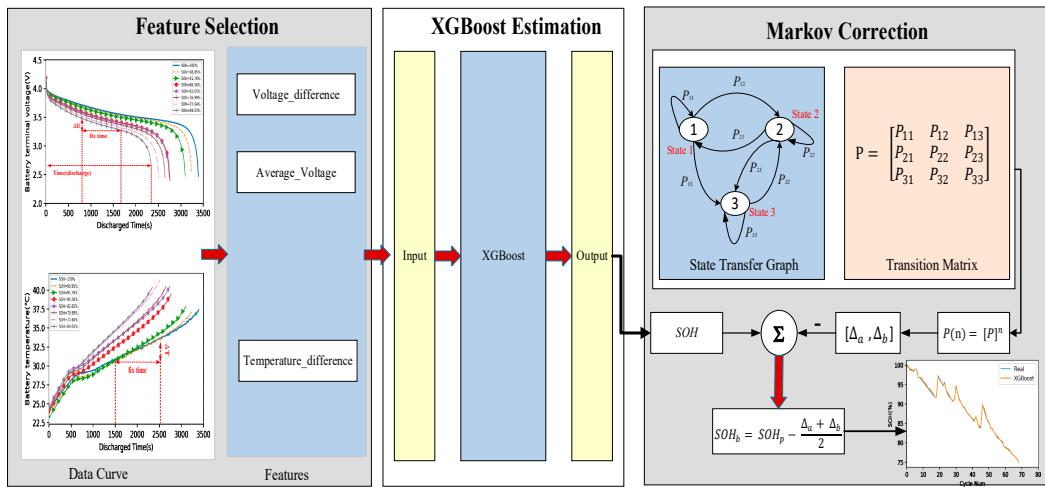


Figure 1. The flowchart of the MC-XGBoost method.

2.1. Feature Selection

Feature selection is of great importance in SOH estimation of lithium-ion batteries, Figure 2 shows the change curve of characteristics with time under different health conditions.

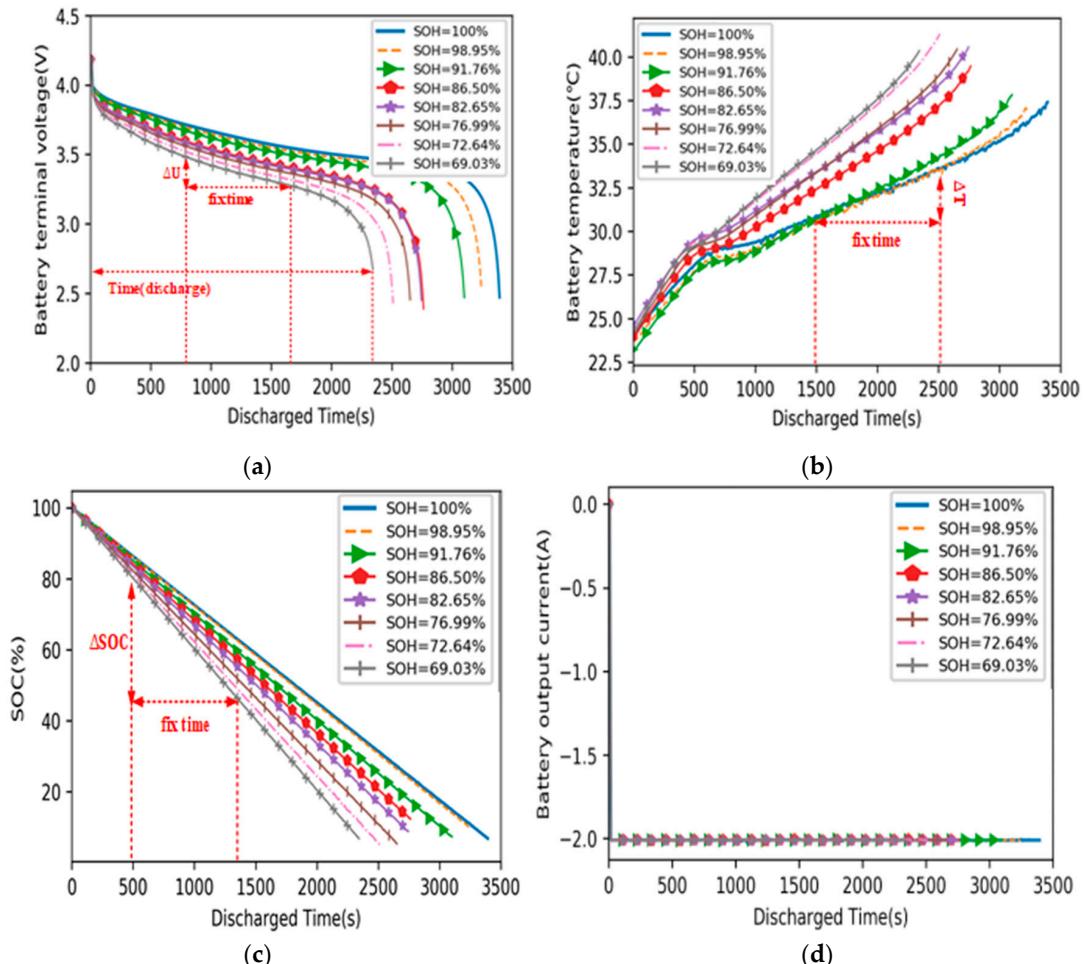


Figure 2. The characteristic curves under different SOH levels. (a) Voltage variation curve; (b) temperature change curve; (c) SOC change curve; (d) current change curve.

The average voltage (U_{ave}) is defined as follows:

$$U_{ave} = \frac{U_0 - U_e}{Time}, \quad (2)$$

Among them, U_0 represents the starting voltage of the battery cycle once, U_e represents the ending voltage of the battery cycle once, and $Time$ represents the time of the battery cycle once.

Figure 2a shows the voltage change curve of lithium-ion battery under different health conditions. It can be seen from the figure that under different SOH levels, the average voltage (U_{ave}) of lithium-ion batteries is not equal. This means that the health status of lithium-ion battery affects the average voltage (U_{ave}), that is, the average voltage (U_{ave}) can reflect the health status of the battery to a certain extent. Therefore, the average voltage (U_{ave}) in the discharge process is used as the aging characteristics of the battery.

The voltage difference (ΔU) represents the arbitrary voltage change in a short fixed time interval. It also can be seen from Figure 2a that the voltage difference (ΔU) in a fixed time increases with the decrease of SOH. Since the internal resistance of the battery increases with the aging of the battery. Therefore, the voltage difference (ΔU) can also be used as the aging characteristics of batteries.

The temperature difference (ΔT) represents the arbitrary temperature change in a short fixed time interval. Figure 2b shows the temperature change curve of lithium-ion battery under different health conditions. It can be seen from the figure that the temperature increment in a fixed time will increase with the decrease of SOH. Since with the aging of the battery, the internal resistance of the battery increases continuously, and the heat generated in the chemical reaction process increases. Therefore, the temperature difference (ΔT) can also be used as the aging characteristics of batteries.

It can be seen from Figure 2c that in a fixed period of time, the SOC goes down as the SOH keeps decreasing. Since the SOC drops, it will inevitably affect the discharge capacity of the battery, that is to say, SOC can reflect the health state of the battery to a certain extent. However, SOC is not generally used as a feature to predict SOH, it is often corrected according to the SOH values.

Figure 2d shows the current curve of lithium-ion battery under different health conditions. It can be seen from the figure that under different SOH levels, the current basically does not change. Therefore, the current difference can also be used as the aging characteristics of the battery.

Therefore, the voltage difference, temperature difference, and average voltage for the same time interval are used as the aging characteristics of batteries.

2.2. The XGBoost Model of SOH Estimation

XGBoost adds split features to generate a tree. It is to learn a new function $f(x)$ to fit the residual of the last prediction. When the k -th tree is trained, it is necessary to predict the score of a sample. According to the characteristics of the sample, it will fall on the corresponding leaf nodes in each tree. Each leaf node corresponds to a score. Finally, the corresponding score of each tree is added up as the predicted value of the sample, as shown in Figure 3.

XGBoost modeling: Give a generalized definition of objective function, find a suitable regression tree in each iteration to fit the residual of the last prediction, so that the objective function is further minimized, and the predicted value is closer and closer to the real value.

For example, data $([\Delta U_1, \Delta T_1, U_{ave1}], SOH_1)$, $([\Delta U_2, \Delta T_2, U_{ave2}], SOH_2)$, ..., $([\Delta U_i, \Delta T_i, U_{avei}], SOH_i)$, $i = 1, 2, \dots, n$. Among them, ΔU_i , ΔT_i , U_{avei} and SOH_i , respectively represent the voltage difference, temperature difference, average voltage, and health status corresponding to group I data.

We define the tree $f_t(x_i)$ as follows:

$$f_t(x_i) = \omega_{q(x)}, q : \mathbb{R} \rightarrow T, \omega \in \mathbb{R}^T, \quad (3)$$

where q represents the structure of each tree, it can map each sample to the corresponding leaf nodes, and T is the number of leaf nodes in the tree. Each f_t corresponds to an independent tree structure q and leaf weight w .

In order to standardize the improved objective function, we define tree complexity $\Omega(f_t)$:

$$\Omega(f_t) = \gamma T + 1/2 \lambda \sum_{j=1}^T w_j^2, \quad (4)$$

where w_j is the weight of the j th leaf node.

The objective function is defined as:

$$obj^{(t)} = \sum_{i=1}^n l(SOH_i, \hat{SOH}_i) + \sum_{k=1}^K \Omega(f_k) = \sum_{i=1}^n l(SOH_i, \hat{SOH}_i^{(t-1)} + f_t(\Delta U_i, \Delta T_i, U_{avei})) + \Omega(f_t) + c, \quad (5)$$

Among them, $\sum_{i=1}^n l(SOH_i, \hat{SOH}_i)$ is the residual of the predicted value and the actual value of the model. $f_t(\Delta U_i, \Delta T_i, U_{avei})$ represents the model to be added to the decision of the t -round. $\sum_{k=1}^K \Omega(f_k)$ is a penalty term for the complexity of each model. f_k is the model function of the tree and c is a constant.

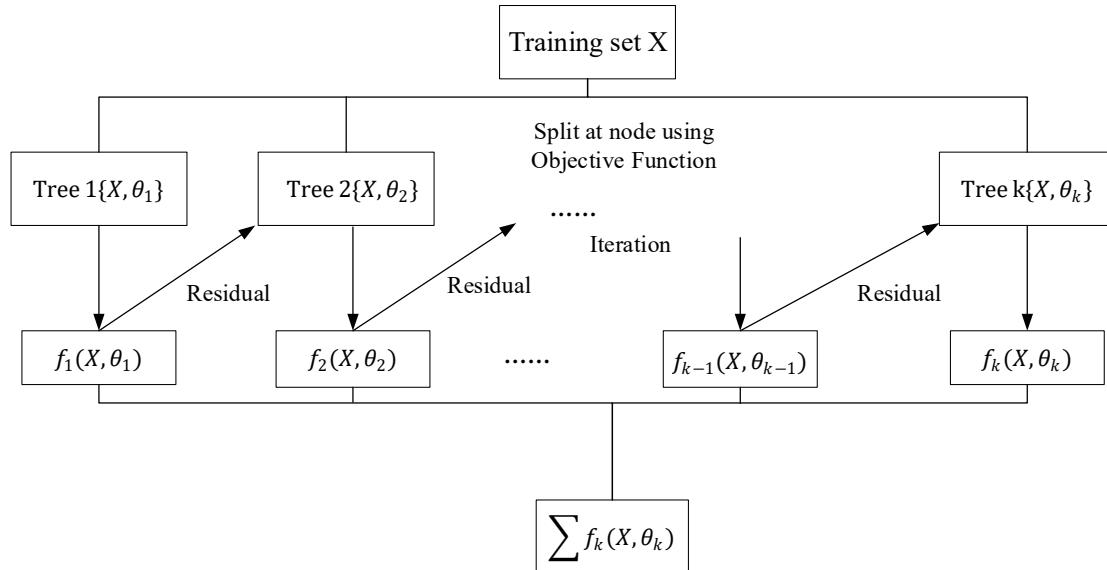


Figure 3. The flow chart of extreme gradient boosting.

The second-order Taylor's formula is expanded and the objective function is as follows:

$$obj^{(t)} \approx \sum_{i=1}^n [l(SOH_i, \hat{SOH}_i^{(t-1)}) + g_i f_t(\Delta U_i, \Delta T_i, U_{avei}) + 1/2 h_i f_t^2(\Delta U_i, \Delta T_i, U_{avei})] + \Omega(f_t) + c, \quad (6)$$

among them, $g_i = \partial_{\hat{SOH}_i^{(t-1)}} l(SOH_i, \hat{SOH}_i^{(t-1)})$, $h_i = \partial_{\hat{SOH}_i^{(t-1)}}^2 l(SOH_i, \hat{SOH}_i^{(t-1)})$,

The new objective function can be defined as:

$$Obj^{(t)} = \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) w_j + 1/2 \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T = \sum_{j=1}^T [G_j w_j + 1/2 (H_j + \lambda) w_j^2] + \gamma T, \quad (7)$$

Define the candidate feature set of splitting nodes of each tree as I_j , $I_j = \{i | q(\Delta U_i, \Delta T_i, U_{avei}) = j\}$.

The optimal weight ω_j^* of leaf node j and the optimal objective function solution $obj^{(*)}$:

$$\omega_j^* = -G_i/(H_i + \lambda), \quad obj^{(*)} = -\frac{1}{2} \sum_{j=1}^T G_i^2/(H_i + \lambda) + \gamma T, \quad (8)$$

Among them, $G_i = \sum_{i \in I_j} g_i$, $H_i = \sum_{i \in I_j} h_i$.

$obj^{(*)}$ is a function of marking tree structure and measuring the quality of tree structure q . The smaller the value of $obj^{(*)}$, the better. In this way, the optimal segmentation point and the optimal decision tree structure can be obtained to minimize the objective function.

XGBoost is improved and optimized on the basis of GBDT [19]. Regular terms are added to the objective function of each iteration to control the complexity of the model and support column sampling, it can not only reduce over-fitting, but also reduce calculation. Compared with the iteration principle of GBDT heuristic, the optimization criterion of XGBoost is based on the minimization of the objective function, and uses second-order Taylor expansion to support custom cost function. XGBoost blocks and sorts each feature, which makes it possible to parallel computing when searching for the best segmentation point. At the same time, by setting a reasonable block size, it makes full use of the CPU cache to speed up reading, and the speed of data reading is faster. When the eigenvalues are missing, XGBoost can also automatically learn the direction of node splitting.

2.3. Markov Chain Correction

Markov chain, also known as discrete-time Markov chain, is a stochastic process in which state space transitions from one state to another. This process requires a “no memory” nature, that is, the probability distribution of the next state can only be determined by the current state, and is independent of the events before it in the time series. Assuming that the sequence state is $\dots X_{t-2}, X_{t-1}, X_t, X_{t+1} \dots$, then the conditional probability in state X_{t+1} depends only on state X_t , i.e.,

$$P(X_{t+1} | \dots X_{t-2}, X_{t-1}, X_t) = P(X_{t+1} | X_t), \quad (9)$$

(1) Partition interval

The difference between the real data and the predicted data is called the relative estimation. The relative estimation of the SOH of the lithium-ion battery is calculated, X represents the relative estimation, and the relative estimations are normalized:

$$\bar{x} = (x - x_{min}) / (x_{max} - x_{min}), \quad (10)$$

According to the golden section method, the normalized values of relative estimation are divided into three states: E_1 , E_2 , and E_3 , and the three states are inversely normalized to the relative estimation sequences by the above formula. The relative estimation can be divided into three intervals (S_1, S_2, S_3).

(2) The computation of state transition matrix

The transition probability P_{ij} is a one-step transition probability representing the transition probability of state i to state j , and the equation is expressed as follows:

$$P_{ij} = c_{ij} / C_i, \quad (11)$$

where c_{ij} represents the access frequency from state i to state j between two adjacent cycles, and C_i represents the number of states moving from state i between two adjacent cycles. The state transition matrix is:

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix}, \quad (12)$$

Therefore, the transition matrix of the n -th step is $P(n) = [P]^n$.

(3) Estimation correction

To further improve the estimation accuracy, Markov Chain is used to correct the estimation results obtained by the XGBoost model when the relative error between the predicted value and the true value is large. The correction is performed by the following equation.

$$SOH_b = SOH_p - (\Delta_a + \Delta_b)/2, \quad (13)$$

where Δ_a and Δ_b are the upper and lower limits of the estimation interval corresponding to their states of the relative estimation, SOH_b indicates the corrected value, and SOH_p indicates the value estimated by the XGBoost model.

2.4. The Implementation of the MC-XGBoost Method

As shown in Algorithm 1, the implementation of the MC-XGBoost method mainly has three steps: Data setting, XGBoost estimation process, and Markov correction process. The three steps are described in the following:

- Steps 1: Preprocess the data and set various parameters, such as eta, max_depth, min_child_weight, seed, colsample_bytree, gamma, etc.
- Steps 2: Set up training data and test data, use XGBoost algorithm to train the prediction model for SOH estimation. If it is not the optimal parameter, the error between the predicted value and the real value is large, it is necessary to reset the parameters, repeat the above operation, and adjust the parameters until the optimal parameters are obtained.
- Steps 3: Calculate the relative residuals by comparing the predicted values obtained by the XGBoost model with the real ones, partition the normalized relative estimations according to the golden section method, and compute the state transition matrix to determine the next state. Then, the corrected values are obtained by using Equation (13).

Algorithm 1. The implementation of the MC-XGBoost Algorithms.

-
- 1: Input: Dataset $X = [\Delta U_i, \Delta T_i, U_{avei}], i \in \{1, 2, \dots, n\}$, the loss function: $l(SOH_i, \hat{SOH}_i)$, the total number of sub-tree: T
 - 2: Output: Predicted health status of lithium-ion batteries: SOH_b ;
 - 3: Repeat
 - 4: Initialize the t -th tree $f_t(x_i)$
 - 5: Compute $g_i = \partial_{\hat{SOH}_i^{(t-1)}} l(SOH_i, \hat{SOH}_i^{(t-1)})$
 - 6: Compute $h_i = \partial_{\hat{SOH}_i^{(t-1)}}^2 l(SOH_i, \hat{SOH}_i^{(t-1)})$
 - 7: Use the statistics to greedily grow a new tree $f_t(x_i)$: $obj^{(*)} = -\frac{1}{2} \sum_{j=1}^T G_j^2 / (H_j + \lambda) + \gamma T$, among them,
 - 8: $G_i = \sum_{i \in I_j} g_i, H_i = \sum_{i \in I_j} h_i$
 - 9: Add the best tree $f_t(x_i)$ into the current model
 - 10: Until all T sub-trees are processed
 - 11: A strong regression tree based on all weak regression subtrees
 - 12: Compute the SOH_p based on the strong regression tree
 - 13: Repeat
 - 14: Normalize the relative estimations: $\bar{x} = \frac{x - x_{min}}{x_{max} - x_{min}}$; determine the status: E_1, E_2, E_3 ; calculate the transition matrix: P ; correct the prediction results by Markov Chain: $SOH_b = SOH_p - (\Delta_a + \Delta_b)/2$
 - 15: Output SOH_b based on Markov Chain
-

3. Experimental Results

The data used in our experiments comes from the NASA Ames Research Center [20]. In the process of stable discharge experiment, four lithium-ion batteries (5, 6, 7, and 18) were obtained by charging, discharging, and impedance, and two kinds of experiments were carried out. One group takes the discharge data of #5, #7, and #18 as the training set of the model, and the #6 discharge data is used to estimate the accuracy of the model. The other group uses the discharge data of #5, #6, and #18 as the training set of the model, and the discharge data of #7 is used for the precision estimation of the mode.

In order to verify the SOH estimation accuracy of lithium-ion batteries by the XGBoost method, we compare it with the Random Forest-based method, the Linear Regression-based method, the KNN-based method, and the SVM-based method on data set #6 and #7. Figure 4 shows the prediction results and prediction error of SOH on data set #6 and #7.

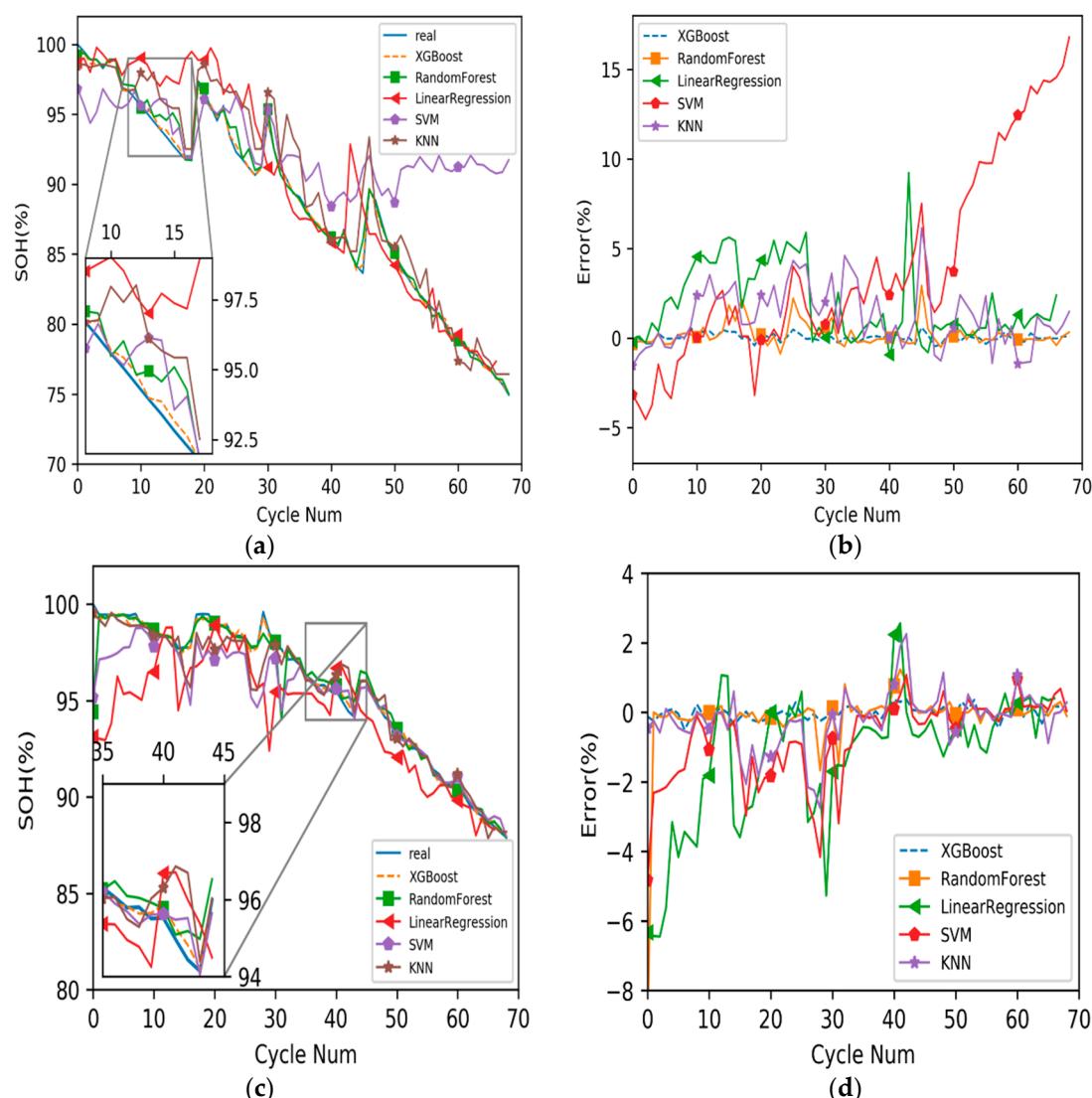


Figure 4. The predicted results and prediction errors of SOH estimation on data set #6 and #7. (a) The predicted results of SOH estimation on data set #6; (b) the prediction errors of SOH estimation on data set #6; (c) the predicted results of SOH estimation on data set #7; (d) the prediction errors of SOH estimation on data set #7.

The predicted results in Figure 4a show that on the #6 data set, the predicted results of linear regression, support vector machine, and KNN have obvious errors with the real values, while XGBoost and random forest have higher estimation accuracy in the latter estimation, but XGBoost is significantly closer to the real values than KNN in the earlier estimation. The prediction errors in Figure 4c shows that there are obvious errors between the predicted results of linear regression, support vector machine, random forest and KNN algorithm and the real values on the #7 data set, while the predicted results of XGBoost are close to the real values.

As shown in Figure 4b, the residual curves of XGBoost and random forest fluctuates up and down around 0 in the second half of the curve, but in the first half, the errors generated by the random forest fluctuates up and down far away from 0. Therefore, the estimation of random forest is not accurate in the early stage.

Table 1 shows that, the performances of the XGBoost method outperforms the other four methods on data set #6 and #7 in terms of mean absolute error (MAE), root mean square error (RMSE), and maximum error.

Table 1. The estimation results by the XGBoost method and the other four methods on data set #6 and #7.

	MAE	Battery #6 RMSE	Maximum Error	MAE	Battery #7 RMSE	Maximum Error
XGBoost	0.001615	0.002132	0.005972	0.001336	0.001712	0.004244
Random Forest (RF)	0.004379	0.006947	0.028183	0.002716	0.006108	0.059234
KNN	0.016193	0.020937	0.061488	0.005732	0.008387	0.027463
Linear Regression (LR)	0.020204	0.028393	0.092293	0.016584	0.025346	0.088665
SVM	0.045509	0.056715	0.118535	0.053396	0.060812	0.099835

It can be seen from Table 1 that the values of XGBoost in three technical indexes are lower than those of the other four algorithms on data set #6 or #7.

In conclusion, no matter on which data set #6 or #7 shown in Figure 5, the prediction results of XGBoost are almost close to the real values and have higher estimation accuracy.

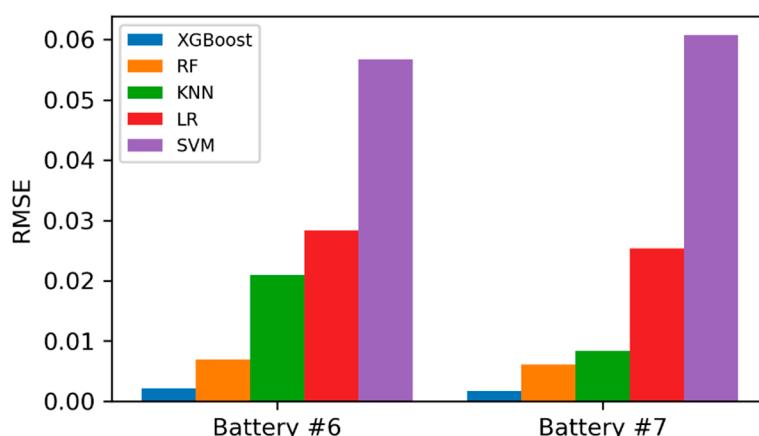


Figure 5. RMSE comparison by different methods.

The discharge process of lithium-ion battery is a time-varying process, and the change of its health state is a long-term process from the beginning of use to the end of life replacement. The state of health decreases with time, and the degree of health decreases. Markov chain has a significant prediction effect on the long-term system state, so we can correct the predicted errors by Markov chain to further improve the estimation accuracy by Markov chain.

It can be seen from Figure 6 that XGBoost and MC-XGBoost both have a good prediction effect in the later stage, and the predicted values are close to the real ones. However, in the early stage, no matter on which data set #6 or #7, the predicted values under MC-XGBoost are closer to the real values than XGBoost.

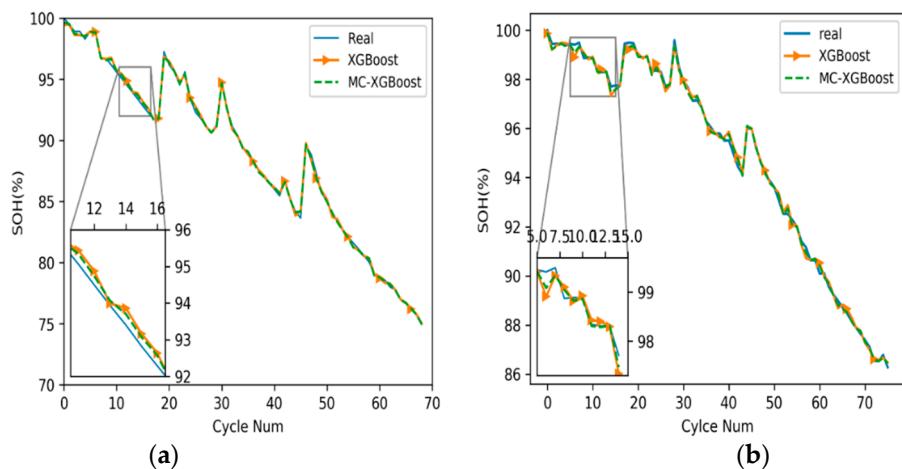


Figure 6. The comparisons of the prediction results by the MC-XGBoost and XGBoost methods. (a) The prediction results on data set #6; (b) the prediction results on data set #7.

The prediction errors generated by MC-XGBoost and XGBoost methods are shown in Figure 7. No matter on which data set #6 or #7, the errors generated by the MC-XGBoost method fluctuate up and down near 0, and the fluctuation range is relatively small. Although most of the errors generated by XGBoost fluctuate up and down near 0, the fluctuation range is relatively large.

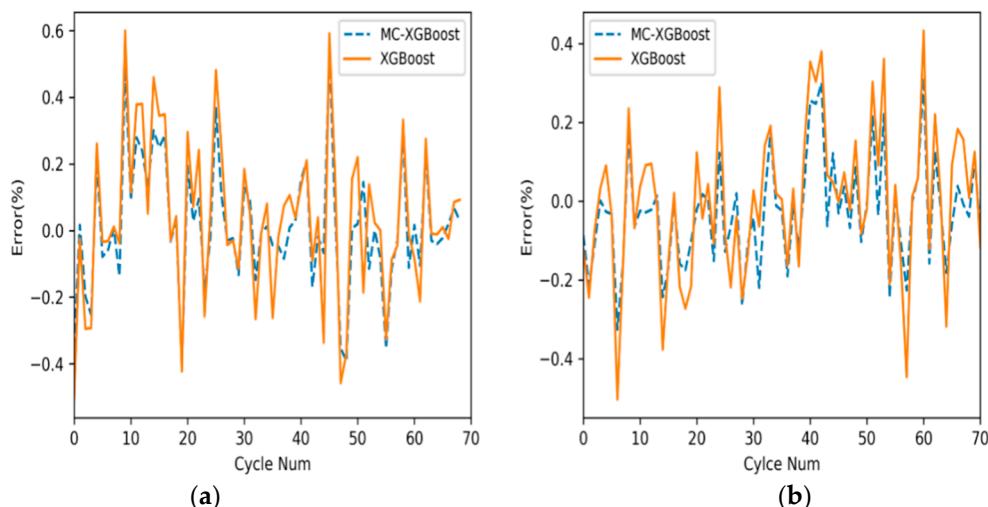


Figure 7. The comparisons of the prediction errors by the MC-XGBoost method and the XGBoost-based method. (a) The comparisons of the prediction errors on data set #6; (b) the comparisons of the prediction errors on data set #7.

The MAE of MC-XGBoost and XGBoost is 0.001364 and 0.001615, respectively on data set #6, and 0.001035 and 0.001336, respectively on data set #7. The RMAE of MC-XGBoost and XGBoost is 0.001878 and 0.002132, respectively on data set #6, and 0.001355 and 0.001712, respectively on data set #7. The maximum error of MC-XGBoost and XGBoost is 0.005355 and 0.005972, respectively on data set #6, and 0.003263 and 0.004244, respectively on data set #7, as shown in Figure 8.

Compared with the XGBoost method, the MC-XGBoost method reduces the maximum error from 0.005972 to 0.005355 on data set #6 and 0.004244 to 0.003262 on data set #7 as shown in Figure 8. Therefore, compared with the other five algorithms, MC-XGBoost has a small error fluctuation range and can guarantee the stability of SOH prediction.

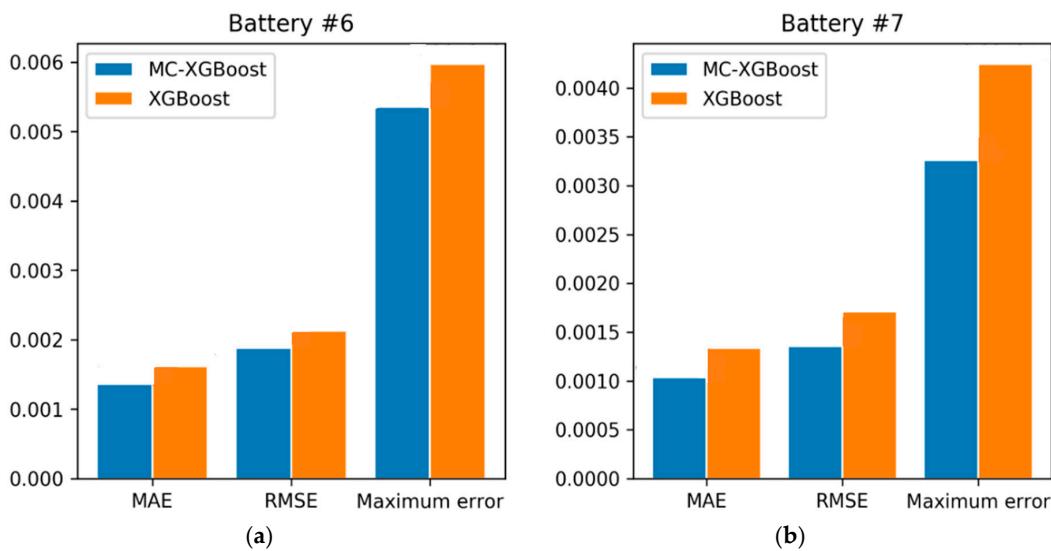


Figure 8. The comparisons of three technical indicators by the MC-XGBoost method and the XGBoost-based method. (a) The comparisons of three technical indicators on data set #6; (b) the comparisons of three technical indicators on data set #7.

The MC-XGBoost method decreases by 15.5%, 22.5% in MAE, and by 11.9%, 20.8% in RMSE compared with the XGBoost-based method. That is to say, the performance ranks of the MC-XGBoost method and the other five methods is the MC-XGBoost method, the MC-XGBoost method, the XGBoost-based method, the RF-based method, the KNN-based method, the LR-based method, and the SVM-based method. Therefore, the Markov correction is effective in improving the accuracy of the predicted values.

Figure 9 analyzes the contributions of the features to the SOH prediction. Among them, the average voltage has the greatest influence on the accuracy of the SOH prediction, followed by the voltage difference, and the temperature difference is the smallest. The feature importance ranking in SOH prediction is: Average voltage, voltage difference, temperature difference.

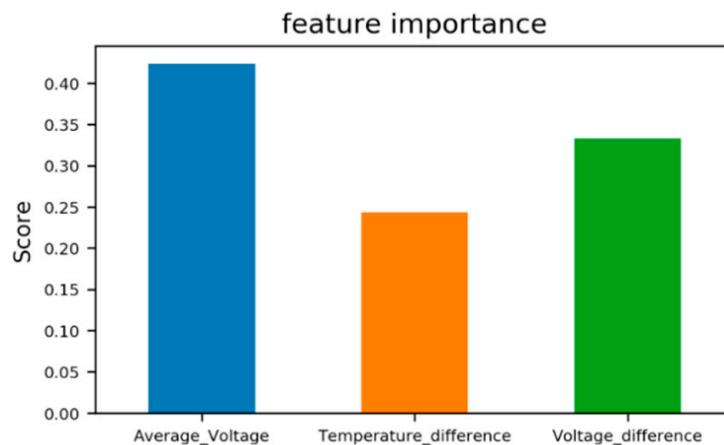


Figure 9. The importance of the battery features.

4. Conclusions

This paper presented a method for estimating the SOH using XGBoost algorithm with accuracy correction. Three battery features including the average voltage, voltage difference, and temperature difference are used to predict the SOH of lithium-ion batteries. SOH is first predicted by XGBoost algorithm, and then Markov chain is established to correct the predicted values. To verify the

effectiveness of XGBoost, we compare XGBoost with five regression algorithms. The XGBoost algorithm has a high prediction accuracy, and strong generalization ability. The comparison between Markov correction and XGBoost shows that the Markov correction can significantly improve the prediction accuracy of XGBoost by 10%–20%. In addition, experiments on #6 and #7 battery data sets verify the generality of this method. In all, the experimental results show that the SOH prediction accuracy of the MC-XGBoost method is superior to the other five methods in three technical indicators, and the proposed method has higher SOH estimation accuracy.

Author Contributions: Conceptualization, S.S., C.F., H.X.; data curation, S.S., C.F.; formal analysis, S.S., C.F., H.X.; methodology, S.S. and C.F.; software, C.F.; supervision, S.S.; validation, C.F., H.X.; writing—review and editing, S.S., C.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the science and technology major project of Guangxi, under grant number AA18118009.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Paul, N.; Wandt, J.; Seidlmaier, S.; Schebesta, S.; Mühlbauer, M.J.; Dolotko, O. Aging behavior of lithium iron phosphate based 18650-type cells studied by in situ neutron diffraction. *J. Power Sources* **2017**, *345*, 85–96. [[CrossRef](#)]
2. Weng, C.; Feng, X.; Sun, J.; Peng, H. State-of-health monitoring of lithium-ion battery modules and packs via incremental capacity peak tracking. *Appl. Energy* **2016**, *180*, 360–368. [[CrossRef](#)]
3. Weng, C.; Feng, X.; Sun, J.; Ouyang, M.; Peng, H. Battery SOH Management Research. In Proceedings of the 4th IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling (ECOSM 2015), Ohio State University, Columbus, OH, USA, 23–26 August 2015; Volume 48.
4. Moura, S.J.; Krstic, M.; Chaturvedi, N.A. Adaptive PDE observer for battery SOC/SOH estimation. In Proceedings of the ASME 5th Annual Dynamic Systems and Control Conference Joint with the JSME 11th Motion and Vibration Conference (DSCC2012-MOVIC2012), Fort Lauderdale, FL, USA, 2012; pp. 101–110.
5. Ramadass, P.; Haran, B.; White, R.; Popov, B.N. Mathematical modeling of the capacity fade of Li-ion cells. *J. Power Sources* **2003**, *123*, 230–240. [[CrossRef](#)]
6. Ng, K.S.; Moo, C.S.; Chen, Y.P.; Hsieh, Y.C. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Appl. Energy* **2009**, *86*, 1506–1511. [[CrossRef](#)]
7. Zhang, J.; Lee, J. A review on prognostics and health monitoring of Li-ion battery. *J. Power Sources* **2011**, *196*, 6007–6014. [[CrossRef](#)]
8. He, Z.; Gao, M.; Wang, C.; Wang, L.Y.; Liu, Y. Adaptive state of charge estimation for Li-ion batteries based on an unscented Kalman filter with an enhanced battery model. *Energies* **2013**, *6*, 4134–4151. [[CrossRef](#)]
9. Sánchez, L.; Couso, I.; González, M. A design methodology for semi-physical fuzzy models applied to the dynamic characterization of LiFePO₄ batteries. *Appl. Soft Comput.* **2014**, *14*, 269–288. [[CrossRef](#)]
10. Hoenig, S.; Singh, H.; Palanisamy, T.G.; Eagan, M. Method and Apparatus for Predicting the Available Energy of a Battery. U.S. Patent 6,618,681, 9 September 2003.
11. Liu, D.; Pang, J.; Zhou, J.; Peng, Y.; Pecht, M. Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression. *Microelectron. Reliab.* **2013**, *53*, 832–839. [[CrossRef](#)]
12. Chen, Z.; Mi, C.C.; Fu, Y.; Xu, J.; Gong, X. Online battery state of health estimation based on Genetic Algorithm for electric and hybrid vehicle applications. *J. Power Sources* **2013**, *240*, 184–192. [[CrossRef](#)]
13. Widodo, A.; Shim, M.-C.; Caesarendra, W.; Yang, B.-S. Intelligent prognostics for battery health monitoring based on sample entropy. *Expert Syst. Appl.* **2011**, *38*, 11763–11769. [[CrossRef](#)]
14. Patil, M.; Tagade, P.; Hariharan, K.S.; Kolake, S.M.; Song, T.; Yeo, T.; Doo, S.-G. A novel multistage Support Vector Machine based approach for Li ion battery remaining useful life estimation. *Appl. Energy* **2015**, *159*, 285–297. [[CrossRef](#)]
15. Nuhic, A.; Terzimehic, T.; Soczka-Guth, T.; Buchholz, M.; Dietmayer, K. Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. *J. Power Sources* **2013**, *239*, 680–688. [[CrossRef](#)]

16. Klass, V.; Behm, M.; Lindbergh, G. A support vector machine-based state-of-health estimation method for lithium-ion batteries under electric vehicle operation. *J. Power Sources* **2014**, *270*, 262–272. [[CrossRef](#)]
17. Zhou, Y.; Huang, M. Lithium-ion batteries remaining useful life prediction based on a mixture of empirical mode decomposition and ARIMA model. *Microelectron. Reliab.* **2016**, *65*, 265–273. [[CrossRef](#)]
18. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), San Francisco, CA, USA, 13–17 August 2016.
19. Yang, S.; Wu, J.; Du, Y.; He, Y.; Chen, X. Ensemble learning for short-term traffic prediction based on gradient boosting machine. *J. Sens.* **2017**, *2017*, 1–15. [[CrossRef](#)]
20. Saha, B.; Goebel, K. *NASA Ames Prognostics Data Repository: Battery Data Set [DS/OL]*; NASA Ames Research Center: Santa Clara County, CA, USA, 2007. Available online: <http://ti.arc.nasa.gov/project/pRognostic-data-repository> (accessed on 10 September 2017).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).