State of Health Estimation and Remaining Useful Life Prediction of The Lithium Battery for New Energy Vehicles with Long Short-Term Memory Neural Network

2020 5th International Conference on Universal Village (UV) | 978-1-7281-9523-0/20/$31.00 ©2020 IEEE | DOI: 10.1109/UV50937.2020.9515119

David Chang*
Machine Learning Algorithm Department
Shanghai CloudReady Technology Co., Ltd
Shanghai, China
david.chang@dtarray.com

Weixia Liu*
Information Technology & Data Technology Department
Beijing Electric Vehicle Automobile Co., Ltd
Beijing, China
liuweixia@bjev.com.cn

Xun Tian
Information Technology & Data Technology Department
Beijing Electric Vehicle Automobile Co., Ltd
Beijing, China
tianxun@bjev.com.cn

Jiayong Xiao
Information Technology & Data Technology Department
Beijing Electric Vehicle Automobile Co., Ltd
Beijing, China
xiaojiayong@bjev.com.cn

Yuan Li
Machine Learning Algorithm Department
Shanghai CloudReady Technology Co.Ltd
Shanghai, China
yuan.li@dtarray.com

Chenxi Liu
Machine Learning Algorithm Department
Shanghai CloudReady Technology Co.Ltd
Shanghai, China
xichen.liu@dtarray.com

Xiaonan Li**
Machine Learning Algorithm Department
Shanghai CloudReady Technology Co.Ltd
Shanghai, China
xiaonan.li@dtarray.com

***Abstract*—This paper introduces a model-based method to estimate the real-time State of Health (SoH) of the lithium battery of NEV (New Energy Vehicle) with machine learning algorithms upon the traditional ampere-hour integral method. The traditional methods for estimating the SoH (State of Health) of the lithium battery are ampere-hour integral, IC-curve, Big data, and Kalman filtering, but the problem of those methods is that it can only estimate the SoH in the past based on the historical battery data rather than the current SoH or the future life cycle. By combining machine learning algorithms and the ampere-hour method, we develop a way to estimate the real-time SoH, enabling the car manufacturer to understand better the current state of the lithium battery of NEV. Upon that, we also develop an algorithm to predict the future decay curve of SoH by using a deep neural network, the long short-term memory network, making the life cycle of the lithium battery more predictable. Our method hits 0.009 absolute mean error of real-time SoH prediction and 0.021 for future decay curve prediction from the real NEVs test by performing on the dataset based on actual real-time monitoring data provided by one OEM (Original Equipment Manufacturer).**

Keywords—the lithium battery for NEV, long short-term memory networks, real time SoH estimation, SoH future decay curve prediction

I. INTRODUCTION

With the promotion of electric vehicles and the application of IoV (Internet of Vehicle) technology in China, more and more electric vehicles have entered the Chinese consumer market. A quantity of real-time driving data has been collected according to the Chinese national standard [1]. Those collected data make it possible to do further research on new energy vehicles, and the power battery analysis is one of the most critical subjects in the area. The power battery is the power source of NEVs, and the capacity of the battery continues to decay as the number of charging and discharging and the mileage increase. This reaction is a typical dynamic nonlinear electrochemical system, and the internal parameters are difficult to measure during the online application. There are still considerable challenges in degraded state identification and estimation. Therefore, developing an algorithm capable of dynamic, self-learning, and optimization by using the collected data is urgently needed to accurately estimate the power battery's current state.

This paper aims to analyze the state of power battery by the evaluation of State of Health (SoH), which refers to the ratio between the actual value and the nominal value of specific directly measurable or indirectly calculated performance parameters after the battery is used for some time under certain conditions, used to judge the battery State of health. This article uses the percentage of the battery's charge or discharge capacity to the battery's nominal capacity to represent the battery's SoH, generally expressed as a percentage. For a new battery, the SoH value is generally higher than 100%. As the battery gets used, it ages, and the SoH will gradually decrease. When the battery capacity, which is quantified as SoH, is less than 80%, the battery should be replaced according to the IEEE Standard 1188-1996 [2].

* David Chang and Weixia Liu make equal contribution.
** Xiaonan Li is the corresponding author.

1) IC curve of open circuit voltage: It uses the Lorentz function to fit the IC peak to obtain the characteristic parameters of the IC peak. The equation is shown in (1), where A_i is the area under the i-th IC peak, W_i is the half-width of the i-th IC peak, V_{0i} is the symmetric center of the i-th IC peak, and V is the external voltage of the power battery. We can simplify it by converting it to an integral form, and we can get the battery charging model shown in (2), where C is a constant.

$$y = \sum_{i=1}^{n} \frac{2Ai}{\pi} \frac{W_i}{4(V-V_{0}i)^2+W_i^2} \quad (1)$$

$$Q = \sum_{i=1}^{n} arctan \frac{2(V-V_{0}i)}{W_i} + C \quad (2)$$

2) Ampere Hour integral method: It calculates the capacity charged in each charge through the SoC (State of Charge) change in each charging segment, and then the SoC change is used to standardize the inrush capacity to obtain the capacity corresponding to every one percent of SoH, multiplied by 100 to get the current time calculated total available capacity.

3) Kalman Filtering method [3]: A Kalman filter has the function of estimating the state of a system from measurements containing defects. It needs the previous sample's system variables, and the current sample's forcing terms and observations.

4) Big Data method [4]: It uses battery historical data and measured data, relying on the massive distributed storage and parallel computing capabilities provided by the big data platform, extracts relevant features to realize SoH online estimation and corrects through the differences between the monomers, and compares multi-dimensional battery SoH attenuation.

The disadvantage of the above methods is that we can only calculate the historical SoH data but cannot know the current state. So we implement a method of employing machine learning knowledge to predict the battery's SoH. With the long-term data collection, we can mine the hidden battery health status information and its evolution rules using the battery rating information and status monitoring data to achieve battery SoH prediction. Traditionally, the SoH is roughly estimated by the ampere-hour integral method; however, this method can only estimate the SoH in the past through historical data. As an improvement and extension of the ampere-hour integral method, we built machine learning models to enable real-time SoH prediction. Based on that, we also developed a future trend prediction model using Long Short-Term Memory networks.

II. DATA

A. Data Source

The dataset is based on actual RTM data provided by one OEM. The data are generated by two types of vehicles, and each has 30 cars, thus 60 cars in total. The data collection starts from the middle of 2018 and ends at the beginning of 2020; the time span is about 500 days. The data is recorded every 10 seconds. So approximately there are over 7,000,000 samples for one car, and thus over 420,000,000 samples in total.

B. Data Preparation

In this step, the data of the battery come from the monitoring data of the electric vehicle. The monitoring data are collected every ten seconds and are generated in different vehicle states. For example, they might come from during driving or charging. The battery monitoring data include batteries' data and the vehicle status data related to the battery during its regular use; thus, there are more than 200 data variables.

The battery usage data are all streaming data based on time series, including current, voltage, temperature, remaining power (SoC), etc. The relevant data content is shown in the Table I.

TABLE I. THE DATA INFORMATION AND DESCRIPTION

Data Type	Description
VIN	Represents the unique identification code of the vehicle
Collection Time	Time stamp, normal collection frequency is 10 seconds
Vehicle mode	The working mode of the vehicle including driving, charging, etc.
SoC	The state of charge of the battery, the current remaining capacity of the battery
Power battery charge/discharge current	The size of the output or input current of the battery during operation
Maximum voltage of single cell	The highest voltage of all batteries
Minimum voltage of single cell	The lowest voltage of all batteries
Cell Voltage of #'s battery	Voltage values of 1~n collected battery, normally n=100
Maximum temperature of single cell	Maximum temperature of all temperature detection points
Minimum temperature of single cell	Minimum temperature of all temperature detection points
Temperature of #'s temperature monitoring point	Temperature of 1~m temperature detection points, normally m=20
Order of Charging	1,2,3,4 …
Starting time of charing	The time when charging starts
Ending time of charing	The time when charging ends
SoC when charing starts	The State of Capacity when charging starts[a]
SoC when charing ends	The State of Capacity when charging starts[a]
Maximum Temperature During Charging	The maximum temperature during charging[b]
Minimum Temperature During Charging	The minimum temperature during charging[b]
Average Temperature During Charging	The average temperature during charging[b]
Maximum Temperature After Charging	The maximum temperature after charging[b]
Minimum Temperature After Charging	The minimum temperature after charging[b]
Average Temperature During Charging	The average temperature after charging[b]
Capacity During Charging	The Current integral during charging[c]
Total Charging Time	The total charging time of until the current charging

Total Driving Time	The total driving time of until the current charging
Total Mileage	The total mileage time of until the current charging

[a.] First check if this SoC data is compliant.

[b.] Normally there are 16-20 temperature stamps, first check if the temperature is normal and then get the maximum/minimum/average values upon them.

[c.] The Δt (the interval of the current integral during charging) is not always10 seconds.

C. Data Processing

Ensuring the high-quality of data is conducive to improving the accuracy of the results, so in this process, we need to sort out the collected data. The first step of data sorting is to clean the data, and thus formulate corresponding cleaning rules to convert the low-quality data into data that meet the data quality requirements. The clean-up rules include:

1) Missing Value Assignment: During the transmission of the battery data, it is common to drop the packet and cause the variable to be missing. The average or intermediate value of the variable or adjacent interpolation is often used to assign the empty variable.

2) Removal of wrong values: By setting a reasonable value range of each variable of the data related to the use of electric vehicle batteries, that is, the threshold, we check whether the data meet the requirements, and we delete or correct the data beyond the normal range.

3) Cross-check: By setting the mutual constraints and dependencies of data related to the use of electric vehicle batteries, logically unreasonable or conflicting data will be deleted or corrected.

We convert all numeric data types to "np. float32" and convert all category data types to "np.int8" type. After cleaning the data, the data is constructed based on the time unit. The time unit may be based on milliseconds, seconds, minutes, etc. Nevertheless, the time unit may not be consistent with the collected frequency.

After that, data constructed based on the time unit need to be evaluated and corrected. The evaluation includes filtering out erroneous data. For example, including, but not limited to, missing values, outliers, period errors, and calculation specification errors. After the evaluation, the erroneous data are basically corrected by using several ways of remedy. For example, for missing values, set the null value to supplement the missing data 0; For outliers, set the negative value to 0 to avoid errors during training; For the value with the wrong time period, we should locate the period properly, and the data from that period should be adjusted and re-run; for the value with the wrong calculation specification, the measurement should be adjusted, and the data should be re-run as well.

D. Feature Processing

The data need to be processed and calculated in subsequent processing steps. To facilitate calculation and identification of the characteristics of the data, the first necessary step is to characterize the sorted data to reveal various features so that the calculation and identification can be more convenient.

In this step, the summary and extraction of data include rolling aggregation. The rolling aggregation refers to setting a time window and calculating the aggregation value using a predetermined variable within the time window. The aggregation value may be a sum of data, an average value, or a standard deviation. As shown in Fig. 1, the t1 node, which refers to the time window, is three nodes, and its rolling aggregation is to calculate the sum, mean, or standard deviation of these three nodes in the time window.

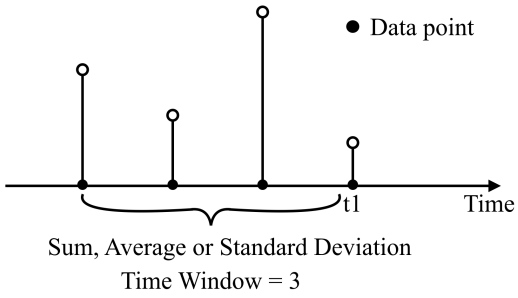

Fig. 1. The processing method of the time series data

More variable data are needed to provide better and even additional learning and prediction capabilities for the learning algorithm, which is summarized and extracted from the battery data based on the time series so that the initial characteristic variables expand. For example, when there are 126 feature variables, the extended data are mainly of two types: the first is to increase the initial 126 feature variables according to the rolling aggregation mean, and thus increase 126-2=124 features; The second category is to increase the initial 126 feature variables according to the standard deviation of rolling aggregation and thus increase 124 as well. As a result, the number of the final variable obtained is 126+124+124=374. In this way, more variable data can be provided, which is beneficial to provide better prediction ability for the learning algorithm.

III. Model

A. Overall Perspective

There are two steps to construct the prediction model:

1) Current real-time battery health prediction model: The model will input the current set of characteristic values in real-time and return the current battery health predicted by the model.

2) Future battery health state prediction model: This model aims to predict the decay curve for a vehicle's SoH in the next three months by analyzing the previous month's SoH data.

The details of the two models will be introduced in the following paragraph.

B. Establishment of Current Real-Time Battery Health Model

Seven models are used to predict the current real-time SoH. Two of them are linear regression model with L1, L2 regularization, the Ridge and Lasso Regression, the other three are Gradient Boosting Decision Tree based models, XGBoost, LightGBM and CatBoost, and the last one is a deep learning model, the deep neural network.

Since it is a regression problem, we select Mean Absolute Error (MAE) to measure the performance of each model.

1) Feature Engineering

To choose the most relevant features to feed in the model, we start with features that have apparent positive and negative correlations with the corresponding SoH, which is calculated by the ampere-hour integral method. The features are in Table II.

TABLE II. THE SELECTED FEATURES

Feature Name	Explanation	Calculation Logic
mile	Accumulated mileage	Modeling cumulative variable characteristics
day	Cumulative days	Modeling cumulative variable characteristics
total_vol	Total voltage	Extraction capacity needs
total_cur	Total current	For Ampere-hour integral
soc	battery capacity	For Ampere-hour integral
cap_28	SoC from 20-80 charge time capacity	For Ampere-hour integral
cap_46	SoC from 40-60 charge time capacity	For Ampere-hour integral
total_ch-arge	Total charge times	Modeling cumulative variable characteristics
total_fast _charge	Total fast charge times	Modeling cumulative variable characteristics
total_low _charge	Total slow charge times	Modeling cumulative variable characteristics
cycle	Cycles	Modeling cumulative variable characteristics
over_in_ charge	Overcharge times	Modeling cumulative variable characteristics
over_out _charge	Over-discharge times	Modeling cumulative variable characteristics
times_of _deep_ch arge	Deep charge times	Modeling cumulative variable characteristics
times_o-ver_char ge	Overcharge times	Modeling cumulative variable characteristics
SoH	Battery Health Index	y (label)

2) Model Selection

a) Linear Regression Model (Ridge and Lasso):

The problem of the linear regression model is that since the relationship between parameters and variables is linear, the result may get extremely high or low when the data are not within the regular range; thus, the accurate rate significantly depends on the dataset itself, so the performance is unstable and the model and the parameter is not generalizable. To solve this problem, we choose to apply gradient-boosting-decision-tree-based models.

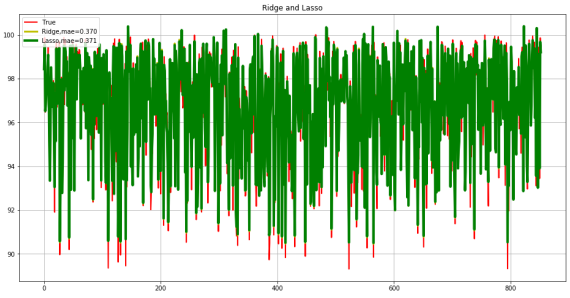

Fig. 2. MAE of Ridge and Lasso

b) Gradient Boosting Decision Tree based models (XGBoost,LightGBM and CatBoost):

Gradient Boosting Decision Tree is more beneficial to handle outliers by using strong loss functions. And XGBoost, LightGBM, and CatBoost optimized distributed gradient boosting library. XGBoost provides a parallel tree boosting [5], LightGBM supports parallel and GPU learning [6], and CatBoost enables categorical variable to have metrics [7]. Their performance is in Fig. 3-5.

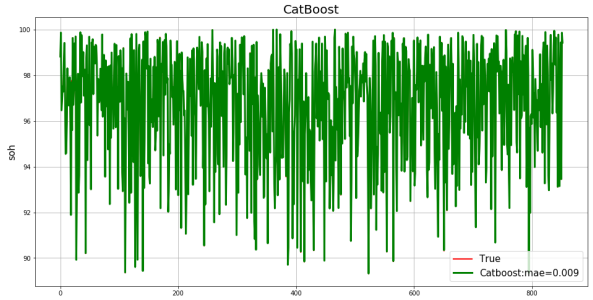

Fig. 3. MAE of CatBoost

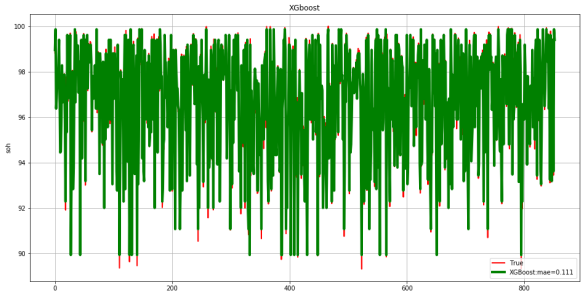

Fig. 4. MAE of XGBoost

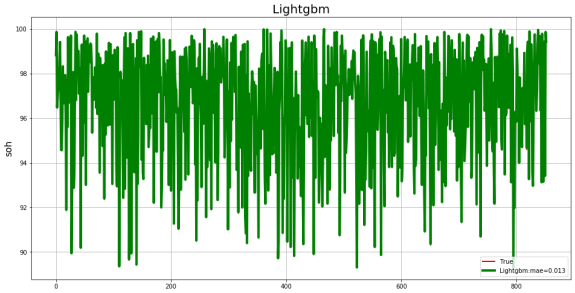

Fig. 5. MAE of LightGBM

c) Deep Neural Network:

We use a regular three-layer neural network with 256 units at layer one, 128 units at layer two, and 64 units at layer three using Rectified Linear Unit as the activation function. The batch size is 128, with 3000 iterations.

We use a learning rate decay technique to dynamically adjust the learning rate because when the optimization reaches a particular bottleneck, the current learning rate is no longer suitable for optimization; relatively speaking, when the learning rate is larger, the step gets large, and as a result, it might be

impossible to reach the bottom, so the learning rate needs to be reduced.

The learning rate is set 0.1 at the beginning, and if the loss does not decrease after 100 iterations, the learning rate is halved.

The mean absolute error of this deep neural network is 0.034, which is not ideal comparing to CatBoost and LightGBM.

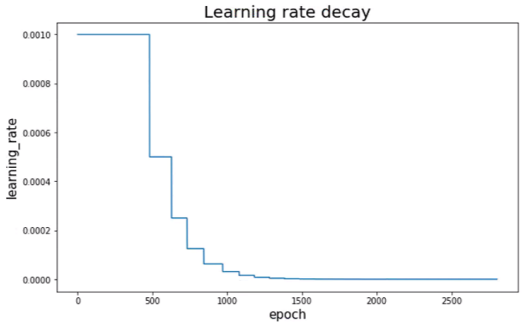

Fig. 6. Learning Rate Decay

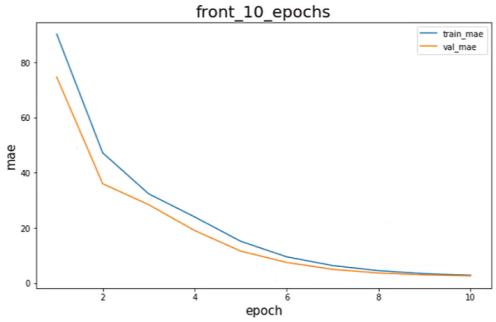

Fig. 7. MAE of front 10 epochs

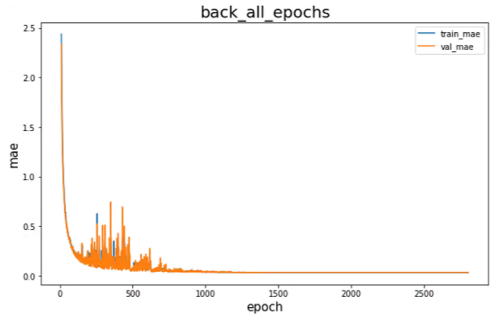

Fig. 8. MAE of all epochs

d) Conclusion

The MAE of each model is shown in Table III. CatBoost has the best performs and then ranked LightGBM. However, the difference between them is insignificant, and there is a big gap between them and other models, so we combine both models by taking the average of outputs to get a better result.

TABLE III. THE MEAN ABSOLUTE EERROR OF EACH MODEL

Comp-arison	Model Names					
	Ridge	***Lasso***	***XGBoost***	***LightGBM***	***CatBoost***	***DNN***
MAE	0.370	0.371	0.111	0.013	0.009	0.034

By performing our combined models on two real vehicles, we get the result in Table IV. The error is insignificant.

TABLE IV. REAL VEHICLE VERIFICATION

Index	Vehicle 1	Vehicle 2
Real Value[a]	97.115%	93.510%
Predicted Value	96.947%	93.452%
Error	0.169%	0.058%

[a] Real value = Full power mileage / Rated mileage

Algorithm 1. Real Time SoH Estimation

Input: $\mathbf{X} = \begin{bmatrix} X_{11} & \dots & X_{1n} \\ \dots & \dots & \dots \\ X_{m1} & \dots & X_{mn} \end{bmatrix}$

Output: SOH

1. Preprocess **X**
2. **function** TRAIN(**DATA**):
3. min loss = inf
4. patience = 0
5. **while** patience < 5 **do**
6. model1 = send **DATA** to CatBoost for training
7. model2 = send **DATA** to Lightgbm for training
8. Calculate average loss function for these two models
9. **if** loss > min loss **do**
10. patience += 1
11. **end if**
12. **end while**
13. **return** model1, model2
14. **end function**
15. Select soc, mile, cycle….and other features of **X** as the new input **X'**
16. **for** i=1 to 10 **do**
17. Split **X'** into **train_cv**, **valid_cv**, **test_cv** randomly according to 6:3:1
18. models = TRAIN (**train_cv**)
19. loss = models.evaluate (**valid_cv**)
20. **end for**
21. Calculate the average loss to tune various hyperparameters
22. Input **test_cv** data to model
23. models = TRAIN (**train_cv**)
24. output = models.predict (**test_cv**)
25. Calculate SOH by Equation in line(24)

C. Establishment of Prediction Model for Future Battery Health

For future state of health prediction, our goal is to obtain a decay curve of SoH. This problem falls a time series prediction, which acknowledges the continuity of the development of things, use past time series data for statistical analysis, and infer the development trend of things [8]. We choose the Long Short-Term Memory (LSTM), one of the most advanced time series prediction models to solve the problem.

We aim to analyze the trend of SoH, so the only feature of the model is SoH. However, instead of using direct SoH data, we use SoH difference for each time interval. The benefice to do so is that no matter what interval the SoH falls in, the difference

we get always within a small range, thus the distribution will not be influenced. In this case, the training set, validation set, and test set will share the similar distribution range.

For the set division of sequence prediction problem, simply randomly dividing the training set and test set will make the obtained model less accurate. Here we use every 30 training data after extracting a test data to ensure that at every stage of SoH decline, there are data to verify its accuracy.

To ensure that the model parameters are not more than the sample size, the models are designed small for such small sample size. And the model structure can be seen at Algorithm 2, the pseudocode. The final MSE is 0.021.

Algorithm 2. SoH Future Decay Prediction

Input: $\mathbf{X} = \begin{bmatrix} X_1 \\ \dots \\ X_m \end{bmatrix}$

Output: SOH Feature Decay Vector = $[SOH_1 \quad \dots \quad SOH_k]$

```
Preprocess X
function TRAIN(DATA):
    min loss = inf
    patience = 0
    while patience < 3 do
        model = send DATA to LSTM for training
        Calculate loss function
        if loss > min loss do
            patience += 1
        end if
    end while
    return model
end function
for i=1 to 5 do
    Split X into train_cv, valid_cv, test_cv randomly according to 6:3:1
    model = TRAIN (train_cv)
    loss = model.evaluate (valid_cv)
end for
Calculate the average loss to tune various hyperparameters
Input test_cv data to model
model = TRAIN (train_cv)
output = model.predict (test_cv)
Calculate SOH future decay by Equation in line(22)
```

IV. Conclusion

We have demonstrated a method of estimating the current real-time SoH and predicting the future trend, the result of both goals is satisfying, suggesting that both models perform well on processing such dataset. The combination of CatBoost and LightGBM shows a low mean absolute error when predicting the current real-time, and this model is easier to adjust the parameters, thus more applicable for a different dataset. The defect of the ampere-hour integral method is covered, and it is possible to know the current SoH using such a model. For future trend prediction, long short-term memory network is proved to have an excellent performance. By constructing such a model, we enable the car company to have more control over the new energy lithium battery state so relevant adjustment can be achieved.

Acknowledgment

The work is supported by Beijing Automotive Group Co., Ltd. and Shanghai Cloud Ready Technology Co., Ltd.

References

[1] General Administration of Quality Supervision, Inspection and Quarantine of the People's Republic of China, "Technical specifications of remote service and management system for electric vehicles - Part 3: Communication protocol and data format 电动汽车远程服务与管理系统技术规范第 3 部分:通信协议及数据格式, " Beijing, Aug 2016.

[2] M. Kipness, "1188-1996 - IEEE Recommended practice for maintenance, testing, and replacement of valve- regulated Lead-Acid (VRLA) batteries for stationary applications," PE/ESSB - Energy Storage & Stationary Battery Committee, 1996.

[3] V. Pop, H. J. Bergveld, P. H. L. Notten and P. P. L. Regtien, "State-of-the-art of battery state-of-charge determination," Measurement Science and Technology, Dec 2005.

[4] W. Xiao, W. Zhong, X. Shu, J. Yan and X. Yuan, "Battery state of health (SoH) estimation method and application based on big data 基于大数据的电池健康状态（SoH）的估算及应用," J Automotive Safety and Energy, vol. 10, No. 1, 2019.

[5] L. Prokhorenkova, G. Gusev, A. Vorobevand, A. V. Dorogush and A. Gulin, "CatBoost: unbiased boosting with categorical features," Neural Information Processing Systems Conference, 2019.

[6] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[7] G.Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T. Y. Liu, "LightGBM: a highly efficient gradient boosting decision tree," Neural Information Processing Systems Conference, 2017.

[8] D. Wu, "Management decision method-theory, model and application 管理决策方法——理论、模型与应用, " CA: Hehai University Publish, 2014.

[9] C. Vidal, P. Malysz, P. Kollmeyer and A. Emadi, "Machine learning applied to electrified vehicle battery state of charge and state of health estimation: state-of-the-art," IEEE Access Access, 2020.

[10] J. He, Z. Wei, X. Bian and F. Yan, "State-of-Health estimation of lithium-Ion batteries using incremental capacity snalysis based on voltage–capacity model," IEEE Trans. Transp. Electrific., 2020.

[11] D. Zhou, Z. Li, J. Zhu, H. Zhang and L. Hou, "State of health monitoring and remaining useful life prediction of lithium-Ion batteries based on temporal convolutional network," IEEE Access Access, 2020.

[12] Y. Wu, Q. Xue, J. Shen, Z. Lei, Z. Chen and Y. Liu, "State of health estimation for lithium-ion batteries based on healthy features and long short-term memory," IEEE Access Access, 2020.

[13] P. Li, Z. Zhang, Q. Xiong, B. Ding, J. Hou, D. Luo, Y. Rong and S. Li, "State-of-health estimation and remaining useful life prediction for the lithium-ion battery based on a variant long short term memory neural network," Journal of Power Sources, vol. 459, May 2020.