2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Applications

An Accurate and Interpretable Lifetime Prediction Method for Batteries using Extreme Gradient Boosting Tree and TreeExplainer

Fu Jiang[1], Youzhi He[1], Dianzhu Gao[2]*, Yun Zhou[3], Weirong Liu[1], Lisen Yan[1], Jun Peng[1]
[1] School of Computer Science and Engineering, Central South University, Changsha, China
[2] School of Automation, Central South University, Changsha, China
[3] School of Information Technology and Management, Hunan University of Finance and Economics, Changsha, China

***Abstract*—Lithium-ion batteries have been widely used in many fields such as electric vehicles and smart grid. Accurately predicting its lifetime is crucial for ensuring safety and accelerating battery technological development. This study aims to develop an interpretable battery lifetime prediction method based on a machine learning model by explaining the features used in the model. Firstly, the battery charge-discharge cycle data is analyzed, and five key features related to lifetime are extracted from the discharge curve of the first 100 cycles. Then, extreme gradient boosting tree (XGBoost) is built to learn the relationship between the features and the lifetime, and its optimal parameters are obtained through grid search and five-fold cross-validation. TreeExplainer is used to calculate shapley value based on game theory to quantitatively interpret and reveal the important features contributing to lifetime. Experimental results on the latest battery dataset demonstrate that XGBoost can effectively predict battery lifetime. At the same time, quantitative analysis and interpretation provide the reasons for the model decision, which improves the credibility of the prediction method and even helps to have a deeper understanding of the battery degradation mechanism.**

***Index Terms*—lithium-ion batteries; lifetime; interpretability; extreme gradient boosting tree; shapley value**

I. INTRODUCTION

Due to the advantages of low cost, high energy density, and long lifetime, lithium-ion batteries have been widely used in electric vehicles, smart grids, etc [1]. However, due to internal chemical reactions and external environmental factors, the battery performance will gradually degrade as the use time increases. Although safety factors have been considered in the design of the battery, there have been severe explosion accidents occasionally in recent years [2]. Accurately predicting battery lifetime is very important to ensure personal and property safety, and can even accelerate the development of battery technology [3].

Existing studies for battery lifetime prediction mainly fall into two categories: model-based methods and data-driven methods [4]. Model-based methods first use available data to build an empirical model of battery performance degradation, and then use filtering algorithms to update model parameters in real time to perform online predictions [5], [6]. However, due to the complexity of the battery structure and the variability of the external operating environment, it is very difficult to establish an accurate mathematical model to describe the battery degradation mechanism. On the contrary, the data-driven method is based on machine learning technology to directly learn the mapping relationship between features and lifetime, which is mechanism-free [7]. This method is very attractive when the amount of data is large.

With the growth of computing power and training data, more and more machine learning methods have been proposed to predict battery lifetime, such as elastic net [8], gradient boosting decision tree (GBDT) [9], neural network [10], stacked denoising autoencoder [11], convolutional neural network (CNN) [12], etc. However, with the increasing complexity of the model, although the prediction accuracy has been improved to varying degrees, the interpretability of the results will continue to decline [13]. Like the GBDT, its internal involves multiple trees and the decision path of the tree is complex, and the prediction results obtained from it are usually difficult to interpret. Deep learning methods such as CNN are even more a black box model [14]. Better interpretability is considered to be one of the four desired attributes of battery lifetime prediction methods [8]. It can not only help judge the rationality of the model's work, but in some scenarios, it can also help discover the hidden physical principles behind the task.

In view of the importance of the interpretability of battery lifetime prediction, it has attracted the attention of some scholars. The literature [15] mentioned that considering that many machine learning methods are black boxes, and battery lifetime prediction requires a certain understanding of the prediction results, simple and direct linear regression methods are more likely to be favored. The literature [16] believes that the development of interpretable prediction methods and the improvement of prediction accuracy should be carried out at the same time, and proposes partial least square regression to predict battery lifetime. Although the above methods obtain a certain degree of interpretability by using simple models, when the amount of data is large and the relationship between features and lifetime is complicated, this may sacrifice accuracy. It is still an important challenge how to ensure high accuracy while also making the prediction results have good interpretability.

Recently, a field called explainable artificial intelligence (XAI) [17] has developed various techniques to interpret com-

*Corresponding author: Dianzhu Gao, Email: gaodianzhu@csu.edu.cn.

978-1-6654-9457-1/21/$31.00 ©2021 IEEE
DOI 10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00164

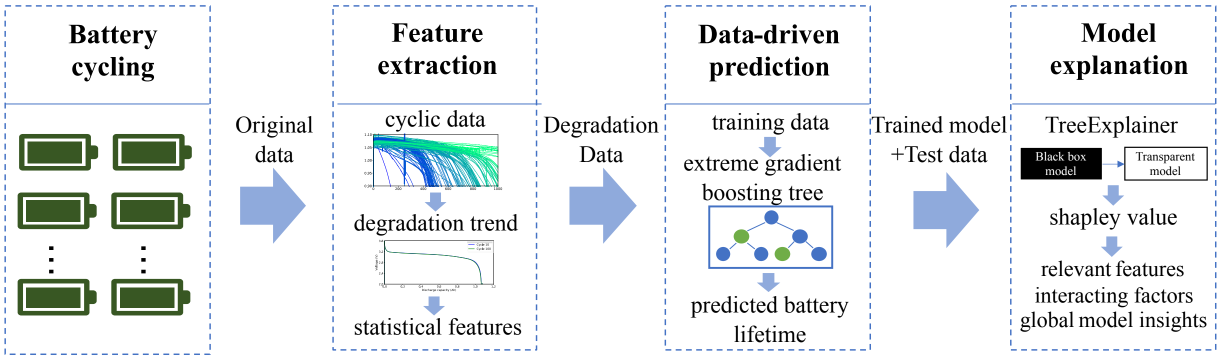

Fig. 1. The workflow of the interpretable battery lifetime prediction method proposed in this paper. A: The public battery dataset is used in this paper, which is generated by 124 lithium-ion batteries being cyclically charged and discharged. B: By analyzing the original dataset, the degradation trend of the battery is found, and then the statistical features are calculated to generate the degradation data. C: Degradation data is used to train the extreme gradient boosting tree prediction model. D: Considering the poor interpretability of the model, this paper uses TreeExplainer to explain the prediction model in order to obtain high-accuracy and well-interpretable lifetime prediction results at the same time.

plex machine learning models to enhance their interpretability. Among them, a type of post-hoc explainability techniques that use additional methods to explain existing models is particularly concerned by researchers. For example, Lundberg et al. proposed TreeExplainer [18] based on shapley value of game theory, which can explain the individual decisions of a tree-based model in terms of input contributions. Considering that the battery degradation dataset is a kind of tabular-style data, and the tree-based model has very good predictive performance on this type of data, even better than the deep learning model [19]. Therefore, combining the tree-based model and TreeExplainer, it is possible to achieve both high accuracy and good interpretability in battery lifetime prediction.

According to the above discussion, this paper proposes an accurate and interpretable battery lifetime prediction method using extreme gradient boosting tree (XGBoost) and TreeExplainer, in order to achieve high accuracy and good interpretability prediction results at the same time. Among them, the XGBoost is a high-performance realization of the tree-based model, which has superior performance in capturing the nonlinear relationship between the feature and the target and the prediction accuracy [19].

In order to verify the effectiveness of the method proposed in this paper, the latest battery public dataset [8] is used for experiments, which includes the cyclic charge and discharge data of 124 batteries. We analyzed the change trend in the battery degradation process, and extracted five features that are most relevant to lifetime. Then we divide the 124 batteries data into two parts: training set and test set. Five cross-validation and grid search are used to obtain the optimal hyperparameters of XGBoost and the comparison method. The prediction performance of XGBoost is compared with elastic net and support vector machine. The experimental results show that the former has better generalization performance, and thus achieves better prediction performance. We continue to use TreeExplainer to explain the prediction results of XGBoost. TreeExplainer gives the contribution of all the features on a single battery sample to the lifetime. For the entire training set sample, it gives the global impact of the features on the lifetime and the interaction relationship between the features. The interpretation provided helps us to better grasp the reason why the model makes decisions, and even to have a deeper understanding of the degradation mechanism of the battery.

In general, the main contributions of this paper are as follows:

1) A complete workflow of battery lifetime prediction based on machine learning is proposed, including four parts: battery degradation data analysis, feature extraction, model training and prediction, and model explanation.

2) The interpretability method combining XGBoost and TreeExplainer was applied to battery lifetime prediction for the first time, eliminating the trade-off of accuracy and interpretability in battery lifetime prediction to a certain extent.

3) Extensive experimental results on the latest battery dataset prove the effectiveness of the proposed method.

The rest of this paper is outlined as follows. The workflow of the interpretable battery life prediction method based on XGBoost and TreeExplainer is presented in Section 2. The experimental results are demonstrated in Section 3. Section 4 concludes this paper.

II. Materials and Methods

We will introduce the accurate and interpretable battery lifetime prediction method proposed in this article from the four aspects of data set, feature selection, prediction method and interpretation method according to the order of description in the workflow Fig. 1.

First, the generation process of the battery dataset used is described, including the battery model, charging and discharging protocol. Then, The selection of degradation features is introduced. These degradation features are extracted from the original dataset to generate degradation data, which are used in subsequent model training and interpretation.

Then, the XGBoost prediction model is presented. The model integrates multiple decision trees and the tree structure is more

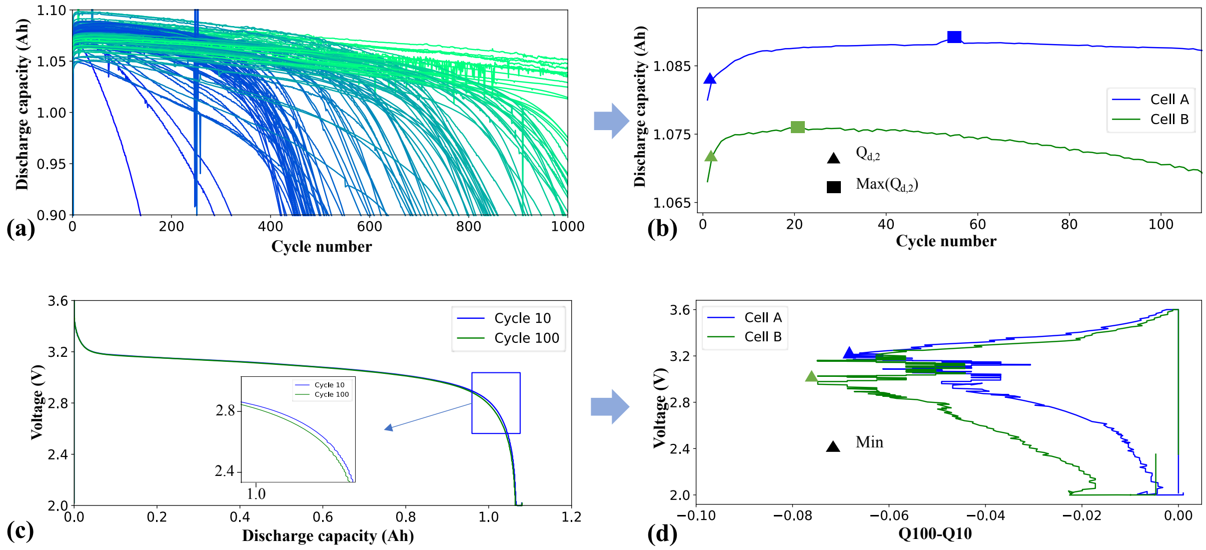

Fig. 2. Dataset description and feature selection. (a), discharge capacity vs cycle number curve, each battery is represented by a curve. The difference in battery lifetime is distinguished by the color of the curve. The green curve indicates a longer lifetime, the blue is the opposite. It can be seen that there are obvious differences in the cycle lifetime of different batteries, and the lifetime ranges from 150 cycles to 2300 cycles; (b), derived from (a), the first 100 cycles in (a). Two batteries numbered "b1c9" and "b2c11" are displayed; (c), the 10th and 100th cycle voltage vs discharge capacity curve of a single battery; (d), derived from (c), difference of the discharge capacity curves as a function of voltage between the 100th and 10th cycles.

complex, which makes it difficult to interpret the prediction results, so it is usually considered a black box model. Finally, The main idea of TreeExplainer is introduced. By inputting the prediction model and data into the TreeExplainer, it gives the contribution of the feature to the prediction result, thus helping us better understand the prediction.

A. Dataset Description

The largest battery dataset so far is used in this article. The dataset is generated by 124 commercial high-power LFP/graphite cells. The nominal capacity of each battery is 1.1Ah and the nominal voltage is 3.3V. These batteries are charged from 0% to 80% of the state of charge (SOC) using 72 different one-step and two-step charging strategies. All cells were subsequently discharged with a constant current-constant voltage (CC-CV) discharge at 4 C to 2.0 V. The cycle data of all batteries is shown in Fig. 2(a).

B. Feature Selection

In this paper, the method of extracting battery degradation features refers to the literature [3]. All the features are taken from the discharge data in the first 100 cycles of the battery, which allows its lifetime to be predicted earlier. They are: $2-cycle$, $max-2$, $minimum$, $variance$, $skewness$. These features all come from the discharge curve. The first two are from the discharge capacity vs cycle number curve, as shown in Fig. 2(b). $2-cycle$ is the discharge capacity at the cycle 2, $max-2$ is the difference between the maximum discharge capacity of the first 100 cycles and the cycle 2. The last three are from the voltage vs Q100-Q10 curve, as shown in Fig. 2(c),(d). The three features of $minimum$, $variance$, and $skewness$ extracted from the curve are the minimum value, variance and skewness of the curve on the x-axis.

C. Prediction Method

In this paper, we use the XGBoost as the battery lifetime prediction method because of its strong nonlinear fitting ability and its wide use in many data science competitions. For XGBoost, its task is to learn the mapping relationship between features and battery lifetime from the training dataset. Then, when a new battery sample data is obtained, it can predict its lifetime.

The following is a brief introduction to XGBoost. First of all, XGBoost is a supervised learning method that requires well-labeled training data. Then, XGBoost is a tree ensemble method, which is internally composed of multiple classification and regression trees (CART), as shown in the Fig. 3(a). The prediction result of the model is jointly determined by all decision trees. Let $D=\{(x_i,y_i)\}_{i=1}^{n}$ be the training set, where x_i is a vector of m features and $y_i \in \mathbb{R}$ is the label value. The predict value for an instance x_i is defined as follows

$$\hat{y}_i=\sum_{k=1}^{K} f_k\left(x_i\right), f_k \in \mathcal{F} \tag{1}$$

Among them, K is the total number of decision trees, f is a function in the function space F, and F is all possible decision trees.

XGBoost uses a training method called boosting, as its name implies. The training method is: fix what has been learned so far, and then add a new tree to eliminate the previous residual. Assuming that the $t-th$ tree is currently being learned, the

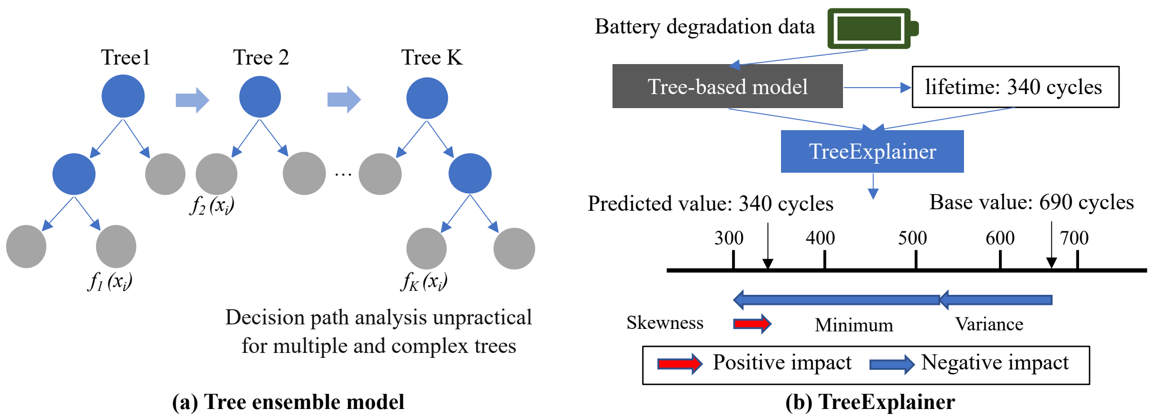

Fig. 3. Tree ensemble model and TreeExplainer. (a) The prediction result of the tree ensemble model is jointly determined by multiple trees. Because the decision path analysis of multiple trees is impractical and the tree structure is complex, the predictions made by the model are considered not interpretable; (b) TreeExplainer provides an interpretation based on the trained model and battery data by giving the contribution of features to lifetime in the form of shapley value.

predicted value $\hat{y}_i^{(t)}$ at this step t is determined by the predicted value $\hat{y}_i^{(t-1)}$ of the previous $t-1$ trees and the result $f_t(x_i)$ of the $t-th$ tree. As shown in the following

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k\left(x_i\right) = \hat{y}_i^{(t-1)} + f_t\left(x_i\right) \tag{2}$$

In order to learn the desired f_t, an objective function needs to be defined to judge whether the current tree is good. The objective function is composed of a loss term and a regularization term. The loss term is used to measure the difference between the predicted value and the true value, and the regularization term is used to prevent the model from overfitting. The objective function can be expressed by the following

$$obj = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{i=1}^{t} \Omega\left(f_i\right) \tag{3}$$

Among them, l is the loss function, Ω is the regularization function. XGBoost uses Taylor's quadratic expansion to obtain the specific loss term. The regularization term is obtained by calculating the value of the leaf node. When XGBoost is learning each tree, it starts from the root node and uses the objective function to determine whether the current node should be split.

The above is the construction and training of the XGBoost model. In addition, XGBoost also optimizes the storage structure in its implementation, which makes it have a faster training and prediction speed.

D. Explanation Method

We use the TreeExplainer to explain the prediction of the XGBoost model. TreeExplainer is a general method that can explain a single prediction result of a tree-based model in a way of feature contribution. Its core idea is derived from the calculation method of Shapley value in coalitional game theory [20]. In a coalitional game, Shapley value is worth a theoretically proven way to reasonably distribute reputation among members in the alliance. Drawing on this idea in machine learning, the alliance is a set of interpretable features, and the Shapley value is the contribution of a reasonable distribution of features to the predicted results. By building on the classic game-theoretic Shapley value, TreeExplainer has a solid theoretical foundation and various desirable attributes.

The calculation method of feature contribution in TreeExplainer is introduced as following. In XGBoost, for each sample, the model will give a predicted value. TreeExplainer calculates the contribution of each feature to the prediction result. This value is called the Shapley value. Assuming that the current sample $x_i = [x_{i1}, x_{i2}, ..., x_{iN}]$, there are a total of N features, and the predicted value given by the model is y_i, then the feature contribution $g(x_{ij})$ satisfies the following equation:

$$y_i = y_0 + \sum_{j=1}^{N} g\left(x_{ij}\right) \tag{4}$$

Among them, y_0 is the average value of the predicted values obtained from all training data, which exists as a base value. When $g(x_{ij}) > 0$, it indicates that the j_{th} feature will increase the predicted value on the basis of the base value.

The lifetime prediction and interpretation of a single battery are taken as the example to analyze the input and output of TreeExplainer to better understand its principle, as shown in the Fig. 3(b).

III. Experiment And Discussion

In this section, we conduct experiments on the battery dataset to prove the prediction accuracy and interpretability of the proposed method. First of all, through comparative

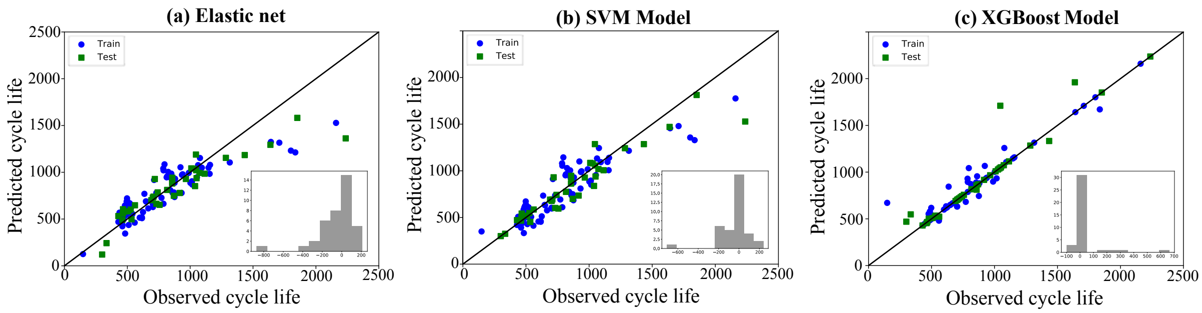

Fig. 4. The prediction results of the three methods on the test set and training set: (a) Elastic net; (b) SVM model; (c) XGBoost model. The histogram in the scatter chart is the distribution of prediction errors. Overall, the best results, especially the prediction results of long-lifetime batteries, were obtained on XGBoost.

TABLE I
THE OBTAINED OPTIMAL PARAMETERS OF XGBOOST.

Parameters	Value
n_estimators	251
min_child_weight	3
max_depth	2
learning_rate	0.45

TABLE II
COMPARISON OF THE PREDICTION RESULTS OF THE THREE METHODS.

Dataset	Method	RMSE
Train set	elastic net	169
	SVM	146
	XGBoost	83
Test set	elastic net	188
	SVM	151
	XGBoost	129

experiments, the superior predictive performance of XGBoost is reflected. Then, by analyzing the rich interpretation results given by TreeExplainer, our method is proved to have better interpretability.

A. Experimental Setup

This experiment was run on an Intel Core i5 laptop with 16GB of RAM. All code runs on PyCharm. The programming language uses Python 3.7. Scikit-learn packages are used for building machine learning models. The SHAP open source library is used to calculate shapley value and its visual interpretation, and its version is v3.90.

B. Prediction Performance

In order to evaluate the prediction performance of XGBoost, elastic net and support vector machine (SVM) are used for comparative experiments. The root mean square error (RMSE) is selected as the evaluation method, as shown in Formula 5. The smaller the RMSE value, the smaller the prediction error of the model, and therefore the better the prediction performance.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{5}$$

The original dataset contains 124 batteries. In this article, they are divided into training set (70%) and test set (30%). The training set is used to train the machine learning model, and the test set is used to verify the accuracy of the model. All data sets are used by TreeExplainer to provide model interpretation.

The distribution of battery lifetime varies greatly. In order to make the generalization performance of the model stronger, we use a random division method, and then take the division result with a more uniform lifetime distribution to train the model. Grid search and 5 cross-validation are used in the three models at the same time to obtain the optimal hyperparameters. Among them, the optimal hyperparameters of XGBoost are shown in Table. I.

The prediction results of the three methods are shown in Fig. 4 and Table. II. In the Fig. 4, we observe that although the three models have achieved good accuracy in the lower battery lifetime, when the lifetime is greater than 1500, the deviation between elastic net and SVM is relatively large, and XGBoot can still achieve accurate prediction result. The results of the Table. II show that the smaller RMSE of XGBoost on the training set allows it to learn a more accurate relationship between features and lifetime, so that it also achieves a smaller error on the test set without overfitting.

The above results show that XGBoost has superior accuracy in battery lifetime prediction. However, compared to elastic nets and SVM, the prediction results given by XGBoost are difficult to explain to some extent because of the complex internal structure. Therefore, we need to provide more information to explain the reasons behind the model.

C. Model Explanation

We have used XGBoost to get superior battery lifetime prediction results. However, this is not enough because it is difficult for us to know what the model is based on. The interpretation of the model is important, especially when the lifetime of a battery is very low or very high, we would like

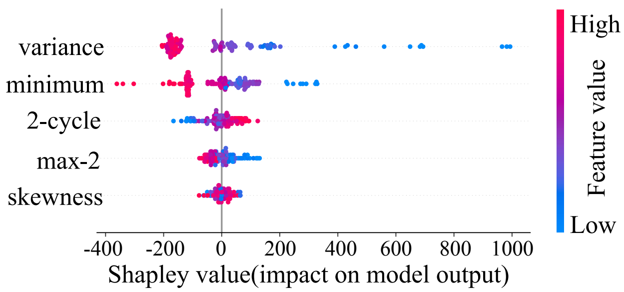

Fig. 5. Shapley summary plot. Each point represents the shapley value of the feature on a single sample. The y-axis is the features, which are sorted from top to bottom according to their importance. The x-axis is the shapley value of the feature. The color bar on the right indicates the size of the feature.

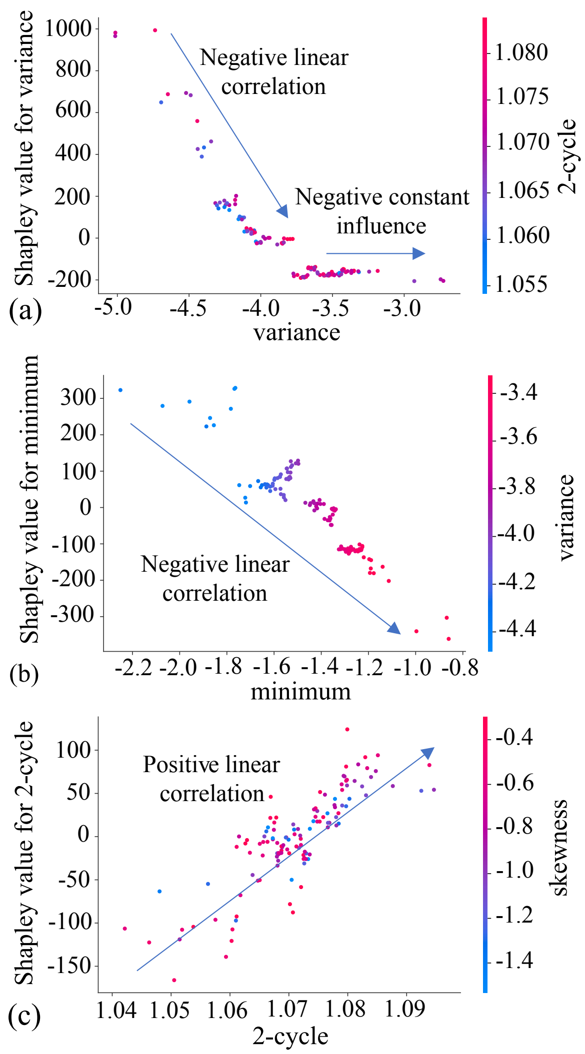

Fig. 6. Shapley value dependence plot of the feature. On the basis of Fig. 5, the relationship between features and their contribution to lifetime is depicted in more detail. At the same time, other features that have the strongest correlation with the features below are also placed on the right side of the figure.

to know the reason behind it, such as which abnormal feature caused it. In addition, the overall impact of the features on the overall battery lifetime and the relationship between the features are also what we need to pay attention to.

In order to improve the interpretability of the model to obtain more information about battery lifetime degradation, TreeExplainer is used to show the importance of features and the contribution of features to lifetime, which are obtained by calculating the shapley value. In this section, we will use three types of diagrams to illustrate the interpretation provided by TreeExplainer.

Fig. 3(b) shows the contribution of features to individual predictions. The current lifetime of this battery is predicted to be 340 cycles. The predicted value is obtained by the combined effect of the base value and the feature contribution. The base value is the average of the lifetime predictions of the training samples, which is 690 cycles in this article. The contribution of the feature is represented as a red and blue arrow in the figure. Red indicates a positive impact on lifetime, while blue indicates a negative impact. The length of the arrow is proportional to the feature contribution. For this battery, the $minimum$ and $variance$ features have a large negative impact on the lifetime, while $skewness$ has a small positive impact. The overall impact is negative, so the lifetime of this battery is very small.

In addition to explaining the prediction of a single battery, by calculating the shapley value of all the features of all training samples, we can get the overall impact of all the features on the battery, as shown in Fig. 5. Each point represents the shapley value of a feature to a single battery. The y-axis represents the feature, and the x-axis represents the shapley value. All features are sorted from top to bottom according to importance, and the color indicates the value of the feature. It can be observed that the importance of the five features of $variance$, $minimum$, $2-cycle$, $max-2$, and $skewness$ decreases in order. Meanwhile, it can also be found that some features have a strong correlation with lifetime. For example, when the battery has a smaller variance value, it may have a longer lifetime.

In order to get a more detailed relationship between features and lifetime, a scatter plot is used, which also provides the relationship between features, as shown in the Fig. 6. We have chosen the three most important features as examples. In Fig. 6(a), the relationship between variance and lifetime is not always linearly related. When the variance is less than -4.0, the two are negatively linearly correlated, and when it is greater than -4.0, the variance has a constant effect on the lifetime. For the other two features, minimum is a stable negative linear correlation, and 2-cycle is a stable positive linear correlation. It should be noted that the features shown on the right side of the figure have the strongest correlation with the features below. In Fig. 6(b), we see that the minimum has the strongest correlation with the variance. When the minimum becomes larger, the variance color becomes red, which shows that its value is also increasing.

In general, TreeExplainer can provide two benefits. First of all, the provided features on the lifetime prediction contribution of a single battery sample and the degree of influence on the entire training sample can allow users to have a basic understanding of the importance of all features, which undoubtedly increases the credibility of the model. Moreover, the feature importance provided can help us choose the best feature subset to train the model. Secondly, XGBoost has a strong nonlinear fitting ability, it can learn the complex relationship between features and lifetime. By using TreeExplainer to interpret the results of XGBoost, perhaps we can find more hidden features and relationships between features on more complex battery dataset. The newly discovered features can help build better battery prediction models. The hidden more complicated relationship between features and lifetime can even help better understand the degradation mechanism of batteries.

IV. Conclusions And Future Work

This paper proposes an accurate and interpretable method for predicting battery lifetime. We use XGBoost to obtain highly accurate battery lifetime prediction results. Then, we explored using TreeExplainer to explain why XGBoost made the decision. The interpretation provided by TreeExplainer allows us to better understand the complex relationship between the proposed features and battery lifetime.

In the future, we will improve our work from two aspects. Firstly, XAI will be used to explore the relationship between more unknown degradation features and lifetime. Secondly, we will try to introduce transfer learning to improve the lifetime prediction method to adapt to more complex environments.

Acknowledgment

This work is supported by National Natural Science Foundation of China (Grant No. 62172448) and Natural Science Foundation of Hunan Province (Grant No. 2021JJ30868).

References

[1] X. Hu, L. Xu, X. Lin, and M. Pecht, "Battery lifetime prognostics," *Joule*, vol. 4, no. 2, pp. 310–346, 2020.

[2] J. Zhang and J. Lee, "A review on prognostics and health monitoring of li-ion battery," *Journal of power sources*, vol. 196, no. 15, pp. 6007–6014, 2011.

[3] P. M. Attia, A. Grover, N. Jin, K. A. Severson, T. M. Markov, Y.-H. Liao, M. H. Chen, B. Cheong, N. Perkins, Z. Yang *et al.*, "Closed-loop optimization of fast-charging protocols for batteries with machine learning," *Nature*, vol. 578, no. 7795, pp. 397–402, 2020.

[4] Z. Fei, F. Yang, K.-L. Tsui, L. Li, and Z. Zhang, "Early prediction of battery lifetime via a machine learning based framework," *Energy*, vol. 225, p. 120205, 2021.

[5] C. Hu, B. D. Youn, and J. Chung, "A multiscale framework with extended kalman filter for lithium-ion battery soc and capacity estimation," *Applied Energy*, vol. 92, pp. 694–704, 2012.

[6] X. Zheng and H. Fang, "An integrated unscented kalman filter and relevance vector regression approach for lithium-ion battery remaining useful life and short-term capacity prediction," *Reliability Engineering & System Safety*, vol. 144, pp. 74–82, 2015.

[7] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Remaining useful life estimation–a review on the statistical data driven approaches," *European journal of operational research*, vol. 213, no. 1, pp. 1–14, 2011.

[8] K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggedakis *et al.*, "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, vol. 4, no. 5, pp. 383–391, 2019.

[9] F. Yang, D. Wang, F. Xu, Z. Huang, and K.-L. Tsui, "Lifespan prediction of lithium-ion batteries based on various extracted features and gradient boosting regression tree model," *Journal of Power Sources*, vol. 476, p. 228654, 2020.

[10] Y. Zhang, Z. Peng, Y. Guan, and L. Wu, "Prognostics of battery cycle life in the early-cycle stage based on hybrid model," *Energy*, vol. 221, p. 119901, 2021.

[11] F. Xu, F. Yang, Z. Fei, Z. Huang, and K.-L. Tsui, "Life prediction of lithium-ion batteries based on stacked denoising autoencoders," *Reliability Engineering & System Safety*, vol. 208, p. 107396, 2021.

[12] J. Hong, D. Lee, E.-R. Jeong, and Y. Yi, "Towards the swift prediction of the remaining useful life of lithium-ion batteries with end-to-end deep learning," *Applied energy*, vol. 278, p. 115646, 2020.

[13] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1721–1730.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[15] M.-F. Ng, J. Zhao, Q. Yan, G. J. Conduit, and Z. W. Seh, "Predicting the state of charge and health of batteries using data-driven machine learning," *Nature Machine Intelligence*, vol. 2, no. 3, pp. 161–170, 2020.

[16] P. M. Attia, K. A. Severson, and J. D. Witmer, "Statistical learning for accurate and interpretable battery lifetime prediction," *arXiv preprint arXiv:2101.01885*, 2021.

[17] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[18] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature machine intelligence*, vol. 2, no. 1, pp. 56–67, 2020.

[19] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[20] L. S. Shapley, *A value for n-person games*. Contributions to the Theory of Games (AM-28), Volume II, 1953.