A comparative study of machine learning based modeling methods for Lithium-ion battery

Peng Wang, Jie Fan*, Yang Ou, Zhe Li, Yi Wang, Bo Deng, Yuanwei Zhang, Zihao Gao.
China Automotive Engineering Research Institute Co., Ltd., Chongqing 401122, China

*Corresponding author: Jie Fan. E-mail address:384954890@qq.com.

Abstract. A suitable battery model plays an important role in assisting accurate state estimation for power battery used in electric vehicles. This paper compares the applications of four commonly used machine learning methods (decision tree, k-nearest neighbour, support vector machine and neural network) in lithium-ion battery modeling. The adaptability on working condition, temperature and degradation of above four modeling methods are analysed in detail. Results show that neural network performs best when working condition changes. All the models basically have the same performance on adaptability to temperature. The battery dynamic characteristics change significantly in the aging process and it is necessary to include battery test data under different degradation levels into training sets as to obtain a model that can predict the voltage response accurately in various aging states.

1. Introduction
To realize safe and efficient operation of the battery, monitoring of its states like state of charge (SOC) and state of health (SOH) is especially necessary[1]. However, due to the impossibility of direct measurement of these state variables, battery models, which can depict the battery dynamic characteristics are needed to assist state estimation with only measurable signals like current, voltage and temperature. Therefore, an accurate battery model is of great importance for management strategy to prolong the battery’s lifetime while alleviating its safety concerns.

Traditional battery modeling methods include electrochemical models, electric circuit models, analytical models, etc[2]. Recently, with the development of sensing and information technology, data-driven methods have attracted popular attention due to its strong capability in data mining and simplicity in system modeling. Compared with traditional modeling methods, data-driven methods can construct the model of complicated systems without previous knowledge about their mathematical or physical properties. The only thing needed is data, which can be collected during systems’ various stages like design, simulation, operation and maintenance.

Machine learning, as a strong tool of data-driven methods, has attracted widespread attention and found its applications in battery modeling[3]. This paper conducts a comparative study of four commonly used machine learning methods including DT (decision tree), KNN (k-nearest neighborhood), SVM (support vector machine) and NN (neural network) in lithium-ion battery modeling. They are compared using cell datasets acquired under different temperatures, different testing cycles and different levels of degradation. The remainder of this paper is organized as follows: Section 2 gives a general introduction to the four machine learning methods utilized in this study.
Section 3 briefly presents the information of the battery experimental datasets. Section 4 discusses the model comparison results and section 5 concludes the whole paper.

2. Basic knowledge of the utilized machine learning methods
This section intends to give a general description of the four machine learning methods utilized in this paper. Because of limited space, we only highlight the key ideas of each algorithm.

2.1. Decision tree
As a powerful and popular prediction model for both classification and regression, DT constructs the map from features to labels based on a tree-like topology. Each internal node in DT represents a feature while each leaf node stands for a label. When judging the label for a new sample, one just needs to place the sample into one specific leaf node according to its features and decision criteria of the internal nodes in the trained DT. Then label of the leaf node that the sample is attributed is just exactly its predicted label. The main hyper-parameter in DT is its depth.

2.2. K-nearest neighbor
KNN is originally an intuitive classification method that has been widely used in pattern recognition. With little modification, it can also be applied effectively for regression purpose. In this paper, KNN estimates the label of a new testing point as an average of the labels of the \( k \) closest training points. Euclidean distance is taken as criterion to define the similarity between the testing point and training points. In order to avoid distance bias brought by different magnitude of different features, normalization is necessary in the data pre-processing. The only hyper-parameter in KNN is \( k \), which decides how many nearest training points are used to make prediction.

2.3. Support vector machine
SVM is a kind of supervised machine learning method and widely used by researchers or engineers both from academia and industry due to its advantages in high dimensional pattern recognition and non-linear function approximation. Basically, the support vector regression is to find a linear function
\[
\hat{y} = \langle w, \phi(x) \rangle + b
\]
Eq.(1)
as flat as possible in the way that the predicted value \( \hat{y} \) has at most \( \varepsilon \) deviation from the real label \( y \), where \( \phi(\cdot) \) is a function that transforms the input data non-linearly into a higher dimensional feature space. In this study, the Gaussian kernel is utilized for feature space transformation and defined as:
\[
k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \exp(-\gamma |x_i - x_j|^2)
\]
Eq.(2)
Support vector regression has three hyper-parameters: \( \varepsilon \), \( \gamma \) and \( C \) (regularization parameter) that need to be optimized, which makes it more complicated compared with DT and KNN.

2.4. Neural networks
A classical fully-connected structure of NN is mainly consisted of input layer, hidden layers and output layer. Each unit in a layer is called neuron, which generates output by weighting inputs from its previous layer and applying specified activation function. The target of NN training is to find a set of suitable weighting parameters that can minimize the error between training samples’ real and predicted labels[4]. Generally, more hidden layers will improve the fitting capacity of NN while also increasing the training computational burden and making NN have higher possibilities of over-fitting. Traditional method for training NN is through error back-propagation, which will result into a long training time when the training set is large. In order to accelerate the training process, ADAM, a method for stochastic optimization is adopted here.
3. Battery experiment introduction
The battery test data is generated in the test bench consisting of an Arbin BT2000 tester, a thermal chamber, a computer for user-machine interface and a switchboard for cable connection. The voltage, current, temperature of each cell are recorded at the sampling time of 10Hz. The tested batteries were 8 LiNMC battery cells with 0.94Ah nominal capacity, 3.7V nominal voltage. Each cell experienced impedance test and characterization tests (including static capacity test, hybrid pulse power characteristic (HPPC) test, resistance test, dynamic stress test (DST) and federal urban dynamic schedule (FUDS) test) under three different temperatures (10℃, 22℃ and 35℃) in different levels of degradation. More details about the battery experiment are available in Ref.[5].

4. Comparison results
In this section, first we will explain the input features for the model. Then the ten-folder cross validation method used to optimize the hyper-parameters in DT and KNN is briefly introduced. Hyper-parameters selection in SVM and structure of NN are also described. Finally, the adaptability to working condition, temperature and degradation level of the four machine learning based modeling methods is discussed in detail. Training time of the four modeling methods is also compared to evaluate their computational burden.

4.1. Feature selection
In this paper, battery current and SOC generally act as the input features for the models while voltage is the output. Despite more complicated features like integration or derivative of current can be constructed to enrich the features, we don’t include them for the following reasons. First, the computational capacity of battery management system (BMS) is very limited. Thus introducing features that cannot be directly obtained from BMS will increase its computational burden and propose higher demands both on its hardware and software resources. Second, this study focuses on the comparison of the four modeling methods rather than deliberately establishing a perfect model for each machine learning based method. Such simple features are enough for comparison and are just appropriate to evaluate the fitting capability of the method when input is very raw and limited.

4.2. Hyper-parameters selection
Ten-folder cross validation is used to find the optimal hyper-parameters for DT and KNN. It first divides the data set into 10 sections which are similar in size and variable distribution and then picks out one section as testing set and the remaining as training set for 10 rounds. Thus 10 groups of training and testing sets are obtained and used for training and validation for 10 times. The average performance on the testing set over the 10 rounds experiments is regarded as the final evaluation result for the model. The hyper-parameter set with the best performance is optimal.

4.3. Adaptability to working conditions
In this part, the HPPC data set is used to train the model, while the FUDS and DST data sets serve as testing set to evaluate the model’s adaptability to working condition. The input features include battery current and SOC, while the output is battery voltage. In order to avoid interference brought by temperature and battery degradation, only the data under 10℃ of the fresh battery is used.

In order to access the performance of different modeling methods, a benchmark is needed for comparison to verify the models’ effectiveness. The reference row in Table 1 lists out the differences between the training set and testing sets (as the same in Table 2 and 3). For example, the root mean squared error (RMSE) between HPPC and FUDS, DST are 333mV and 283mV respectively. It needs to be mentioned that interpolation is needed to calculate RMSE because of the different data length of training data and testing data. More specifically, to calculate the RMSE between HPPC and FUDS, we apply cubic interpolation to expand the HPPC data to the same length as FUDS and then calculate the RMSE between FUDS and expanded HPPC data.
Figure 1  Results of decision tree’s adaptability to working condition: (a) HPPC  (b) FUDS  (c) DST
Figure 2  Results of k-nearest neighbor’s adaptability to working condition: (a) HPPC  (b) FUDS  (c) DST

Figure 3  Results of support vector machine’s adaptability to working condition: (a) HPPC  (b) FUDS  (c) DST
Figure 4  Results of neural network’s adaptability to working condition: (a) HPPC  (b) FUDS  (c) DST

Figure 1-4 show the four different model’s predicted voltage verses real voltage and Table 1 compares their training error, testing error and training time. The error defined here is the RMSE (root mean square error) between real and predicted voltage. It can be seen that all the models can reach relatively low error over training set, which implies all the models have enough fitting capacity to capture the dynamic properties demonstrated in HPPC test. Among them, DT reaches lowest training error because the depth of the tree here is 13, which is optimized by 10 folder cross validation method mentioned before and such a depth enables DT reach a relatively elaborate division of the feature space. Despite the complex structure of NN deployed here, it seems that the fitting capacity of NN is inferior to DT due to its higher training error over HPPC test data, which is a little contrary to the common description that NN can approximate any function with high accuracy as long as it possesses enough hidden layers. Explanation for higher training error of NN is not because its hidden layers is insufficient but we deliberately stop the training procedure before its parameters reach the global optimal. The advantages of this trick include lowering the over-fitting risks of NN and save the corresponding training time. As can be seen from the testing errors over FUDS and DST, NN obtains best generalization performance due to its drastically lower error compared with other three modeling methods. However, although DT gets lowest training error, its testing error over DST is about 3 times that of NN, which implies its over-fitting situation and relatively poor predicting performance over new samples that are not included in the training set. From Table 1 and Figure 2-3, both KNN and SVM perform well over training set while terribly over testing set. This phenomenon reflects these two modeling methods are very sensitive to the change of battery working conditions. In addition, the testing error of KNN and SVM mainly appears at the late period of FUDS and DST when the battery is nearly exhausted of its SOC, which is apparent in Figure 2(b)(c) and Figure 3(b)(c). Especially for
SVM, its estimated voltage basically divergence near the end of the charge. Analysis shows that the testing errors over FUDS for KNN and SVM when SOC is above 0.3 are 46mV and 82mV respectively, which is also relatively desirable. It indicates that KNN and SVM cannot generalize the dynamic characteristics of the battery in low SOC range given different current loads. But these two modeling methods are still of value considering battery SOC seldom reaches below 0.2 in real-world electric vehicle applications.

| Modeling method | Training error | FUDS | DST | Training time |
|-----------------|----------------|------|-----|--------------|
| DT              | 10mV           | 49mV | 141mV| 0.11s        |
| KNN             | 17mV           | 166mV| 250mV| 1.69s        |
| SVM             | 55mV           | 125mV| 154mV| 1630s        |
| NN              | 32mV           | 42mV | 53mV | 17.69s       |

The configuration of the experiment computer is i5-6400 CPU @ 2.7GHz and 8G RAM. All programmes are written in Python language. The training time for the modeling methods is listed in the rightest column of Table 1. Among the four methods, DT has least training time of 0.11s. Considering its satisfied performance over testing sets, DT is a good balance between accuracy and computational burden. The training time of NN is 17.69s, which is also desirable considering its excellent generalization ability to different working conditions. For SVM, it not only performs terribly over the testing set when SOC is low, but also requires a long time for training (in this case, training the SVM model takes nearly half an hour given the hyper-parameters specified before). Thus, SVM is the least machine learning method that we would recommend in battery modeling.

In conclusion of this part, DT is most recommended for its desirable generalization capacity and lowest demanding on computational resources. However, for more accurate modeling of the battery, NN is the best choice despite its relatively longer training time. SVM and KNN is not recommended because of their relatively severer sensitiveness to working conditions.

4.4. Adaptability to temperature

In this part, we will discuss the different models’ robustness to the temperature. Figure 5 illustrates the voltage response of the fresh battery and aging battery in FUDS test under 3 different temperatures. It can be seen that the voltage between 22℃ and 35℃ for the fresh battery is highly resembling. Though the voltage difference enlarges under 10℃, it is still not distinct enough. However, after 1024 aging cycles, there exist obvious distinctions among the voltage responses under the three different temperatures. In order to make the following comparison more tangible, we use the aging battery data to test the models’ adaptability to temperature.

Table 2 Comparison of the four machine learning methods’ adaptability to temperature

| Modeling method | Training error | 22℃ | 35℃ | Training time |
|-----------------|----------------|-----|-----|--------------|
| DT              | 28mV           | 56mV| 119mV| 0.08s        |
| KNN             | 23mV           | 64mV| 118mV| 0.09s        |
| SVM             | 39mV           | 52mV| 107mV| 290.21s      |
| NN              | 39mV           | 59mV| 112mV| 11.47s       |
Table 2 show the comparison results. In this case, all the models achieve desirable performance on the testing sets, especially for KNN and SVM, which arise big errors when working condition changes. Contrary to the conclusion in section 4.2, SVM demonstrates best generalization performance when temperature changes and it even outperforms NN by 7 mV over FUDS and 5mV over DST. Generally, all the models are not very sensitive to the temperature variation.

There is still one point worth of mention here. Because the battery voltage under low temperature drops faster than under high temperature, it will reach cut-off value sooner in 10℃, which can be reflected by the phenomenon in Figure 5 that duration for discharging process under 10℃ is around 1.8 hours while under 22℃ and 35℃ are about 2.2 and 2.5 hours. In this paper, we train the model using the data under 10℃, which is least in data amount, while achieving good predicting results on relatively larger data sets under 22℃ and 35℃. This implies the information provided by the training data under 10℃ is enough for the machine learning models to capture the battery’s inherent characteristics (or the relationship between battery current, SOC and voltage) that is not sensitive to the temperature variation and enables accurate voltage response prediction under whatever temperature.

Like in section 4.3, training SVM costs longest time while DT still enjoys lightest computation burden. Due to the reduction in training data amount, training time for all the models decreases compared with section 4.3. It needs to be mentioned that the training time decreases from nearly half an hour to less than 5 minutes when the training data length shrinks from 3 hours to around 1.8 hours, which indicates less data is greatly beneficial for reducing the training time of SVM.

4.5. Adaptability to degradation
In this part, different models’ adaptability to degradation is investigated. Figure 6 shows the FUDS battery voltage responses under 3 different degradation levels. It can be seen that as battery ages, the discharging time from full-charged state to energy-depletion decreases gradually (when the battery is fresh, the discharging time is around 4 hours. While it decreases to around 3 hours and 2 hours after 555 and 1024 aging cycles respectively), which implies the battery capacity shrinks in its degradation process.
Here, the fresh battery test data are used to train the model, while the aged battery test data are utilized to evaluate the models’ generalization capacity as battery degrades. The input features are still battery current and SOC, while the output is battery voltage.

Due to the same performance of the four modeling methods, only results of decision tree are demonstrated in Figure 7. Table 3 concludes all the models’ performance. It can be seen that both DT and KNN are over-fitting because their training errors are only 1mV while the testing errors are much larger.
higher. Due to the training data amount is quite large, the training time for SVM takes above half an hour. Despite its efforts paid to training process, SVM still doesn’t show better prediction results on testing sets. Basically, no model achieves desired performance because the testing errors of different models roughly equal to the reference values, which means all the methods cannot predict the dynamic behaviors of the battery under aged states only with the information provided by the fresh battery test data. This phenomenon can be interpreted as there exists significant changes to the battery dynamic characteristics during battery’s aging process and it is necessary to train the models with battery data under different degradation levels so as to enable models capture the characteristic changes happened with battery aging.

| Modeling method | Training error | 555 cycles | 1024 cycles | Training time |
|-----------------|----------------|------------|-------------|---------------|
| DT              | 1mV            | 79mV       | 115mV       | 0.30s         |
| KNN             | 1mV            | 81mV       | 115mV       | 0.15s         |
| SVM             | 15mV           | 76mV       | 114mV       | 1951.95s      |
| NN              | 23mV           | 92mV       | 130mV       | 22.88s        |

5. Conclusion
This paper compares four commonly used machine learning methods applied in battery modeling and discussed their adaptability to working conditions, temperature and degradation. Results show that NN possesses the strongest robustness to working conditions with a rational computational burden. DT achieves both desirable prediction results and little computational demand when working condition varies. All the models perform roughly the same on the adaptability to temperature. SVM outperforms the other modeling methods with minor improvement in accuracy while requiring more training time. To obtain a model that can reflect the battery dynamic characteristics under different levels of degradation, the training set must include battery test data under different aging states because battery characteristics changes significantly in its degradation procedure.

Acknowledgements
The authors acknowledge the Innovation Support Program for the Leader in Science and Technology of Chongqing under Grants (010267), Project No.010513 and CEVE (China Electric Vehicle Evaluation Procedure) for financial support.

References
[1] Fan J, Zou Y, Zhang X and Guo H 2019 A novel State of Health estimation method for Lithium-ion battery in electric vehicles J. Phys. Conf. Ser. 1187
[2] Jie Fan Y Z and Zhang X 2019 Quantifying Electric Vehicle Battery’s Ohmic Resistance Increase Caused by Degradation from On-board Data IFAC-PapersOnLine 52 297–302
[3] Zhao R, Yan R, Chen Z, Mao K, Wang P and Gao R X 2019 Deep learning and its applications to machine health monitoring Mech. Syst. Signal Process. 115 213–37
[4] Shen W X, Chau K T, Chan C C and Lo E W C 2005 Neural network-based residual capacity indicator for nickel-metal hydride batteries in electric vehicles IEEE Trans. Veh. Technol. 54 1705–12
[5] Hu X, Li S and Peng H 2012 A comparative study of equivalent circuit models for Li-ion batteries J. Power Sources 198 359–67