Università della Svizzera italiana

Stochastic Methods 2022

Final Project Report

Li-Ion Battery State-of-charge Estimation with Stochastic Methods

Abstract

Battery powered electric cars are crucial for reducing the green house gases when compared to internal combustion engined vehicles. However, the technical aspects of operating and predicting the battery levels to calculate the range offers a fundamental challenge. Therefore, this article aims to estimate and create an infrastructure for optimally calculating the remaining state-of-charge capacity of Li-Ion batteries by using one of the NASA's well-known data sets with numerous famous stochastic methods available on the literature as well as on the market.

Student
Mert Alagözlü

Student 2
Muhammet Uslu

Student 3
Shi Xuetian

1 Introduction & Definition

Battery Electric (BEVs) and Hybrid Vehicles (HEVs) become popular for a variety of reasons. The major grounds for their adoption are because they utilize less or no fossil fuel (no carbon dioxide emissions). Since the transport industry produces the bulk amount of greenhouse gas emissions and pollution to the environment, the transport sector can be improved by the introduction of the e-mobility applications. Even for BEVs registered today, the results reveal that BEVs have by far the lowest life-cycle emissions. Emissions from average medium-size BEVs registered today are already 66 % –69 % lower than equivalent gasoline vehicles in Europe and 60 % –68 % lower in the United States. [10] However, this technology has several obstacles, such as complicated and difficult to correct state of battery health estimate, which is impacted by a variety of battery and user behaviors. [7]
Nowadays, most BEV manufacturers use Li-Ion batteries, which take the biggest portion of vehicle architecture. Therefore, batteries need to be controlled, diagnosed and monitored by sophisticated systems, so-called Battery Management Systems (BMS). [2] it is critical to have an accurate and reliable assessment of state-of-charge (SoC) and state of health (SoH) under various environmental circumstances. This enables better BMS performance for optimal battery pack usage in a variety of operating scenarios for a better user experience. [7] Our research on this topic showed that prognostics is one of the most important functions of BMS, and it is found that some scientific works[5] proposed that stochastic methods such as Random Forests [6] have promising results in comparison to physically determined battery depreciation models.

2 Presentation of Li-Ion Battery Data set

The data set was gathered by NASA Ames Prognostics Center of Excellence's custom-built battery prognostics test bed (PCoE). At various temperatures, Li-Ion batteries were put through three different operating profiles (charge, discharge, and impedance spectroscopy) represented in table 2. Discharges were performed at various current load levels until the battery voltage fell below predetermined voltage thresholds. Repeated charge and discharge cycles cause the batteries to age faster. Other generic experimental parameters are represented in table 1 with additional details.
At room temperature, four Li-Ion batteries were put through three distinct operating profiles (charge, discharge, and impedance). Charging was done in constant current (CC) mode at 1.5A until the battery voltage reached 4.2V, then in constant voltage (CV) mode until the charge current decreased to 20mA. Discharge was carried out at a constant current (CC) level of 2A until the battery voltage dropped to 2.7V, 2.5V, 2.2V, and 2.5V for different batteries. The impedance was measured using impedance spectroscopy(EIS) frequency sweep from 0.1Hz to 5kHz. Repeated charge and discharge cycles accelerate battery aging, whereas impedance measurements give insight into internal battery properties that vary as the battery ages. The tests were terminated when the batteries met end-of-life (EoL) requirements, which was a 30% decline in rated capacity (from 2Ahr to 1.4Ahr). This information may be used to forecast the remaining charge (for a specific discharge cycle) as well as the Remaining Useful Life (RUL).

Generic Parameters	
Cycle	Top level structure array containing the charge, discharge and impedance operations
Type	Operation type, can be charge, discharge or impedance.
Ambient Temperature	Ambient temperature (degree C)
Time	The date and time of the start of the cycle, in MATLAB date vector format.
Data	Data structure containing the measurements

Table 1. Dataset Characteristic Parameters

Charge		**Discharge**		Impedance	
Voltage_measured	Battery terminal voltage (Volts)	Voltage_measured	Battery terminal voltage (Volts)	Sense_current	Current in sense branch (Amps)
Current_measured	Battery output current (Amps)	Current_measured	Battery output current (Amps)	Battery_current	Current in battery branch (Amps)
Temperature_measured	Battery temperature (degree C)	Temperature_measured	Battery temperature (degree C)	Current_ratio	Ratio of the above currents
Current_charge	Current measured at charger (Amps)	Current_charge	Current measured at load (Amps)	Battery_impedance	Battery impedance (Ohms) computed from raw data
Voltage_charge	Voltage measured at charger (Volts)	Voltage_charge	Voltage measured at load (Volts)	Rectified_impedance	Calibrated and smoothed battery impedance (Ohms)
Time	Time vector for the cycle (secs)	Time	Time vector for the cycle (secs)	Re	Estimated electrolyte resistance (Ohms)
		Capacity	Battery capacity (Ahr) for discharge till 2.7V	Rct	Estimated charge transfer resistance (Ohms)

Table 2. Data Structure of Charge, Discharge, and Impedance

3 Methodology & Implementation

In this benchmark, there are 12 stochastic methods. Coarse, medium, fine, bagged, boosted tree, linear regression, linear step wise regression, linear, quadratic, cubic support vector regression, random forest and neural networks. Along with brief explanation of the methods, their respective performances and optimal hyperparameters are also investigated.

3.1 Ridge Regression

Similar to linear regression, Ridge and Lasso regression are some simple techniques to reduce model complexity and prevent over-fitting that may result from simple linear regression (susceptible to outliers), where residual's desired to be minimized too with a similar formula: $\mathbf{y} - ||\mathbf{y} - \mathbf{bX}||_2$. In ridge regression n represents instance, m represents features. In ridge regression, cost function's altered by adding a coefficient to the sum of weights, which means coefficients are restricted with λ in equation 5 because as λ increases, weights are needed to be closer to zero to compensate the cost. So, ridge regression shrinks the coefficients, and it helps to reduce the model complexity. As you can also notice, if λ is zero, our cost function becomes linear regression. It is easy to deduce a point where reducing λ lead to generalized linear regression behavior. [12]

$$y = w_1 x_1 + w_2 x_2 + ... + w_n x_n \tag{1}$$

$$\beta = X^+ y \tag{2}$$

$$\beta = (X^T X)^{-1} X^T y \tag{3}$$

$$minimize \sum_{i}^{n} (y_i - \tilde{y_i})^2 \tag{4}$$

$$\mathbf{y} - ||\mathbf{y} - \mathbf{bX}||_2 + \lambda \sum_{j}^{m} w_j^2 \tag{5}$$

Given some scarce prior knowledge about battery SoC, this portion of the study also predicts how much battery capacity will depreciate using ridge regression. Predicted lines are plotted with different lambda values. Since ridge regression enables us to predict desired outputs using multi-co-linearity, up until the wasted battery life in 6.54%, it is predicted red dashed line and then, interpolated the result until 3600 seconds. However, as it can be seen, approximately 6 % prior knowledge about the discharge isn't enough to make correct prediction because SOC of the battery depreciates rapidly after 1000 seconds. It's crucial to select a portion of the data set to mimic the rest of the data. In this case, prior knowledge up until 1500 seconds will give better results. However, the more prior knowledge it requires, the more latency in estimation of RUL, which reduces the usability and practicality of the specific application, such as in BEVs. Inability to predict SOC rapidly and correctly can further cause problems such as unnoticed energy shortage to reach a destination. In this case, the prediction's overly optimistic and it'll cause energy shortage to reach an intended target in BEVs. Now, on the second subplot, with approximately 19 % usage as prior knowledge, the prediction is better than the 6.54 %, but this doesn't quite fit to our needs since the user would be used almost 19.05% of the RUL to know how much time left for the application be ceased.

3.2 Classification and Regression Trees(CaRT)

CaRT recursively subdivides the sample space to perform classification or regression locally in each partition. Subdivision is achieved by binary tree structure, leaf nodes perform the actual task on the samples. Fundamentally, regression returns the average of the node samples, on the other hand, classification returns the relative frequency per class.[1]For CaRT training each mode has two parameters to learn; one axis aligned split along one dimension and maximum height of tree, parameter update follows the following training steps: Finally, automatic hyperparameter optimization by using Matlab $fitrtree$ function. according to hyperparameter optimization application. [8] There are two optimizable hyperparameters: Maximum number of splits and Split criterion, which is $Gini$ index function.

[1]Note : CaRT have tendency to over-fit, because trees with high training accuracy oftentimes subdivides the sample space into too small regions.[3]

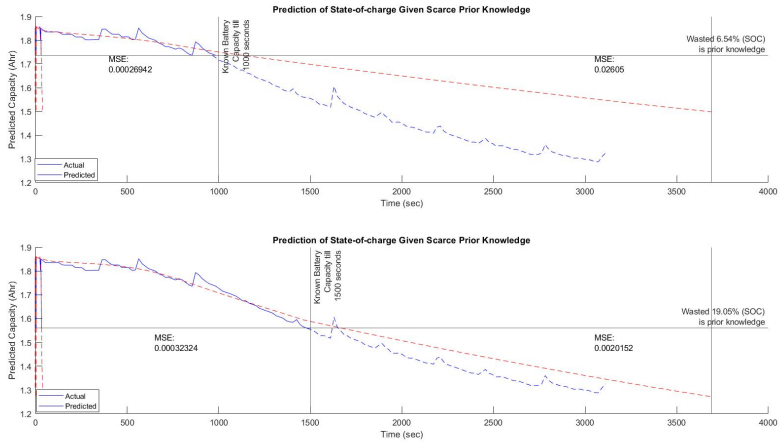


Figure 1. Scarce prior knowledge prediction

Algorithm 1 CaRT Training steps

1) Grow trees from root to leaves.

1.1) At each node, perform an exhaustive search for the best split with respect to the objective functions below.

while Regression Tree **do**

Use Squared Loss Function :

$$Q_m(T) = \sum_x (y_i - y_m)^2 \tag{6}$$

end while

while Classification Tree **do**

Use Impurity Function :

$$\underbrace{Q_m(T) = \sum_k^K P_k * (1 - P_k)}_{\text{Gini Index}} \quad \textbf{or} \quad \underbrace{Q_m(T) = \sum_k^K P_k \log P_k}_{\text{Negative Cross Entropy}} \tag{7}$$

end while

Overall Objective Function:

$$minimize \quad \frac{|S_l|}{N} * Q_l(T) + \frac{|S_r|}{N} * Q_r(T) \tag{8}$$

3.3 Random Forest

Random Forest forms an ensemble (average votes from individual trees) of randomized CaRTs to counter over-fitting. [6] [3] Each tree is a weak learner (better than guessing) and the ensemble is strong. There are three randomization components:

Bagging Trains each tree on a random subset of the training data

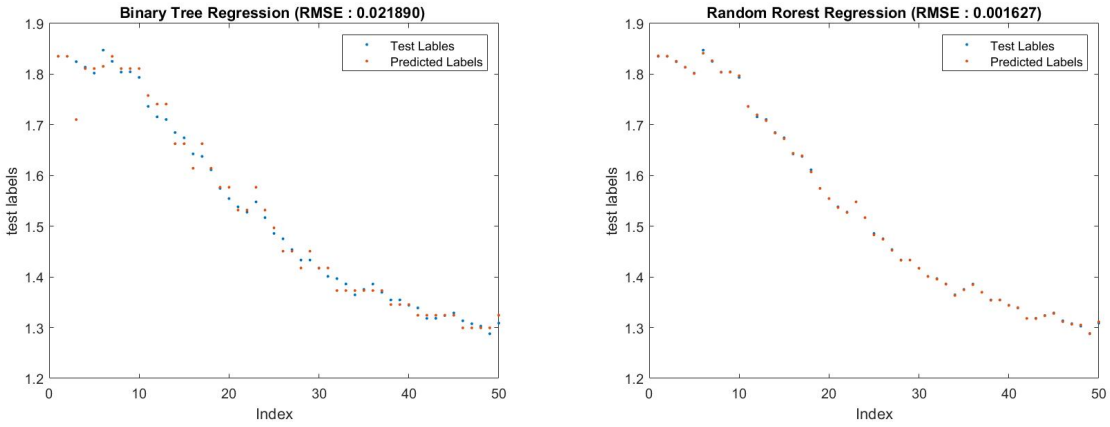
Random Subspace Projections Per node, searches only in d random dimensions for the best split.

Extremely RF Selects best split from randomly drawn split candidates. [3]

Random Forests have multiple hyperparameters that are automatically optimized by $fitrensemble$ function, Least-Squares Boosting (LSBoost) is used as a learning method as in the figure 2, additionally, bagged tree is implemented because of the one of the randomized aforementioned components. The next hyperparameter is inherited from binary tree structure, where the maximum number of splits is 154. Learning rate is selected as 0.18482 according to the optimal model. The number of predictors to sample may be tuned in order to save the resources, in other words, some of the least important features can be dropped out for less computation cycles and for saving memory. For instance, it is calculated that feature importance scores represented in table 3.

Measured Voltage	0.0091
Measured Current	0.0110
Load Voltage	0.1644
Load Current	0.0058
Temperature	0.0007
time	0.0008

Table 3. Feature Importance Coefficients



(a) Binary Tree

(b) Random Forest

Figure 2. Tree Based Regression Methods

3.4 Support Vector Regression

Support Vector Regression (SVR) is a supervised learning approach for predicting discrete values. [11] Support Vector Regression operates on the same principles as SVMs. SVR's primary concept is to identify the optimum fit line. The best fit line in SVR is the hyperplane with the greatest number of points. Unlike other Regression models, which attempt to minimize the difference between the real and predicted values, the SVR attempts to fit the best line within a certain threshold value. The threshold value is the distance between the hyperplane and the boundary line. [11]

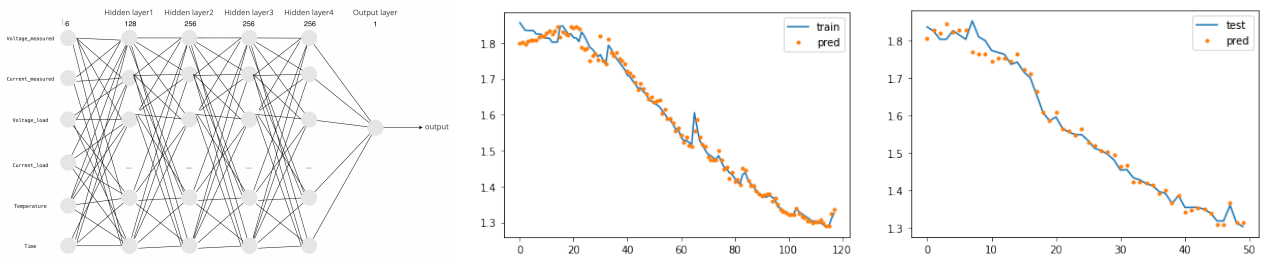
Linear SVR or SGD Regressor is used for large data sets, because linear SVR is quicker than SVR, but it just takes into account the linear mapping kernel. A kernel is a collection of mathematical functions that accept data as input and turn it into the desired form. [11] These are typically used to locate a hyperplane in higher-dimensional space. In this article, there are three different kernel functions linear, quadratic, and cubic kernel function to represent the inner product of sample space in different dimensions.

Hyperparameters are determined to be the most optimal ones within each different kernel function with the help of the regression learner app. [9] In standardized SVR, epsilon values are selected as 0.028, where epsilon defines residuals lower than the selected value have no effect and imposing no effect. The kernel scale value is set to 1 because our model has an appropriate scale and needs no further re-scaled flexibility imposed on the kernel model. If one desires to create more flexible data to represent with this model, the kernel scale should be set to lower than 1, which is likely to increase the root-mean-square error of the model. On the other hand, the box constraint is set to 0.283, where it controls the penalty imposed on observations with large residuals. Therefore, increasing it results in a more flexible model.

3.5 Neural Network

Neural networks are architectures with interconnected nodes that function similarly to neurons in the human brain. [4] It can detect hidden patterns and correlations in raw data, as well as cluster and categorize it, and – over time – continuously learn and improve itself. [4] However, it has black box properties, which means the inner working principles of a neural network aren't fully comprehensible. To predict the SoC, forward propagation, ReLU activation function, back propagation, MSE loss function, and gradient descent method: Adam methods are combined.

Our model results for training data and test data are shown in figure 3. The yellow points are prediction of capacity and the blue line is the true labels. It can be observed that the prediction of the training data set is generally consistent with the true values of training data; thus, the model reasonably behaves.



(a) Structure of NN (b) Model results (c) Test results

Figure 3. Neural Network

Hyperparameters are determined among many choices to obtain the best performance. We choose to learn rate to be $1e-4$, since if the learning rate is smaller, the model will converge very slowly or even does not converge, a larger learning rate allows the model to learn faster, but at the cost of arriving at a suboptimal final set of weights.

4 Experimental Results

	COARSE TREE	MEDIUM TREE	FINE TREE	BAGGED TREE	BOOSTED TREE	LINEAR REGRESSION	LINEAR REGRESSION STEPWISE	LSVM	QSVM	CSVM	RANDOM FOREST	NEURAL NETWORKS
RMSE	0.0728	0.0260	**0.0168**	0.0176	0.0674	0.0231	**0.0166**	0.0241	0.0232	0.0253	0.0188	0.0707
Training Time (sec)	**0.613**	0.629	**0.643**	1.584	1.089	4.464	8.725	8.484	8.306	7.952	6.743	5.60
Prediction Speed (obs/sec)	**19000**	22000	**18000**	4900	4900	2100	2300	9500	9400	9100	5427	6250
Hyper-parameters	-Min. Leaf size: 36 -No surrogate splits	-Min. Leaf size: 12 -No surrogate splits	-Min. Leaf size: 4 -No surrogate splits	-Min. leaf size: 8 -# of learners: 30	-Min. leaf size: 8 -# of learners: 30 - Learning rate: 0.1	-Linear	-Max. # of steps: 1000 -Linear	-Linear Kernel function -Standardize data	-Quadratic kernel function -Standardize data	-Cubic kernel function -Standardize data	Automatically tuned parameters during training learning rate : 0.18482 Number of predictors to sample : 6 Ensemble method: Bagging	-Adam -ReLU -Back Propagation -Learning rate: 1e-4 - Epochs: 1000

Table 4. Final Results

Among other stochastic methods, two of which have the least error accumulated as linear regression and fine regression tree. However, the training time of both strictly differs from each other. The training time of fine regression tree is 0.629 seconds. In comparison, linear step-wise regression takes 8.725 seconds for training, even though their root mean squared errors (RMSE) is almost identical. From the aspect of prediction time, it is crystal clear that fine regression tree dominates linear step-wise regression method. In other words, the prediction time of fine regression tree is approximately 7.82 times of step-wise regression method. Among SVR family, RMSE of LSVM, QSVM, and CSVM result in similar, where LSVM has slightly better performance than other SVM methods. SVM families generally have almost the half performance in predicting samples as regression trees. It is observed that SVM family has exceptionally high training time compared to other methods except for random forest. The training time of SVM family is around 13.12 times of training time of regression trees. If one desires a multi-sensor-fusioned-data with high bandwidth for estimating the SOC of a Li-Ion battery configuration, the low prediction time should be favorable decision as in regression trees. The code can be reachable from GitHub repository. [1]

References

[1] Alagözlü, Mert and Uslu, Muhammet and Xuetian, Shi. Battery-SoC-Estimation.

[2] V. Chandran, C. K. Patil, A. Karthick, D. Ganeshaperumal, R. Rahim, and A. Ghosh. State of charge estimation of lithium-ion battery for electric vehicles using machine learning algorithms. *World Electric Vehicle Journal*, 12(1), 2021.

[3] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Found. Trends. Comput. Graph. Vis.*, 7(2–3):81–227, feb 2012.

[4] M. Y. Kiang. Neural networks. In H. Bidgoli, editor, *Encyclopedia of Information Systems*, pages 303–315. Elsevier, New York, 2003.

[5] C. Li, Z. Chen, J. Cui, Y. Wang, and F. Zou. The lithium-ion battery state-of-charge estimation using random forest regression. pages 336–339, 2014.

[6] C. Li, Z. Chen, J. Cui, Y. Wang, and F. Zou. The lithium-ion battery state-of-charge estimation using random forest regression. pages 336–339, 2014.

[7] S. S. Madani, R. Soghrati, and C. Ziebert. The lithium-ion battery state-of-charge estimation using random forest regression. 8(4):2313–0105, 2022.

[8] Mathworks. Hyperparameter Optimization in Classification Learner App.

[9] Mathworks. Regression Learner App.

[10] Ministry for Primary Industries. A global comparison of the life-cycle greenhouse gas emissions of combustion engine and electric passenger cars - international council on clean transportation 2022, 2022. Online; accessed 19 June 2022.

[11] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. 14(3), 2022.

[12] X. Wang, J. Li, B.-C. Shia, Y.-W. Kao, C.-W. Ho, and M. Chen. A novel prediction process of the remaining useful life of electric vehicle battery using real-world data. *Processes*, 9(12), 2021.