# REMAINING USEFUL LIFE PREDICTION FOR LITHIUM-ION BATTERIES

A PREPRINT

**Qiyuan Zhao**
Department of Chemical Engineering
zhao922@purdue.edu

**Manush Bhatt**
Department of Computer Science
bhatt16@purdue.edu

**Shoban Kumar Rajamani Vimalarani**
Department of Computer Science
srajaman@purdue.edu

November 16, 2020

**ABSTRACT**

Lithium-ion batteries play a crucial role in our daily life. Our mobile phones, laptops, and the majority of the devices use lithium-ion batteries. An accurate estimation of remaining useful battery life (RUL) will aid the users to operate their smartphones and laptops more efficiently. In this work, we attempt to address the problem by evaluating various machine learning techniques to generate a user-dependent prediction and eventually propose a model that gives the best performance. The results of this prediction will enable the users to determine the health as well the amount of usable charge left in the battery.

***K*eywords** remaining useful battery life (RUL) · data-driven approach · model-based approach · machine learning

## 1 Introduction

In modern society, lithium-ion (Li-ion) batteries have been widely used in vehicles, household electronic equipment, smartphones and many other fields[1]. Meanwhile, the safety and reliability of the battery needs to be guaranteed in these applications[2]. Therefore, it is necessary to accurately predict the Remaining Useful Life (RUL) of Li-ion batteries and provide a better decision support for the to optimize maintenance strategies and reduce costs.

The typical methods to predict RUL of a Li-ion battery are usually divided into two areas, one is model-based approach and the other is data-driven approach[3]. Model-based methods use theoretical physics and chemical models to simulate the battery working procedures such as Particle filters[4, 5] and linear non-equilibrium thermodynamics[6]. Data driven approach is used when applying machine learning algorithms to an existing data set[7, 8]. For instance, Saha et. al raised a Bayesian framework combining the relevance vector machine (RVM) and particle filter to predict for RUL of Li-ion battery[9]. Liu et. al used an adaptive recurrent neural network for system dynamic state forecasting which can be used to predict the RUL of Li-ion battery[10]. Later on, Zhang, et al. [13] focused on a battery capacity prognostic approach using the empirical mode decomposition (EMD) denoising method and the multiple kernel relevance vector machine (MKRVM) approach, and proposed MKRVM approach for effective prediction. Considering the nonlinear and non-Gaussian capacity degradation characteristics of LIBs, a remaining useful life (RUL) based on Particle filter based approach is more suited traditionally[14]. There also exists another new approach which uses the Autoencoder-Deep-Neutral-Network (ADNN) integrated deep learning approach[1] and helps to predict the remaining useful life of the Li-ion batteries.

In this project, we will use data driven approach to predict RUL of Li-ion batteries. For data driven approach there are many Machine Learning algorithms proposed in previous papers[15, 18]. For our understanding, we will implement Support Vector Machine, Linear Regression, Relevance Vector Machine and Random Forest Regression in this project.

The next step is to apply them to the data set[11] and find possible outcomes. Root Mean Square Error (RMSE) will finally be used to assess prediction accuracy and compare performances of different prediction models.

We will also introduce some ideas from model-based approach and focus on studying the relationship among time series. We will compare these two different approach methods and find out a better way to predict the RUL of lithium-ion batteries.

## 2 Methodology

### 2.1 Data description

We used the lithium-ion batteries data set provided by the Prognostics CoE at NASA Ames[11]. This data set contains recorded charging and discharging processes at different temperatures and also records of the impedance as the damage criterion. The experimental steps are listed as follows:

- Charging process: Charging was carried out in a constant current (CC) mode at 1.5A until the battery voltage reached 4.2V and then continued in a constant voltage (CV) mode until the charge current dropped to 20mA.
- Discharge process: Discharge was also carried out at a constant current (CC) mode at 2A until the battery voltage fell to 2.7V, 2.5V, 2.2V and 2.5V for batteries #5, #6, #7 and #18 respectively.
- The experiments were stopped when the batteries reached end-of-life (EOL) criteria (battery capacity decays to 70% in this case).

The specific part of the dataset we used contains information such as

- $I_{measured}$: Battery output current (A) in charge and discharge process.
- $V_{measured}$: Battery terminal voltage (V) in charge and discharge process.
- $V_{charger}$: Voltage measured at charger (Volts) in charging process.
- $V_{load}$: Voltage measured at load (V) in discharge process.
- Temperature: Battery temperature (degree C) measured in charge and discharge process.
- Capacity: Battery capacity (Ahr) for discharge till 2.7V.

However, most of the features offered by this data set are time series. If we directly feed the whole data set to ML algorithms, it will be a time-costly effort. So, we performed some preprocessing before running the algorithms on the data set.

### 2.2 Feature Extraction

For Charging process, initially, we check the current change. The current in the charger is quite similar to the battery output current so we only focus on battery output current. It can be found that there are two characteristic times in each current curve, one is decay time and the other is half-life time which are shown in Figure 1. Decay time refers to the time current from which the current beings to decay. Since the current decay follows exponential function, we introduce half-life time to describe the decay speed. Then, we consider the voltage change and it can be noticed that although $V_{measured}$ and $V_{charger}$ curves are quite different, they have similar characteristic time: charger time which is shown in Figure 1. After reaching the peak voltage, $V_{measured}$ will maintain a constant value (4.2V). $V_{charger}$-decay follows exponential function and similarly we introduce half-life time to describe the decay speed.

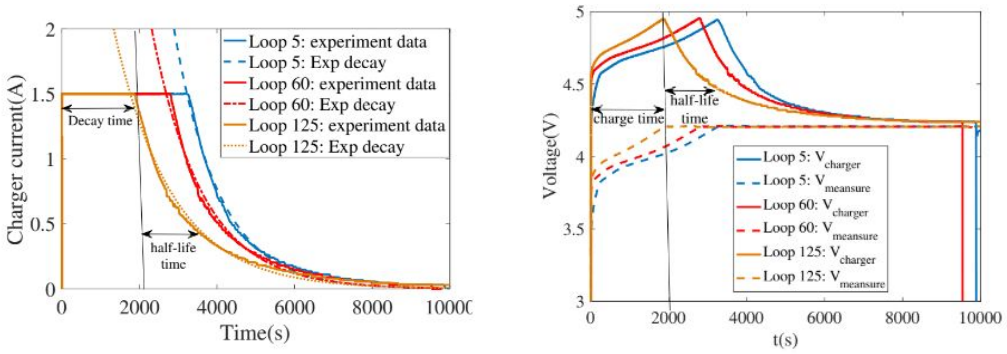


Figure 1: Charging Process with respect to Current(left) and Voltage(right)

As for discharge process, the absolute value of the current in the load is same as battery output current so we only focus on **battery output current**. Since the change of current is just like delta function, we can only obtain one characteristic value: decay time. As for the voltage change, the discharge time obtained from $V_{\textbf{measured}}$ and $V_{\textbf{load}}$ are same as the decay time. So we only use one discharge time to represent all the characteristic time. For $V_{\textbf{load}}$, we can additionally extract max voltage and for $V_{\textbf{measured}}$ we can extract min voltage and final voltage which are shown in Figure 2.

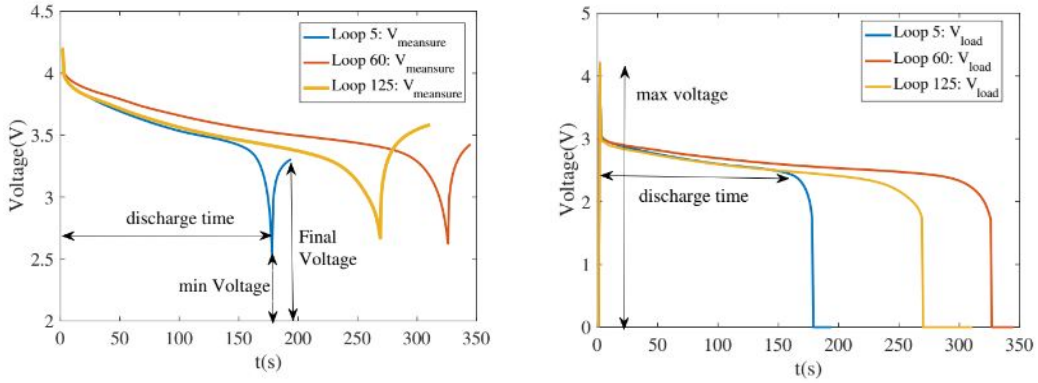


Figure 2: Discharging Process with respect to Voltage measured(left) and Voltage load(Right)

The specific features and the formulas are given as follows:

- $\text{CI}_{decaytime}$ =$\text{t}_{min\{i\}}$, s.t I($\text{t}_{min\{i\}}$) $< 1.5A$. Current decay time in charge process.
- $\text{CI}_{half-lifetime}$ =$\text{t}_{min\{i\}}$, s.t I($\text{t}_{min\{i\}}$) $< 1.5/eA$. Half-life time of current decay in charge process.
- $\text{CU}_{chargetime}$ =$\text{t}_i$, s.t U($t_i$) $= \max\{\text{U}\}$. Charge time in charge process.
- $\text{CU}_{half-lifetime}$ =$\text{t}_{min\{i\}}$, s.t U($\text{t}_{min\{i\}}$) $< \max\{\text{U}\}/e$. Half-life time of voltage decay in charge process.
- $\text{T}_{max}$,$\text{T}_{maxtime}$ $= \{\text{t}_i,\text{T}_i)|\max(\text{T}_i)\}$. The time when the battery temperature reaches the maximum value $T_{\max}$ in both charge and discharge process.
- $\text{DU}_{decaytime}$ =$\text{t}_{min\{i\}}$, s.t U($\text{t}_{min\{i\}}$) $= 0$. Current/Voltage decay time in discharge process.

## 3 Experiments

As mentioned above, we use the lithium-ion batteries data set provided by the Prognostics CoE at NASA[11]. For this experiment, we use data of four lithium ion battery #5, #6, #7 and #18. The data of #5, #6, #7 batteries are used to train and validate the model and #18 battery data is used for testing the results.

### 3.1 Preprocessing

Due to the impact of battery life, the sample collected in a battery charge and discharge process is not the same size. Some batteries have 800 points others have more than 5000 points. Therefore, we can not enter the charge and discharge data directly into the Machine Learning algorithms. We need to preprocess the original data and extract the typical characteristics of each cycle of the battery. Thus, all the features mentioned above are calculated for each cycle of the battery. Finally, we end up with 9 features for evaluation out of the 15 available ones and discard the insignificant ones.

### 3.2 Data Normalization

Because there are 168 charge and discharge cycles on the each battery, the RUL value corresponding to the $i^{th}$ charge and discharge cycle is $169 - i$, so that the supervised data of #5 battery can be obtained. In the same way, there is supervised data for #6 and #7 Battery. Different features have different values ranges, which will have a negative impact on the training of the model. In order to eliminate this effect, we transformed the range of the compressed features to $[0, 1]$ using the minimum-maximum normalization method.

We have used four models in this work: Linear Regression, Support Vector Machine, Random Forest Regression and Relevance Vector Machine.

### 3.3 Cross Validation and Scoring

For each of these four learning models, we employ 5-fold cross-validation to tune the hyper parameters. This method allows us to make the best use of our small training set. We score the models based on the root mean-squared error values.

### 3.4 Linear Regression

Linear regression is used for finding linear relationship between target and one or more features. Considering the relationship between the features in our dataset, we used the linear regression approach. It uses the least-square method to fit the data points. We observe that there is more amount of non-linear relationship among the features than the linear ones. The error-rate was also higher compared to all the other models.

### 3.5 Support Vector Machine

We decided to try Support Vector Machine as it uses a subset of training points in the decision function (called support vectors). It finds the optimal hyper-plane which separates data points using a hyper-plane with the largest amount of margin. Linear Support Vector Machine results were not conclusive enough, so we went with kernel approach. We used polynomial kernel with a degree value of 3. The polynomial kernel also takes combination of the features into consideration and our feature vector was not large enough for degree values greater than 3. Hence, we stopped with degree 3 to prevent over fitting.

### 3.6 Random Forest Regression

Since our dataset contains a relatively small number of training data, we found that our models tend to be prone to over fitting. We decided to try random forest regression because of its ability to reduce over fitting by aggregating results. Random forest regressor calculates the mean from the results of a number of decision trees. Decision tree regression works by splitting the data recursively and assigning the mean value of the leaf training samples to all points contained in that leaf. At each split, it selects the optimal feature and splits value to minimize the loss. For a random forest with N trees, the first step is to generate N different training sets of the same size as the original by sampling the original with replacement (a process known as bootstrapping). The individual decision trees are then trained separately on their respective data sets. To further decorrelate the trees, only a randomly selected subset of the features are made available at each split decision (square root of the number of features). Finally, after the trees are trained, results of all the trees are averaged and the final prediction is calculated. Because the predictions are averaged in this way, the variance of the model can be greatly reduced by simply increasing the number of decision trees. Applying random forest regression to our data, we swept over the number of trees and the maximum depth of trees. We found that the results generally improved with increasing number of trees and increasing tree depth. For our final testing we chose a model with 100 trees and a limit of 8 to the maximum tree depth.

### 3.7 Relevance vector machine

Since all the above methods showed obvious fluctuations in predicting the RUL, we tried Relevance Vector Machine which predicts the RUL based on past observations taken into consideration.

Many researches focus on ways to decrease these fluctuations such as ADNN approach[1] and multistage Support Vector Machine approach[7]. However, if we go back to the definition of RUL, we notice that RUL is directly proportional to the capacity change. The RUL of the first cycle that capacity is lower than 70% is defined as 0. So, if we can predict the change of capacity, we can get a better prediction of RUL. Pure data driven approaches focus more on features in a single step. However, if we consider the previous data as input X and future data as output y, we can build a dynamic prediction model which offers us a better result. In this way, we use Relevance vector machine to predict future observation series.

Relevance vector machine (RVM) has a similar form with support vector machine (SVM)[17]. But it has some advantages:

- It provides probabilistic estimates, as opposed to the SVM's point estimates.
- Typically provides a sparser solution than the SVM.
- Does not need a complexity parameter to be selected in order to avoid over-fitting.

To construct the input, at first we only focus on "capacity". Consider history observation data $y(t = 1 : k)$ where t is cycle number. Define d as embedding dimension which means the number of data points in the testing set. For example, if we can see how capacity changes in the first 20 cycles of a new battery, use can such that information to predict the RUL of that battery. We define $d = 20$. The input matrix can be constructed as:

$$X = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \\ \dots \\ \mathbf{x_{k-d}} \end{bmatrix} = \begin{bmatrix} y(t=1) & y(t=2) & \dots & y(t=d) \\ y(t=2) & y(t=3) & \dots & y(t=d+1) \\ \dots & \dots & \dots & \dots \\ y(t=k-d) & y(t=k-d+1) & \dots & y(t=k-1) \end{bmatrix} \tag{1}$$

construct output Y as

$$Y = \begin{bmatrix} y(t=d+1) \\ y(t=d+2) \\ \dots \\ y(t=k) \end{bmatrix} \tag{2}$$

We want to use previous d data to predict next data, in other words, build up such mapping relationship

$$y(l+d) = g(\mathbf{x_l}) = \sum_{j=l}^{l+d-1} w_j \phi_j(\mathbf{x_{l+d}}, \mathbf{x_j}) + b \tag{3}$$

in this way, we can sequentially predict $y(t = d + 1), y(t = d + 2)...$ until y (capacity in this case) reaches 70%.

## 4 Result and discussion

### 4.1 Capacity Estimation

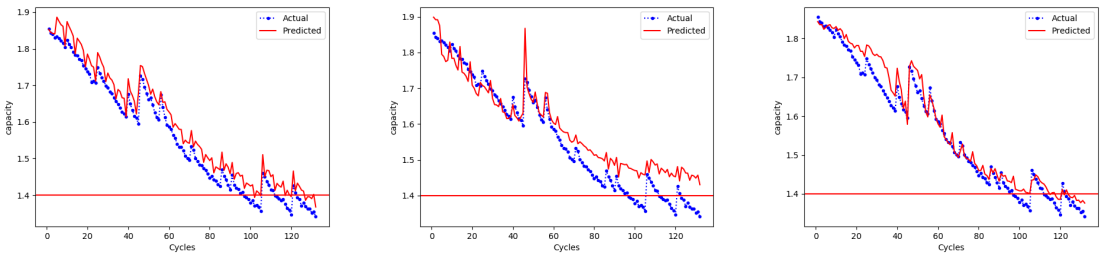


Figure 3: Capacity Estimation of Linear Regression(left),Support Vector Machine(center) and Random Forest(right) Algorithms

Figure 3 shows the result of predicting the Battery #18 for Linear Regression, Support Vector Machine and Random Forest Algorithms. We took a threshold value of 1.4V as mentioned in data set source[11]. Below the threshold value, the battery cannot be used and it needs to be replaced. The red line shows the true value and black dotted line shows the predicted value.

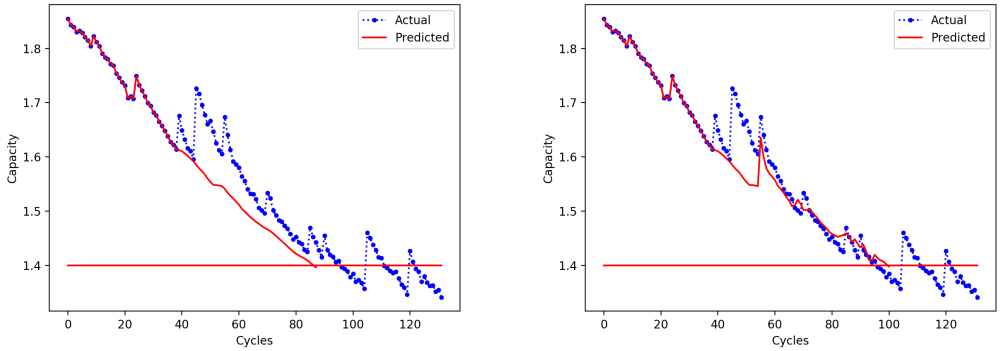


Figure 4: Prediction of capacity with d=40. (left) without randomness (right) add randomness shift in capacity

We then use RVM to investigate capacity time series. We first test for battery No. #18 with embedded dimension $d = 40$ which means our prediction begins from cycle 41. We notice than the trends of prediction and actual result are quite similar. However, there are some random shifts in the actual batteries which can not be predicted by our model. In Eq.(3), if we introduce some random "noise" into the prediction, we might be able to mimic the actual performance. The above figure on the right side shows that in some cases, adding randomness in the prediction can make the prediction more accurate. However, if we always add random shifts into the system, it will introduce more fluctuations into the prediction and may even lead to a worse prediction. We need to always consider when to introduce such random shifts and the shift range.

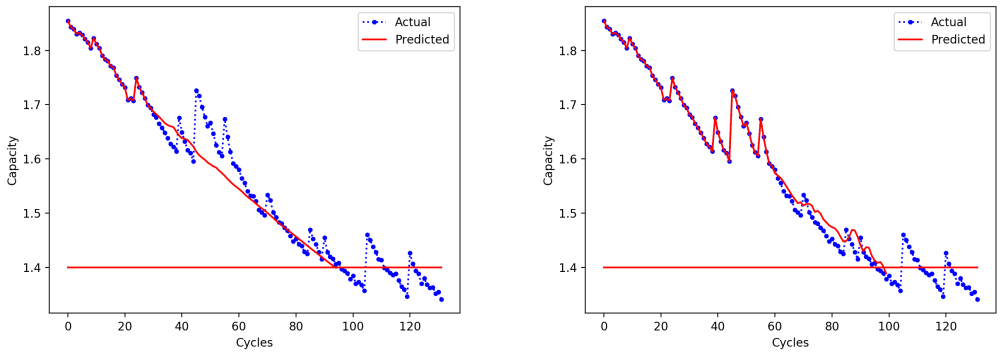


Figure 5: Prediction of capacity for battery No.18 start from 30 cycle (left) start from 60 cycle

As for the testing battery. Initially, we don't introduce randomness into the prediction. Figure. 5 shows the predictions of capacity beginning after 30 and 60 cycles. There is only one sudden increase of capacity in the first 30 cycles, the RVM prediction will capture such sudden change but will give us a smooth curve prediction for the remaining cycles. As for the first 60 cycles, four sudden changes in capacity can all be captured and since there is no more huge sudden change in the remaining cycles, the prediction will be accurate. Interestingly, sometimes the capacity prediction can't capture all of the real capacity changes but ends up giving a accurate RUL prediction. However, to build a more robust RUL predicting method, we need to also guarantee the accuracy of capacity prediction for the whole process. In this way, we consider introducing randomness. However, when and how to add which kind of randomness should be determined by training a even large database.

### 4.2 Remaining Useful Life Prediction

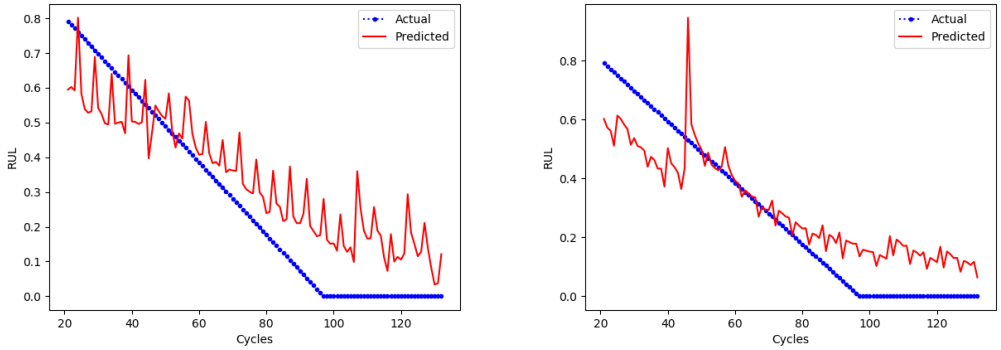


Figure 6: RUL Prediction of Linear Regression(left) VS Support Vector Machine(right) Algorithms

Figure 6 shows the prediction of Remaining Useful Life for Lithium-ion Batteries.The actual values are mentioned in blue dotted lines and predicted values are mentioned in Red lines.The predicted RUL is directly proportional to the number of cycles i.If the number of cycles increases, RUL decreases and at some point it reaches zero which means that the battery is unusable.RMSE is 14.21% for linear regression model and its accuracy is 85.79%.On similar lines,RMSE is 12.95% and accuracy is 87.05% for Support Vector Machine Model.As we can observe,RMSE value is quite high for these two models making the prediction not so optimal.

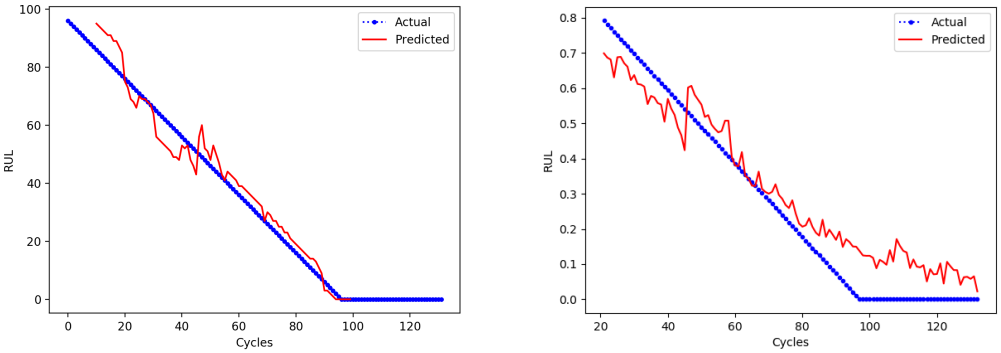


Figure 7: RUL Prediction of Relevance Vector Machine (left) and Random Forest Regression(right) Algorithms

Figure 7 shows the prediction of Remaining Useful Life for Lithium-ion Batteries. The actual values are mentioned as blue dotted lines and predicted values are mentioned as Red lines. Because of variance being less in RF model and the calculating the prediction from number of trees,the performance of Random Forest is better and RMSE is just 8.27% and Accuracy is 91.73%. For RVM method, we only consider single feature: "capacity" as mentioned in previous part. By using RVM, we can predict the capacity change and then use it to predict the RUL. For the prediction of RUL, the result is shown in Figure 7. We can see if we only use capacity from first 20 cycles we will over estimate the RUL since there will be a sudden change around cycle 20 and the initial data will just cover this peak. However, if we use capacity from first 30-40 cycles, we may under-estimate the RUL because it can't predict the sudden change between cycle 40-50. If we predict based on the more than first 60 cycles, the prediction will be stable and accurate. Also, compared with other methods, time series prediction is more stable and accurate.RMSE is 4.92% and Accuracy is 95.08%.

| Method | RMSE | Accuracy |
|---|---|---|
| Linear Regression | 14.21% | 85.79% |
| SVM | 12.95% | 87.05% |
| Random Forest | 8.27% | 91.73% |
| RVM | 4.92% | 95.08% |

Table 1: Comparison of Experimental Results

## 5 Conclusion and Future work

RUL prediction is of great importance to the state estimation and health management of lithium-ion battery. The developments of AI and deep learning areas provide new promising methods for lithium-ion battery RUL prediction. Overall, we found that Relevance Vector Machine gives better Prediction than other methods. In our future work,we plan to investigate time series of other parameters we extracted from the database. Although RUL is hugely dependant on capacity, other parameters can help do a better prediction. We can also use other kernels in RVM. For example some theoretical models[18], capacity change follows exponential decay rule, so maybe exponential kernel will be more suitable for this case. Lastly,we can try model driven approach using Gaussian Mixture Models and Particle Filter using Markov chain Monte Carlo (MCMC) approach.

## References

[1] Ren, Lei and Zhao, Li and Hong, Sheng and Zhao, Shiqiang and Wang, Hao and Zhang, Lin, Remaining Useful Life Prediction for Lithium-Ion Battery: A Deep Learning Approach, IEEE Access, (2018) 50587-50598.

[2] Hu, Xiaosong, Dongpu Cao, and Bo Egardt. Condition monitoring in advanced battery management systems: moving horizon estimation using a reduced electrochemical model. IEEE ASME Transactions on Mechatronics 23.1 (2018) 167-178.

[3] Schwabacher, Mark, and Kai Goebel. A survey of artificial intelligence for prognostics. Aaai fall symposium. 2007.

[4] Zhang, Qi, and Ralph E. White. Capacity fade analysis of a lithium ion cell. Journal of Power Sources 179.2 (2008): 793-798.

[5] Zhang, Qi, and Ralph E. White. Calendar life study of Li-ion pouch cells: Part 2: Simulation. Journal of Power Sources 179.2 (2008): 785-792.

[6] Virkar, Anil V. A model for degradation of electrochemical devices based on linear non-equilibrium thermodynamics and its application to lithium ion batteries. Journal of power sources 196.14 (2011): 5970-5984.

[7] Patil, Meru A. A novel multistage Support Vector Machine based approach for Li ion battery remaining useful life estimation. Applied energy 159 (2015): 285-297.

[8] Song, Yuchen, et al. Data-driven hybrid remaining useful life estimation approach for spacecraft lithium-ion battery. Microelectronics Reliability 75 (2017): 142-153.

[9] Saha, Bhaskar, et al. Prognostics methods for battery health monitoring using a Bayesian framework. IEEE Transactions on instrumentation and measurement 58.2 (2009): 291-296.

[10] Liu, Jie, et al. An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries. Annual Conference of the Prognostics and Health Management Society, Portland, Oregon, October 2010.

[11] B. Saha and K. Goebel (2007). Battery Data Set, NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA

[12] Tse, P. W., and D. P. Atherton. Prediction of machine deterioration using vibration based fault trends and recurrent neural networks. Journal of vibration and acoustics 121.3 (1999): 355-362.

[13] C. Zhang, Y. He, L. Yuan, and S. Xiang. Capacity prognostics of lithiumion batteries using EMD denoising and multiple kernel RVM IEEE Access, vol. 5, pp. 12061–12070, Jun. 2017.

[14] Lijun Zhang, Zhongqiang Mu, Changyan Sun. Remaining Useful Life Prediction for Lithium-Ion Batteries Based on Exponential Model and Particle Filter IEEE Special Section on Battery Energy Storage and Management Systems,2018.

[15] Cheng, Yujie, et al. Residual lifetime prediction for lithium-ion battery based on functional principal component analysis and Bayesian approach. Energy 90 (2015): 1983-1993.

[16] Liu, Datong, et al. Lithium-ion battery remaining useful life estimation based on fusion nonlinear degradation AR model and RPF algorithm. Neural Computing and Applications 25.3-4 (2014): 557-572.

[17] Tipping, Michael E. Sparse Bayesian Learning and the Relevance Vector Machine. Journal of Machine Learning Research. 1: 211–244. 2001

[18] Liu, Datong, et al. Lithium-ion battery remaining useful life estimation based on fusion nonlinear degradation AR model and RPF algorithm. Neural Computing and Applications 25.3-4 (2014): 557-572.