Duc Thanh Nguyen

REPORT

I. Dataset

Capacity of the battery was only recorded in Discharge process. As being said, a cycle is counted from the begin of a Discharge process to the begin of the next Discharge process. I have conducted a complete Data Frame as can be seen in Picture 1

```
dataset.tail(10)
```

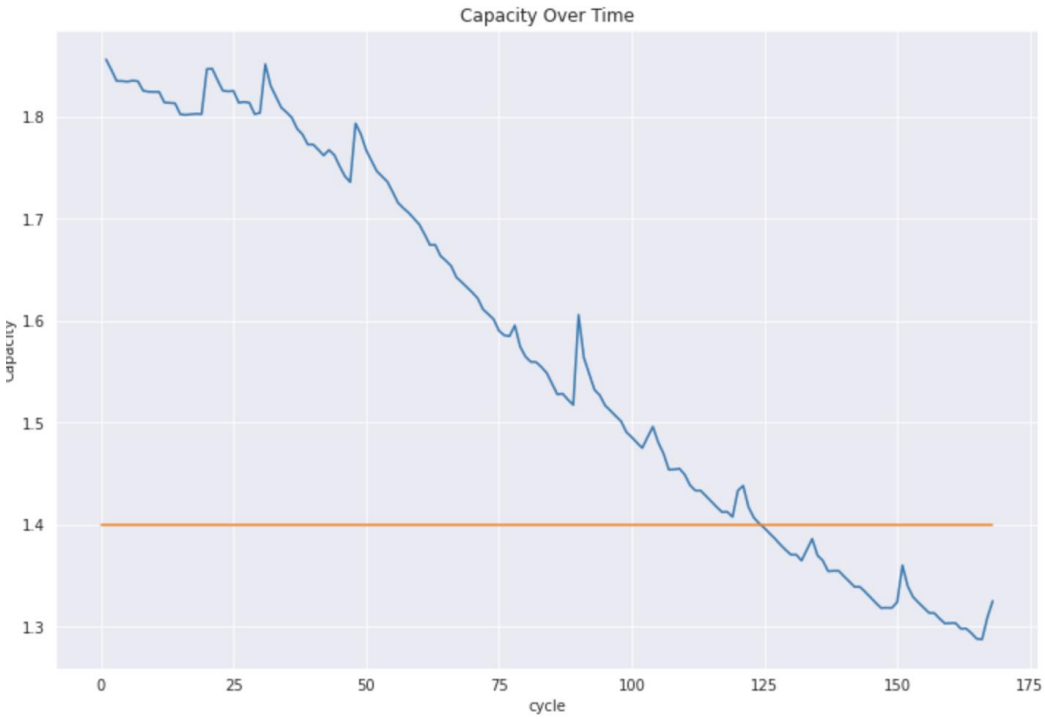
	cycle	ambient_temperature	datetime	capacity	voltage_measured	current_measured	temperature_measured	current_load	voltage_load	time
50275	168	24	2008-05-27 20:45:42	1.325079	3.563350	-0.000948	35.623242	0.0006	0.0	2732.359
50276	168	24	2008-05-27 20:45:42	1.325079	3.566589	0.000416	35.479866	0.0006	0.0	2742.093
50277	168	24	2008-05-27 20:45:42	1.325079	3.570132	-0.000338	35.345455	0.0006	0.0	2751.843
50278	168	24	2008-05-27 20:45:42	1.325079	3.573139	0.001471	35.171253	0.0006	0.0	2761.687
50279	168	24	2008-05-27 20:45:42	1.325079	3.576159	0.001138	34.966434	0.0006	0.0	2771.500
50280	168	24	2008-05-27 20:45:42	1.325079	3.579262	-0.001569	34.864823	0.0006	0.0	2781.312
50281	168	24	2008-05-27 20:45:42	1.325079	3.581964	-0.003067	34.814770	0.0006	0.0	2791.062
50282	168	24	2008-05-27 20:45:42	1.325079	3.584484	-0.003079	34.676258	0.0006	0.0	2800.828
50283	168	24	2008-05-27 20:45:42	1.325079	3.587336	0.001219	34.565580	0.0006	0.0	2810.640
50284	168	24	2008-05-27 20:45:42	1.325079	3.589937	-0.000583	34.405920	0.0006	0.0	2820.390

Picture1: Data Frame of the Discharge Process

In this dataset, I have dropped out all entities belong to the Charge and Impedance process, since there was no data of capacity of this process, and I cannot fill the missing cells with capacity of Discharge process because they are irrelevant and would disrupt my model. Therefore, after dropping out, the final data that I used has the shape of (50285,10), target value are Capacity and Variables include: voltage measured, current measured, temperature measured, current load, voltage load, time.

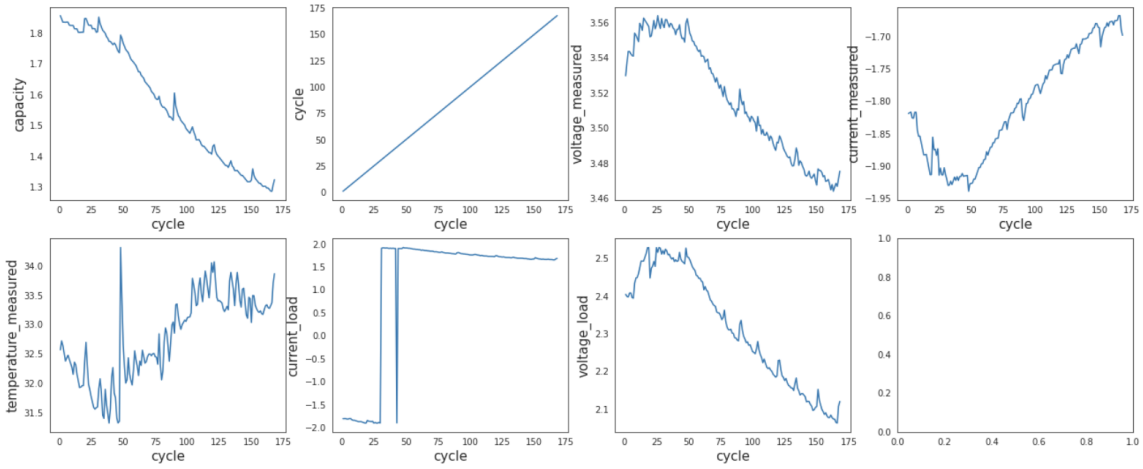
II. Data Exploration

Firstly, I plotted the capacity of each cycle to obtain the decrease of capacities overtime and reach the threshold line (1.4 Ahr)



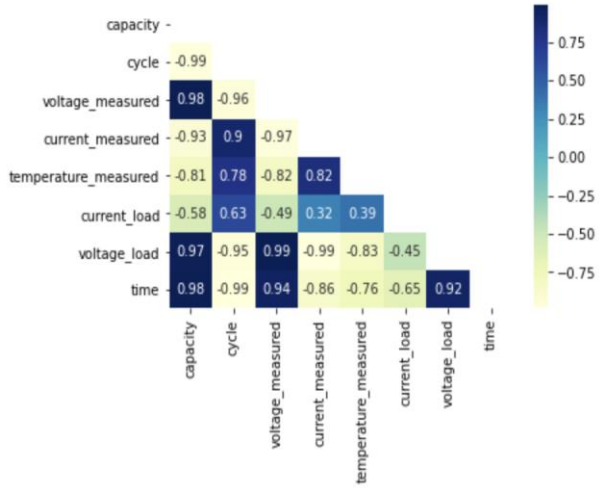
Picture 2: Capacity Over Time

Secondly, I plotted the trends of variables over time to obtain the changes of variables during the Discharge Process. Furthermore, I plotted the correlation matrix to confirm the linear relationship between variables and Capacity



Picture 3: Variables over time

Voltage measure and Voltage Load decreased overtime in Discharge proceeds, but the value increased in the first 50 cycles. In contract, Current measured and Temperature had reverse trends.



Picture 4: Correlation Matrix

As can be seen from the correlation matrix, variables had very high correlation score to each other and to Capacity. Current measured and Current Load had negative correlation scores.

III. Fitting data using Deep Learning LSTM model

Firstly, I split the data into trainset and test set

The train set is the whole dataset which has shape X_train (50285,6), y_train (50285,1)

As for the test set, I pick a time to predict the remain useful life which is cycle 50th, so the test data y_test has the shape of (119,1)

	cycle	capacity
0	1	1.856487
1	2	1.846327
2	3	1.835349
3	4	1.835263
4	5	1.834646
5	6	1.835662
6	7	1.835146
7	8	1.825757
8	9	1.824774
9	10	1.824613
10	11	1.824620
11	12	1.814202
12	13	1.813752
13	14	1.813440
14	15	1.802598
15	16	1.802107
16	17	1.802580
17	18	1.803068

```
attribs=[ 'voltage_measured', 'current_measured',
         'temperature_measured', 'current_load', 'voltage_load', 'time']
train_dataset = dataset[attribs]
sc = MinMaxScaler(feature_range=(0,1))
train_dataset = sc.fit_transform(train_dataset)
print(train_dataset.shape)
print(soh.shape)
```

```
(50285, 6)
(50285, 1)
```

- Fitting model

  For algorithm, I only use neural network with 3 layers and dropout to train the model

  LSTM model with 4 layers and 200 units each layer then dropout with chance of 30%

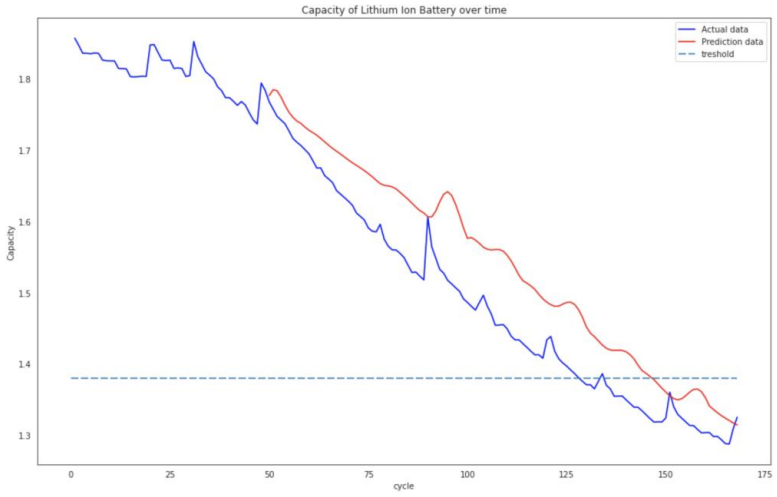
RUL Estimation Model receive R-Square~ 0.729 indicate that our prediction is 72.9% match the expectation and RMSE: 0.07 means that we might have about 0.07 of deviation from the true value in our prediction.

```
regress = Sequential()
regress.add(LSTM(units=200, return_sequences=True, input_shape=(X_train.shape[1],1)))
regress.add(Dropout(0.3))
regress.add(LSTM(units=200, return_sequences=True))
regress.add(Dropout(0.3))
regress.add(LSTM(units=200, return_sequences=True))
regress.add(Dropout(0.3))
regress.add(LSTM(units=200))
regress.add(Dropout(0.3))
regress.add(Dense(units=1))
regress.compile(optimizer='adam',loss='mean_squared_error')
regress.summary()
```

```
rmse = np.sqrt(mean_squared_error(tests, pred))
print('RMSE: %.3f' % rmse)
metrics.r2_score(tests,pred)
```

```
(119, 1)
RMSE: 0.070
0.7292918970279102
```

- Results
    - Actual time from beginning to fail: 128 cycles
    - Predict time from beginning to fail: 146 cycles
    - Error = 18 cycles



```
print("The actual mean time to failure: "+ str(Actual_cycle+ln))
print("The prediction usefulife: "+ str(Pred_cycle+ln))
error=Actual_cycle - Pred_cycle
print("The error of RUL= "+ str(error)+ " Cycle(s)")
```

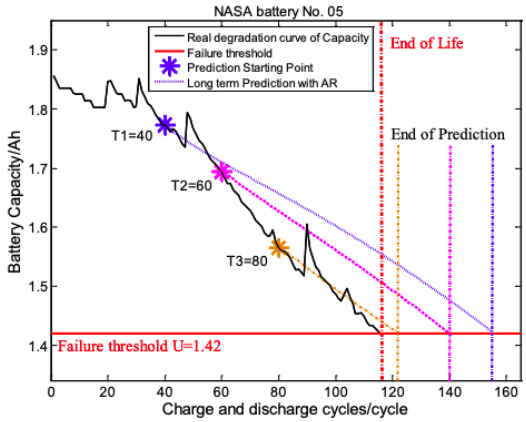
```
The actual mean time to failure: 128
The prediction usefulife: 146
The error of RUL= 18 Cycle(s)
```

Picture 5: Results

- Comparison

  As many publications that I read, accuracy of the model is evaluated by the error of the predict time to failure time and actual failure time. Compare to the predictive model using AR model of Liu at al, the result of my model seems to have a better accuracy because of lower error. The closer picked time is, the more accurate model is. The pick time of my model is 50, but having a lower error than Liu model at picked time 60 (18 cycles vs 24 cycles)

Starting point	End of Prediction (cycle)	RUL prediction result(cycle)	prediction error(cycle)
T1=40	154	114	38
T2=60	140	80	24
T3=80	122	42	6



Picture 6: Liu et al model

IV. Predict Remaining Useful Life using AR model

The second method that I used to predict the remain useful life is AR model, in which the best number of data used to predict the next data point (Best order p of the model). This method is introduced in the paper of Liu et al, in which the purpose of using this model is predict useful life of the battery in case we only have few data on hand. The AR model is used more extensive than MA model, this is a very simple model, as well as the computing load is small. The formular model is provided below, as can be seen from the first constrain that AR model is linear model and the number of data used to predict is the number of variables of the model

For time series $\{x_t\}$,

$$\begin{cases} x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + a_t \\ \phi_p \neq 0 \\ E(\varepsilon_t) = 0, Var(\varepsilon_t) = \sigma_\varepsilon^2, E(\varepsilon_t \varepsilon_s) = 0, s \neq t \\ Ex_s \varepsilon_t = 0, \forall s < t \end{cases}$$

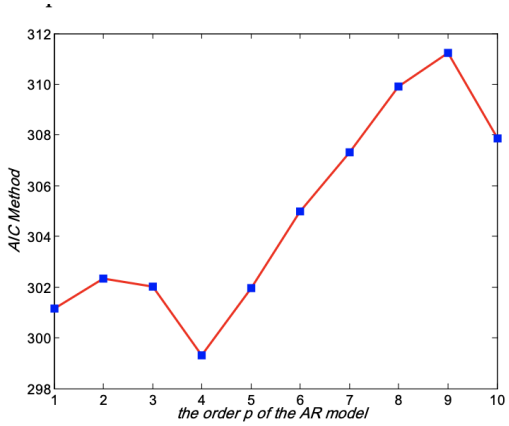
- Φ is the coefficient of the auto regression
- P is the order of the model
- a_t is the white noise (mean = 0, variance σ^2)

The first step of the method is finding the Best order p of the model using curve fitting, the best order model will obtain the lowest AIC (Akaike Information Criterion)

$$AIC(p) = N \ln \sigma_p^2 + 2p$$

- p is the model order
- σ^2_p is the prediction variance of the p order model

As Liu et al, based on the experience perspective, the order of the model should not be greater than 10 (p = [1:10]) To find the best p, they calculate predict value using Burg Algorithm with the order of model increase from 1- to 10 → best p is 4



Order *p*	AIC Value
1	201.6376
2	198.5693
3	200.7816
4	196.6361
5	203.1455
6	205.2109
7	205.4390
8	206.8059
9	207.0218
10	202.0553

To re-produce the result using this method, I used the library package ar_select_order, this library will formulate linear model with the number of orders from 1 to 20. The library also calculates the AIC score from each AR model with the formular: AIC = 2k – 2ln(L),
The pick time I chose is T1: cycle 40th, T2: cycle 60th, T3: cycle 80th, after finding the best order p of the model, I will predict the remain useful life from those pick time.

1. Using the amount of data before pick time
   Pick time T1: cycle 40th, best order p=1, AIC = - 8.709

```
                            AutoReg Model Results
==============================================================================
Dep. Variable:               capacity   No. Observations:                   40
Model:                   AutoReg(1)   Log Likelihood                117.485
Method:              Conditional MLE   S.D. of innovations              0.012
Date:               Wed, 03 Nov 2021   AIC                             -8.709
Time:                       14:48:45   BIC                             -8.581
Sample:                            1   HQIC                            -8.663
                                  40
```

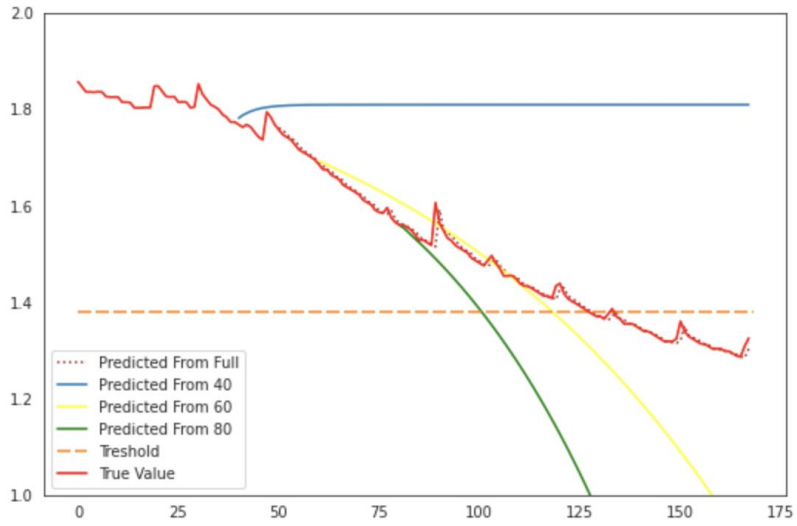
Pick time T2: cycle 60th, best order p=3, AIC = -8.488

```
                            AutoReg Model Results
==============================================================================
Dep. Variable:               capacity   No. Observations:                   60
Model:                   AutoReg(3)   Log Likelihood                166.042
Method:              Conditional MLE   S.D. of innovations             0.013
Date:               Wed, 03 Nov 2021   AIC                            -8.488
Time:                       14:48:55   BIC                            -8.309
Sample:                            3   HQIC                           -8.419
                                  60
```

Pick time T3: cycle 80th, best order p=1, AIC = - 8.779

```
                            AutoReg Model Results
==============================================================================
Dep. Variable:               capacity   No. Observations:                   80
Model:                   AutoReg(1)   Log Likelihood                237.687
Method:              Conditional MLE   S.D. of innovations             0.012
Date:               Wed, 03 Nov 2021   AIC                            -8.779
Time:                       14:48:43   BIC                            -8.689
Sample:                            1   HQIC                           -8.743
                                  80
```

After finding best order of model for each pick time, I fitted AR model using the dataset of 168 cycles and obtained the predicted plot, we have the chart below

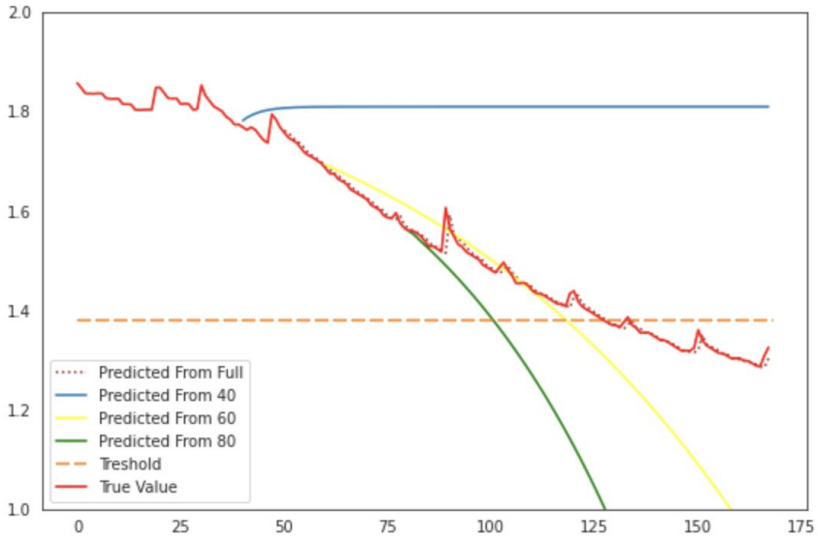


As can be seen, the AR model at the pick time cycle 40th could not capture the decreasing trend of actual data. The prediction line went horizontally, since 40 data point at the early state can not update the coefficient of variables to obtain the down trend intercept.

2. Using the whole dataset of 168 cycles

```
                            AutoReg Model Results
==============================================================================
=
Dep. Variable:               capacity   No. Observations:                   16
8
Model:                     AutoReg(4)   Log Likelihood                483.41
0
Method:              Conditional MLE   S.D. of innovations             0.01
3
Date:               Tue, 02 Nov 2021   AIC                            -8.66
0
Time:                       15:57:53   BIC                            -8.54
7
Sample:                            4   HQIC                           -8.61
4
                                 168
```

The result obtained from the model when fitting the whole dataset with the number of orders from 1-20, the best model with the lowest AIC is p=4. Using the 4 data points two predict the data from each pick time, we have the plot:



Start Point	End of prediction life	RUL	Error
T1=40	N/A	N/A	N/A
T2=60	120	60	8
T3=80	105	23	23

V. Predict Remaining Useful Life using Machine Learning models

1. Dataset

   To recall, the dataset I used to fit machine learning model has the dimension (50285,10), in which there are 5 variables include: Voltage Measure, Current Measure, Temperature, Voltage Load, Current Load. The dataset includes 168 cycles, threshold

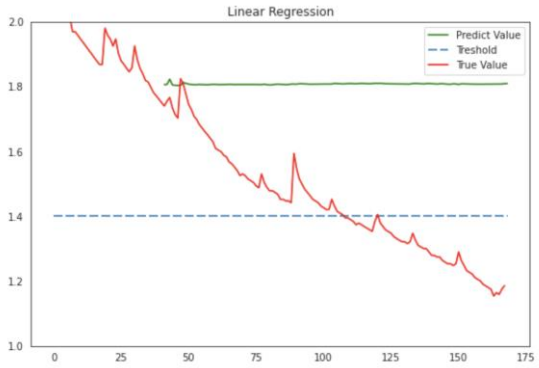
that a battery is failed is 1.4 Ahr and the actual failure was in cycle 128th. At first, I pick the pick time at cycle 40th, in which I used the data of the fist 40 cycles as trainset and the remaining time is test set. The dimension of train set, and test set are:

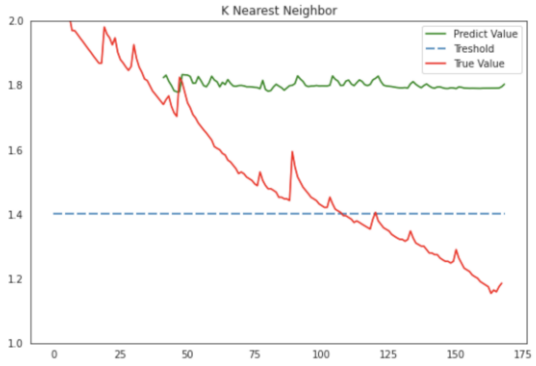
- Train set shape: X= (9254,5), Y= (9254,)
- Test set shape: X= (41031,5), Y= (41031,)

2. Fitting Machine Leaning models

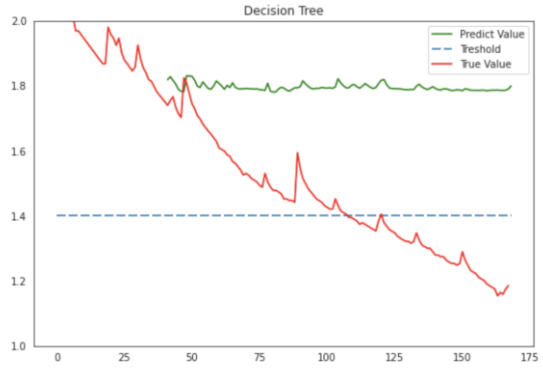
   To perform Machine Learning, I fitted the train set to get the model, the predict the capacity of test set and plot the result as the green line as shown below

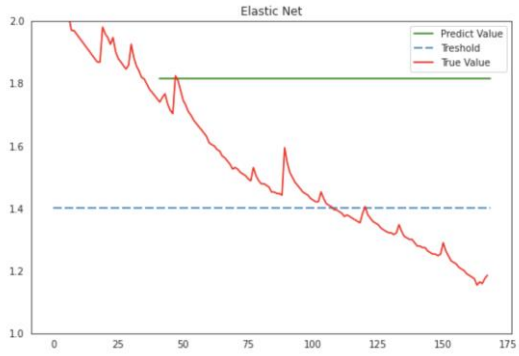
   a. Linear Regression and KNN



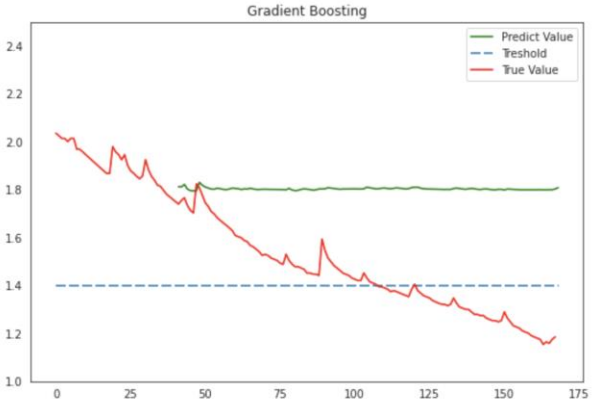


   b. Decision Tree and Elastic Net





   c. Gradient Boosting and results



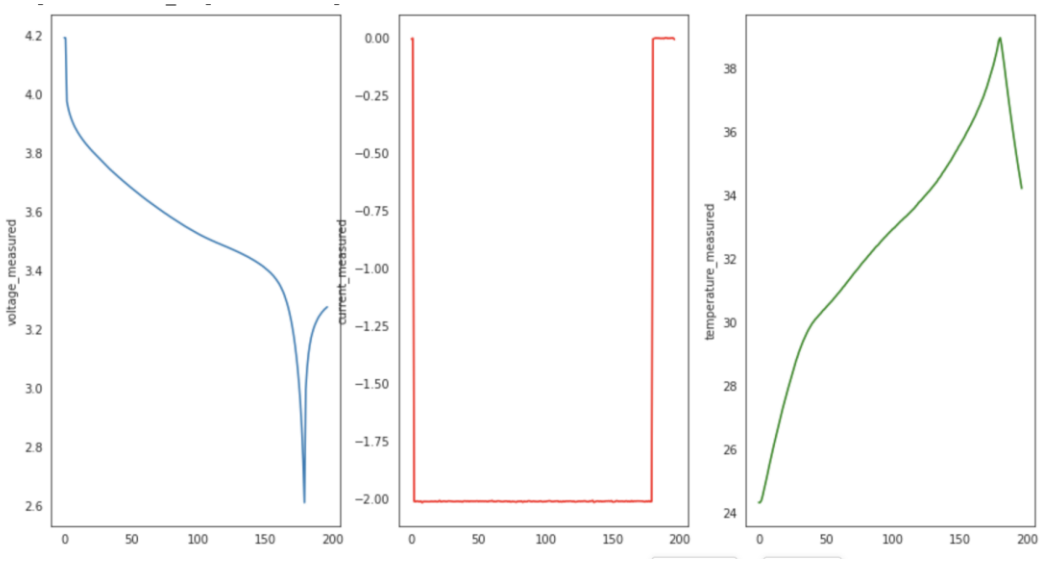
	RMSE
Linear Regression	0.345874
K Nearest Neighbor	0.337173
Decision Tree	0.332426
Elastic Net	0.353231
Gradient Boosting	0.341182

I have performed 5 machine learning models and all five models can not capture the decresing trend of capacity over time. I have calculated the Root Mean Square Error (RMSE) to compare the perfomace among models, the results were really high for all models, the accaptable RMSE for a good model should be around 0.07 as my experience.

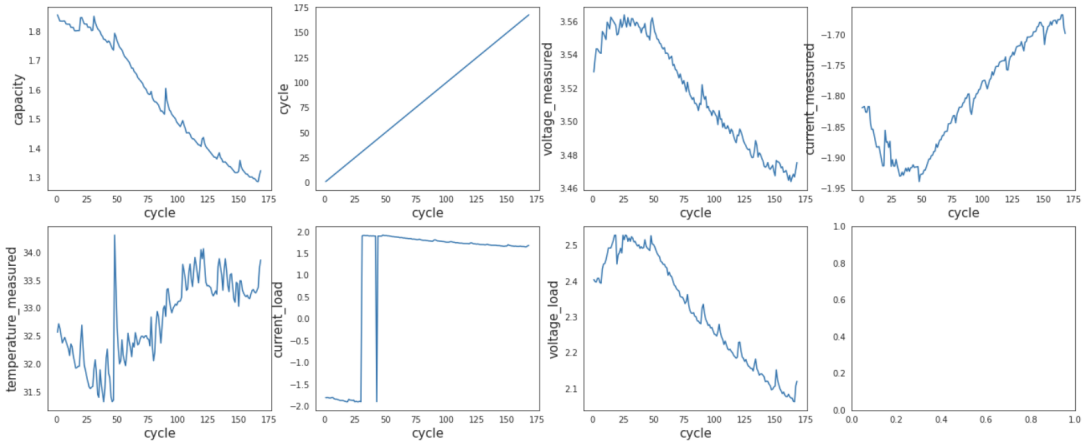
d. Investigating the issue

As my experience, the reason for prediction model has a completely different trend compared to the actual data is that variables do not have high correlation to the target value "capacity". I have performed the data exploration for the 168 cycles, however, there were hundreds of data within a cycle that might indicate different trends.

By that reason, I investigated the data of cycle 1 only by filtering out data of cycle 1 only, cycle 1 includes 197 data point. Then I plotted the 3 variables: Voltage measure, Current Measure, Temperature



Plot of 1 cycle



Plot 168 cycles

As can be seen from the 2 plots, the plot of the dataset of 168 cycles indicates very clear Positive/Negative relationship between variables and capacity, but if

looking in to cycle 1, the variables went different trends which interrupts the coefficient of machine learning models, making prediction model can not indicate the decreasing trends.

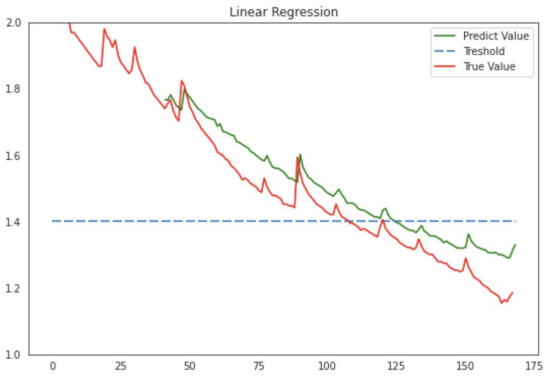
VI. Predict Remaining Useful Life using Machine Learning models with only 168 cycles

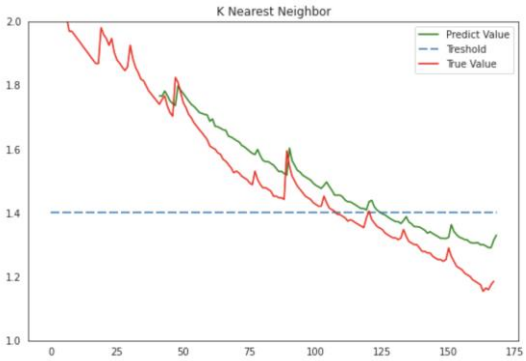
1. Dataset

   In this step, I used the dataset of only 168 data point which are the mean value of each cycle. The total variables are 5, train set includes 40 cycles and test set has 128 cycles.

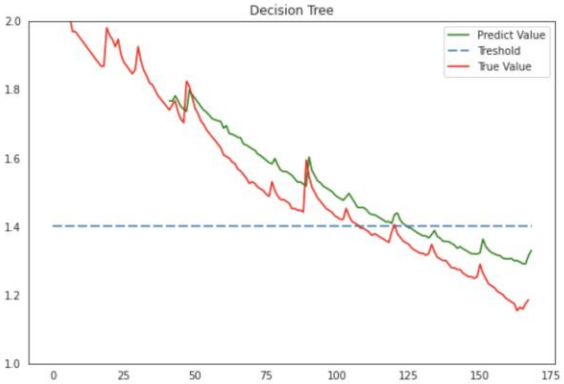
2. Fitting machine learning models

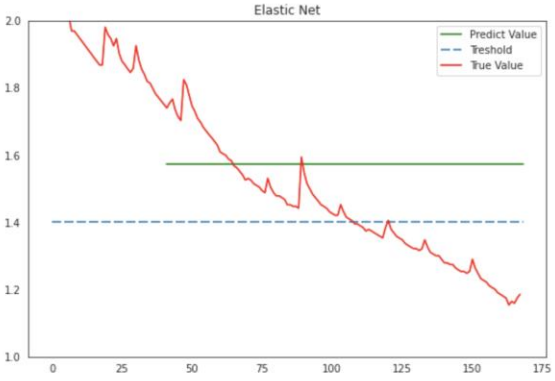
   a. Linear Regression and KNN



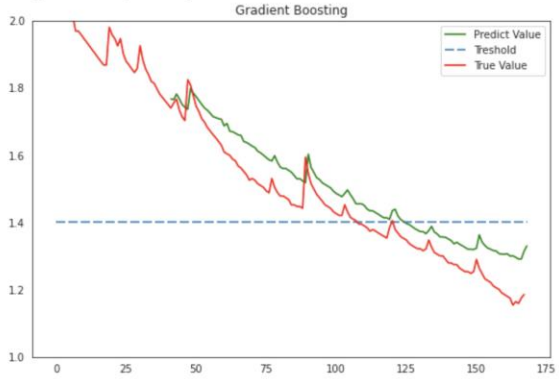


   b. Decision Tree and Elastic Net





   c. Gradient Boosting and results



	RMSE	Fail Cycle	Error	R_Square
Linear Regression	0.017246	118.0	-10.0	0.986748
SVR	0.071428	145.0	17.0	0.772684
K Nearest Neighbor	0.011955	125.0	-3.0	0.993632
Decision Tree	0.000131	125.0	-3.0	0.999999
Elastic Net	0.168327	0.0	-128.0	-0.262392
Gradient Boosting	0.002588	125.0	-3.0	0.999702

As can be seen from the result table, all models have performed very well, captured the decreasing trends of the actual value of capacity as expected, excepts Elastic Net. The Root Mean Square Error (RMSE) of all models are acceptable, showing that the prediction plot is close to the actual data. The S-square score of all models excepts Elastic Net, are very high, showing that all models are capable to predict future failure accurate. As the results, Decision Tree and Gradient Boosting are the best models, which have a highest R-square score and lowest RMSE. Except logistic regression, it is noticeable that the other models are classification models. However, Python has 2 types of libraries for classification, classifier for categorical data and regressor for continuous data. In regressor library, continuous data is discretized into different classes, and prediction value is formulated based on the class that the actual data belonged.

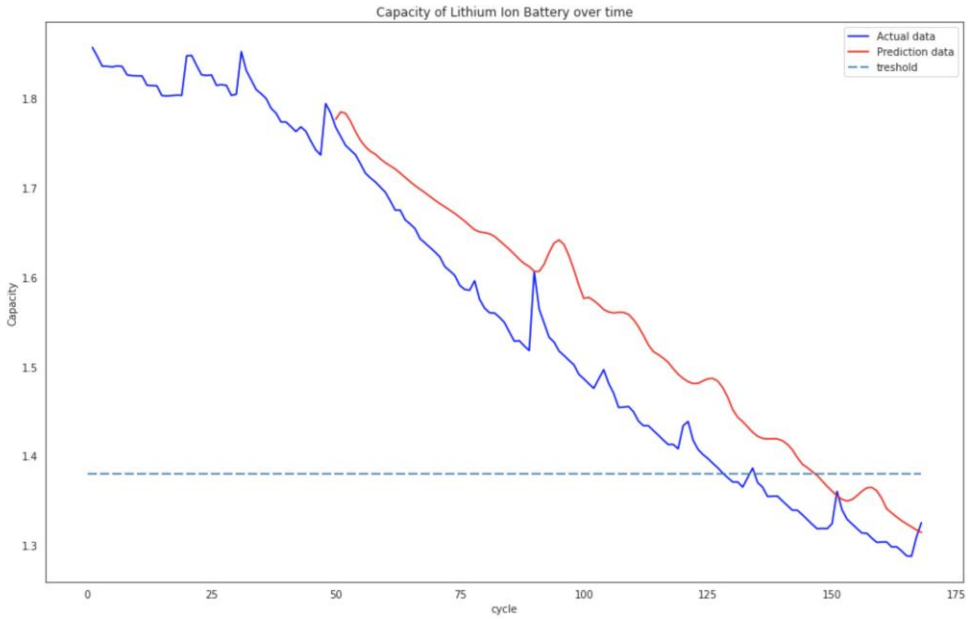
VII. Conclusion

Using the Lithium-Ion battery, I have performed 3 main methods in order to predict the remain useful life of the battery: Deep Learning model, AR model and Machine Learning model. The error of prediction failure time compared to actual failure time using 3 methods if we predict from the cycle 40^{th}

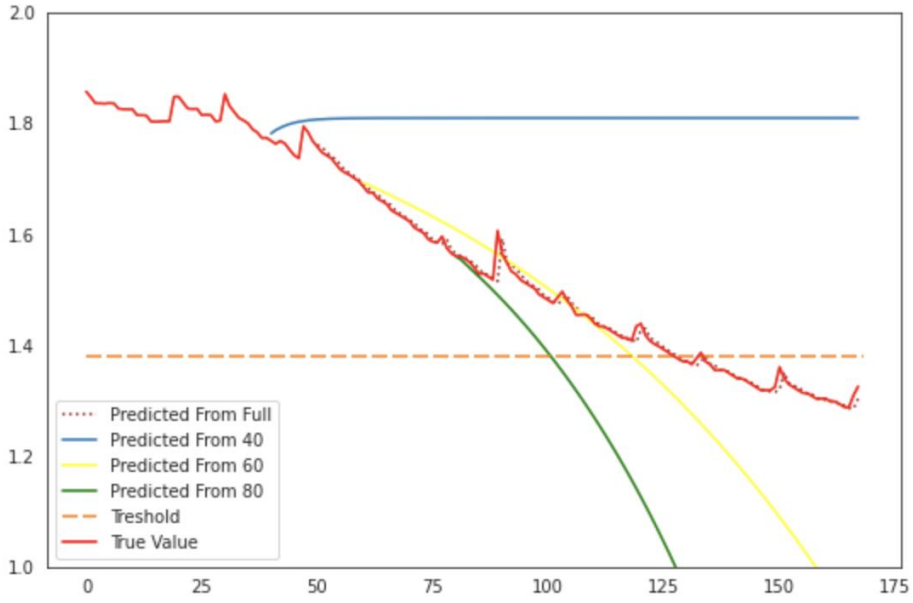
Deep Learning LSTM: 18 cycles

Linear Regression: 10 cycles

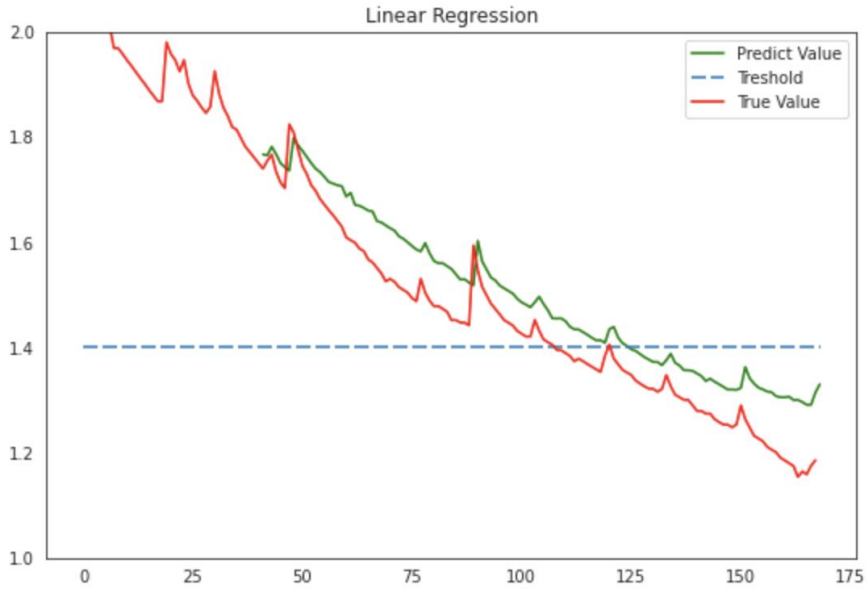
AR model with best order of 4: N/A (8 cycles of predict from cycle 60^{th}



Deep Learning LSTM



AR model



Linear Regression

1. LSTM (Deep Learning model): By obtaining the performance of this model, we can see that LSTM can process the full data of 50285 data point but still capture the decreasing trend of capacity, and predicted value is accurate. However computing load is very extensive and time consuming since the whole data went through different hidden layers. Most of deep learning model required complex computation to recognize the data pattern

2. AR model is simple, only required target value (capacity) and computing load is small. However, the model cannot predict the future if there are few data on hand, since few data cannot update the coefficients of AR model to create linear trends
3. Machine Learning models: Only works on data that has clear pattern on trend in time series. Prediction of the model is accurate. Computing load is larger than AR model but much smaller than LSTM.

Both Deep Learning and Machine Learning models allows us to continuously monitoring the changes of target values and variables, determine the status of the battery any specific time, meanwhile in AR model, we can only predict the target value capacity but unable to determine value of variables as temperature, voltage, etc.