Using Machine Learning to Predict the Remaining Useful Life of NASA Li-Ion Batteries

Authored by Arch Wilson
CSE 6748 - Practicum Project

Background

The purpose of this project was to examine data from NASA batteries to predict their remaining useful life (RUL). Data[1] obtained online from Kaggle presented a set of Li-ion batteries run through charge and discharge operations under varying conditions until a specified end-of-life (EOL) criteria was met. Repeated charge and discharge operations resulted in an accelerated aging of the batteries and a change in the batteries' internal parameters. These internal parameters were used to create models to predict RUL.

Battery health is most typically assessed by taking impedance measurements. Impedance is a measure of the ability of a circuit to resist the flow of electrical current [2]. In short, healthier batteries have a lower resistance to the flow of electrical current. This measurement was performed separately from the charge/discharge operations via an electrochemical impedance spectroscopy (EIS) frequency sweep. The resulting impedance was measured in electrolytic resistance (Re) and charge transfer resistance (Rct). Impedance measurements are not necessarily similar between all battery cells of the same type, but gross changes in impedance can indicate a degradation in battery performance.

Given that impedance measurements must take place separately from charge and discharge operations, it would be useful in practice to use other battery parameters to predict RUL. The NASA battery data measured voltage, current, and temperature over time for each charge/discharge operation. These are parameters typically measured in commercial battery applications and the ability to predict RUL based on these alone is valuable to potential users. Reference 3 lists an example of a commercial Battery Management System (BMS) used in solar applications.

Data Exploration and Cleaning

Several different experiments were performed within the dataset examined. Batteries 5, 6, 7, and 18 were run through the same experiment and were considered for this project.

Data for each charge/discharge operation was stored in separate .csv files. During cleaning, charge and discharge operations were paired together into cycles and a dataframe was made to store all data related to each cycle. The cycles were labeled, and the RUL was imputed by reversing the cycle count.

In the data, impedance measurements were taken at irregular intervals and missing for the first 20 cycles of batteries 5, 6, and 7. These first 20 rows were dropped and not considered for modeling. Impedance data was then forward-filled and back-filled for the remaining data.

Each .csv file included data taken before and after the charge and discharge operations. It was noticed that some charge operations were ‘incomplete’ and charging was not performed from a fully discharged state. This was resolved in the data cleaning process and is shown in Figures 1 and 2.

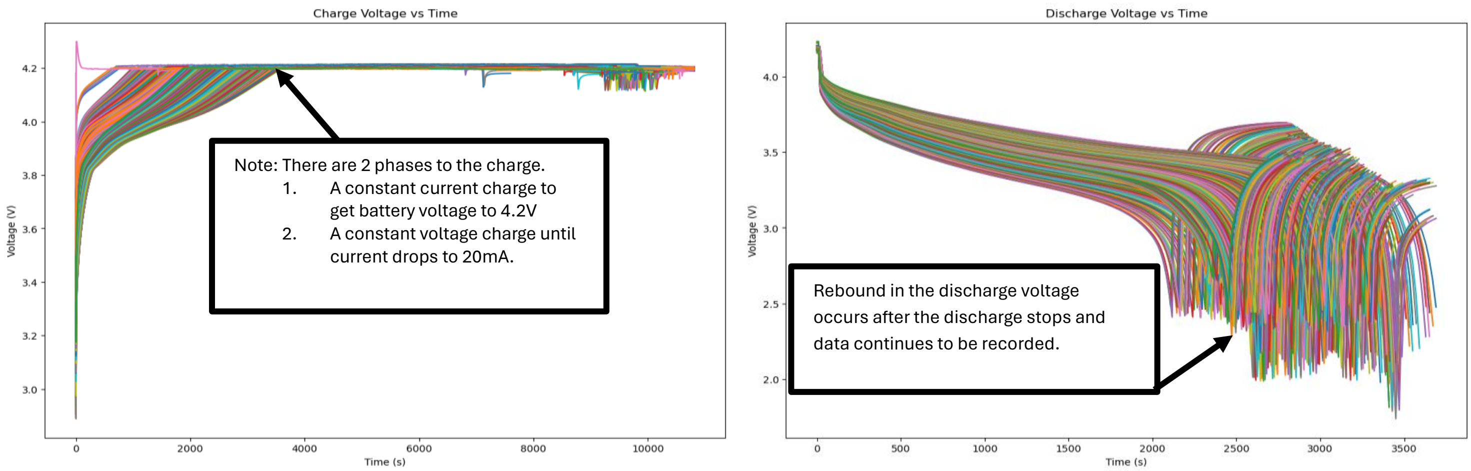

Figure 1: Raw data before cleaning

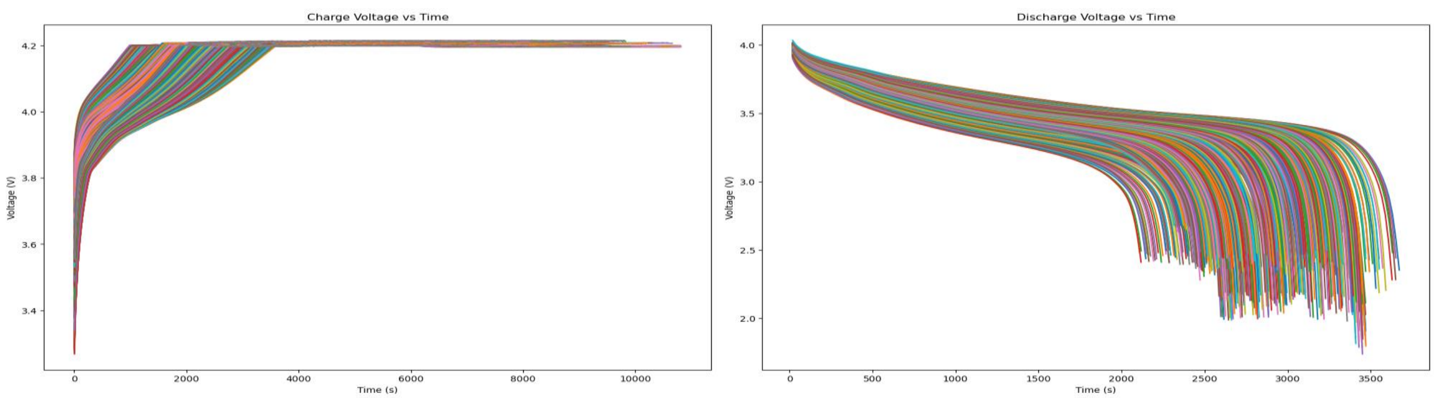

Figure 2: Data after cleaning

Modeling Features

There was a limited number of features given in the data specific to each complete charge/discharge cycle. Other features were imputed from the .csv files for each individual charge/discharge operation.

Feature	Reason
Capacity	Battery capacity in Ahr measured during discharge.
Re	Electrolyte Resistance (Impedance)
Rct	Charge Transfer Resistance (Impedance)

Table 1: Summary of given features for a complete cycle

To effectively predict changes in RUL, there must be changes in the features observed over time. Figure 3 shows the changes in the shape of the charge/discharge profiles as the batteries progress through their RUL.

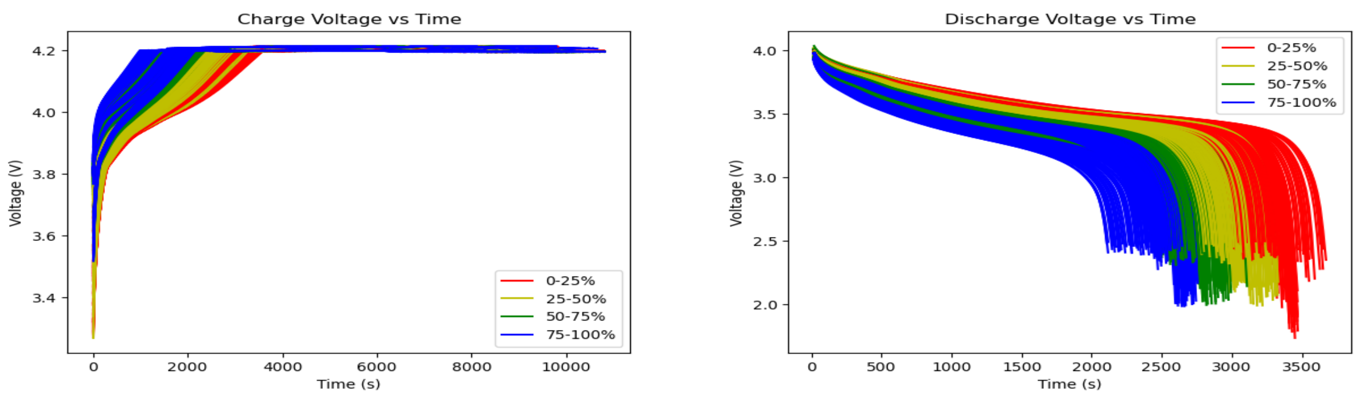

Figure 3: Charge/discharge profiles through percent of RUL

The S-shaped curves seen above are best approximated by a sigmoid function. Due to the constant current portion of the charge providing no differentiation in shape, only the constant voltage portion was considered. The formula below was used to transform the time for each profile and a curve was fitted to the data for each cycle. Figure 4 shows the profiles as approximated by the fitted sigmoid curves.

$$Voltage = \frac{L}{(1 + \exp(-k(x - x_0))} + b$$

Where:

L – scales sigmoid response
b – shifts sigmoid response
k – scales sigmoid input
x_0 – sigmoid inflection point
x – cycle time

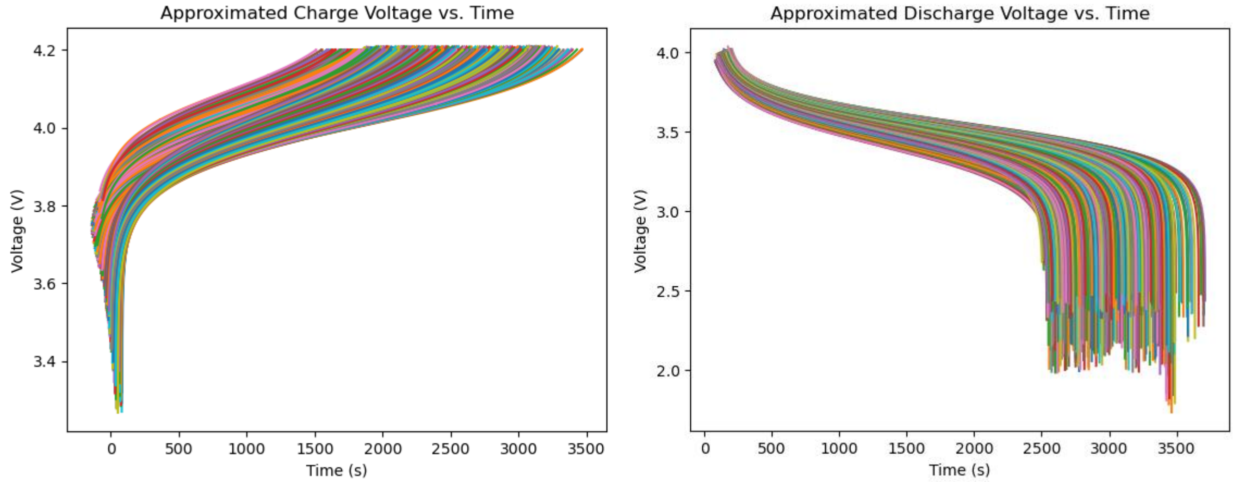

Figure 4: Sigmoid approximations of charge/discharge profiles

Additional features were imputed from the .csv files to try to capture underlying patterns in the data. Table 2 summarizes the list of imputed features.

Feature	Explanation
Charge_time_0	Time to perform constant current charge
Charge_time_1	Time to perform constant voltage charge
Charge_temp_gain	Temperature gained during charge
Discharge_max_temp	Maximum temperature achieved during discharge
Discharge_avg_temp	Average temperature during discharge
Discharge_currend_std	Standard deviation of discharge current
L_charge, b_charge, k_charge, x_0_charge	Sigmoid coefficients of charge curve
L_discharge, b_discharge, k_discharge, x_0_discharge	Sigmoid coefficients of discharge curve

Table 2: Summary of imputed features

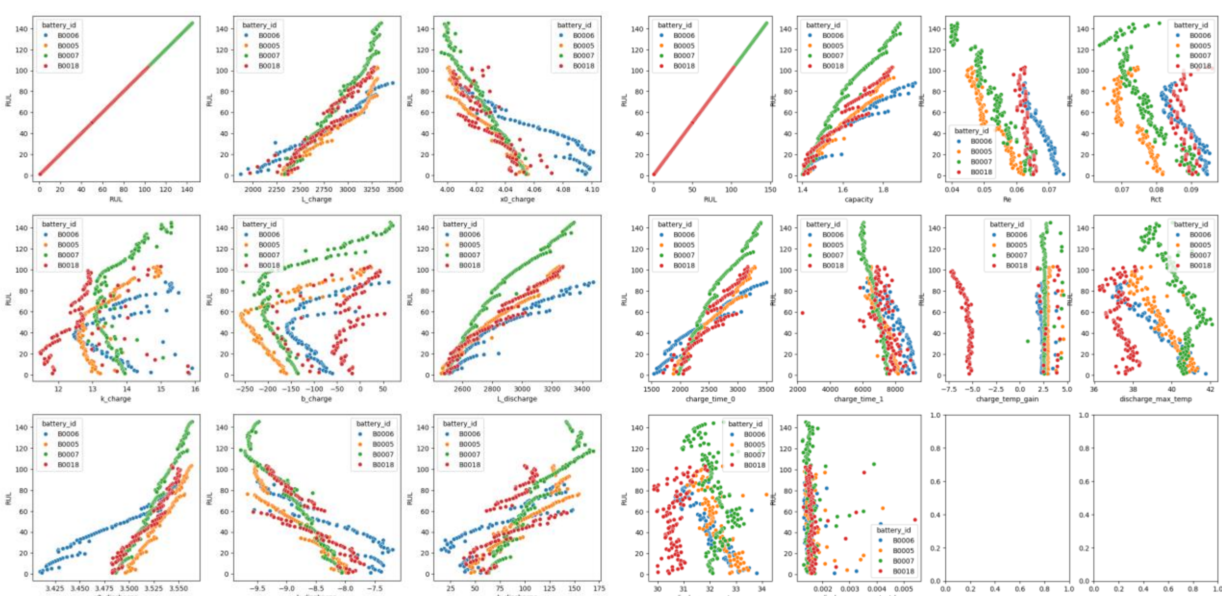

Figure 5: Composite of correlation plots of each feature to RUL colored by Battery ID

Baseline Development

Two sets of baseline models were considered for this project. The first was a set of models using only impedance features. These features are special because they are ones typically used to measure battery health, however, they can only be measured outside of normal operations (charging or discharging) and require equipment not typical in a BMS. The second set of models considered capacity as the only feature. Battery capacity is what the EOL criteria was based on and is a direct indicator of how much energy the battery can store.

Models Considered

Linear Regression (LinReg)
The first model considered was a simple linear regression model. This was used based on the observed linear relationship between the baseline features and RUL. Additionally, linear regression models are computationally cheap to build, fit and predict with.

Decision Tree (DT)
Two types of decision trees models were considered. The first was a small tree with simple parameters, and the second used a grid search to find optimal parameters. The tuned decision tree used the following parameters and performed a grid search to select the best model.

Max_depth: 3, 5, 7, 10, 15, 20, None
Min_samples_split: 2, 5, 10, 15, 20
Min_samples_leaf: 3, 5, 10, 15, 20
Max_features: sqrt, log2, None

Random Forest (RF)
As a follow on to the decision tree models, random forest regression was also performed. As an ensemble learner, a random forest was expected to have better performance with the drawback of not having visibility into how the data is being split. The tuned random forest model was given the following parameters for a grid search.

N_estimators: 50, 100
Max_depth: 3, 5, 7, 10, 12
Max_leaf_nodes: 3, 6, 9, None

Neural Network (NN)
Two neural network models were built. These included a simple 2 layer network with 64 nodes in each as well as a network that was tuned with respect to the number of layers and nodes within each layer.

Modeling Process

A ModelLibrary class was written which included methods for implementing each model above. The class was initialized with a file path to the cleaned dataset. During initialization, the data was split into training/test sets to ensure the same split was used for each model. Default hyperparameters for some of the models were also set upon initialization. Class methods were developed to run each model. Inputs to the methods for each model allowed the following:

1. Specific feature selection
2. Hyperparameter specification
3. Tuning the model with default or specified parameter grid
4. Visualize predicted vs. actual results and plot loss for neural network models
5. Write scores to a dataframe
6. Return the model

Evaluation Criteria

Baseline and feature models were evaluated using an R^2 score for each model as well as the mean absolute error (MAE). The R^2 score provided a metric of how well the variance in the response is explained by the model. MAE provided an overall explanation of the error between predicted and expected.

Baseline Results

The performance of the baseline models is summarized in Table 3 below. The optimal performing models for each type are shown below.

Model	Features	Parameters	R^2 Train	MAE Train	R^2 Test	MAE Test
LinReg	Re, Rct		0.512	19.535	0.618	19.721
DT	Re, Rct	Min_samples_split: 15	0.775	10.394	0.871	10.233
RF	**Re, Rct**	**Max_Depth: 10** **Min_Samples_Leaf: 3** **Min_Samples_Split: 15**	**0.839**	**8.448**	**0.906**	**8.073**
NN	Re, Rct	Layers: 3 Nodes in layer: 190, 190, 180	0.760	12.912	0.789	14.209
LinReg	Capacity		0.807	11.196	0.797	13.169
DT	Capacity	Max_Depth: 7 Max_Features: ‘sqrt’ Min_Samples_Leaf: 15	0.810	11.311	0.798	13.195
RF	Capacity	N_estimators: 50 Max_depth: 5	0.785	11.596	0.818	12.407
NN	Capacity	Layers: 3 Nodes in layer: 160, 190, 10	0.829	10.547	0.797	13.233

Table 3: Performance of baseline models

The best two best performing models were the decision tree and random forest models using impedance features Re and Rct. Although impedance features have less of a linear relationship compared to capacity, the tree style learners were able to capture the individual relationships of Re and Rct compared to RUL into a better learner. The benchmark model in bold above was a random forest model with impedance features as inputs.

Improved Models

The goal of this project was to use features outside of impedance to develop an improved model for predicting RUL. Feature Set 1, which included features readily available in a BMS, looked at values directly related to current, voltage, and temperature. Feature Set 2 comprised solely of the coefficients describing the shape of the charge and discharge profiles. This set was considered separately because they are not readily available in a BMS and would require additional computational overhead to compute as inputs to a model.

Feature Set 1	Feature Set 2	
Capacity	L_charge	L_discharge
Charge_time_0	b_charge	b_discharge
Charge_time_1	k_charge	k_discharge
Discharge_max_temp	x_0_charge	x_0_discharge
Discharge_avg_temp		

Table 4: Feature Sets

Each feature set was run independently through the same models as the baseline features. The results are summarized in Table 5 below.

Model	Features	Parameters	R^2 Train	MAE Train	R^2 Test	MAE Test
LinReg	Feature Set 1		0.918	7.387	0.939	7.075
DT	Feature Set 1	Max_Depth: 10 Min_Samples_Leaf: 3 Min_Samples_Split: 2	0.929	5.978	0.949	5.725
RF	**Feature Set 1**	**Max_Depth: 10** **N_estimators: 50**	**0.957**	**4.639**	**0.966**	**4.475**
NN	Feature Set 1	Layers: 3 Nodes in layer: 190, 70, 90	0.977	3.498	0.964	4.546
LinReg	Feature Set 2		0.858	9.923	0.859	11.175
DT	Feature Set 2	Max_Depth: Min_Samples_Leaf: 15	0.926	6.072	0.921	6.090
RF	Feature Set 2	N_estimators: 100 Max_depth: 12	0.944	4.937	0.959	4.796
NN	Feature Set 2	Layers: 3 Nodes in layer: 180, 170, 70	0.973	3.156	0.964	4.716

Table 5: Improved Model Results

The training R^2 and MAE scores were noted as being lower than the test scores for most models except the neural network. These other models underwent hyperparameter tuning and development using an average of 7-fold cross validation for the training R^2 and MAE values. For final evaluation models were fitted on the entirety of the training data, so it is not surprising that the test scores are superior to the training scores. Cross validation was not used in evaluating the neural network models since the data is automatically shuffled between training and validation sets after each epoch.

The highest performing model was the random forest model that used Feature Set 1. Both neural networks had similarly low R^2 and MAE scores compared to the random forest model. All models, except the linear regression, performed for Feature Set 2 had better performance compared to the baseline.

Feature Set 1 proved better than Feature Set 2 when comparing both R^2 and MAE values. The reason behind this can be explained by the type of data contained in each set of features. Features Set 1 has data related to both the charge profile and temperatures during discharge. Feature Set 2 has data related to both the charge profile and discharge profiles. Given this, the data for temperatures during discharge operations appears to have more value in predicting RUL compared to the shape of the discharge profile.

Model Scoring

Outside of the performance of the model, the time to train and query models was also considered. In a practical application, these models will be stored in the BMS which will use data from the battery to provide a value for RUL. With the BMS performing these operations, computational time for the models should be considered.

Each model was scored on a weighted combination of its attributes using the equation below. The highest weight was given to the R^2 score as it relates to the explaining power of the model. Model query and train times were given lower weights but were still included as metrics for how computationally intensive the model was.

$$Model\ Score = \sum_{i=1}^{n} weights_i * \frac{features_i}{freatures_{max}}$$

Where:

Weights = [0.8, 0.6, 0.2, 0.2]
Features = [R2_test, MAE_test, query_time, train_time]

Notes:

1. *Query_time: time it took to query 1000 random points of test data.*
2. *Train_time: time to train the model.*
3. *Features for each model were divided over the maximum value for that feature across all models.*

Model	Features	R^2 Test	MAE Test	Query Time (s)	Train Time (s)	Score
LinReg	Feature Set 1	0.939	7.075	0.000534	0.000000	0.708777
DT	Feature Set 1	0.949	5.725	0.000566	0.002154	0.853669
RF	**Feature Set 1**	**0.966**	**4.475**	**0.001668**	**0.117741**	**0.980858**
NN	Feature Set 1	0.964	4.546	0.035015	4.685122	0.325088
LinReg	Feature Set 2	0.859	11.175	0.000551	0.000000	0.193915
DT	Feature Set 2	0.921	6.090	0.000617	0.000000	0.758950
RF	Feature Set 2	0.959	4.796	0.002568	0.350254	0.913955
NN	Feature Set 2	0.964	4.716	0.035349	5.085977	0.152225

Table 6: Improved Model Scores

The random forest model remained the highest performing. Of note, the neural network models suffered heavily from high training times and were even outperformed by the linear regression model for Feature Set 1. The weights were general assumptions and can be tuned dependent on the application.

Model Examination

The Decision Tree and Random Forest models were examined to see what features are the most important within them. Considering the desire to have a model that predicts RUL internal to the BMS, this will aid in prioritization of measured battery parameters. If desired, BMS developers may want to change the way key features are sampled. The accuracy, sampling rate, reliability, and redundancy are examples of design choices for measuring key features. Table 7 shows the feature importance for the highest scoring models.

Feature Set 1	DT	RF	Feature Set 2	DT	RF
Capacity	0.075	0.078	**L_charge**	**0.740**	**0.576**
Charge_time_0	**0.820**	**0.782**	b_charge	0.013	0.016
Charge_time_1	0.053	0.096	k_charge	0.011	0.014
Discharge_max_temp	0.041	0.035	X_0_charge	0.011	0.094
Discharge_avg_temp	0.010	0.010	L_discharge	0.038	0.041
			b_discharge	0.006	0.026
			k_discharge	0.140	0.123
			X_0_discharge	0.042	0.110

Table 7: Feature Importance for Decision Tree and Random Forest Models

The features charge_time_0 (time for constant voltage charge) and L_charge were the most significant. This was not surprising given that L_charge was used to scale the sigmoid function for the charge profile over a period of charge_time_0.

Applying the Results

The main drawback of this dataset was that it didn't show real world usage of the batteries. Rather, it showed the results of what happens when they are repetitively cycled through fully charged/discharged states. This is hardly an accurate representation of how batteries are actually used, but the models developed are still useful with proper implementation.

For this individual type of battery, these results showed a fairly robust modeling of RUL using parameters that are readily available in a BMS. In a real-world application, these models can be developed during battery testing prior to production. Once the battery is in it's commercial use, it can cycled fully on a periodic basis in order to get inputs for RUL modeling.

Alternatively, the battery can undergo periodic impedance testing to model RUL. This would be performed using the baseline model developed. This is less ideal because of the extra equipment needed to measure impedance which isn't typical in current BMS systems[2].

Future Expansion of Analysis

Additional testing would absolutely be required prior to implementation. The relationship between the percent of capacity discharged and RUL is unknown. Data given only involved full cycles, so the method of scaling model results of RUL would need to be explored (i.e., the relationship between a higher number of small cycles vs a lower number of full cycles would need to be accounted for).

Possible expansions for follow-on experiments include:

- Does the duration of the constant voltage charge affect battery capacity?
- What is the relationship between final discharge voltage and RUL?
- Does cycling the battery differently affect its ability to transfer energy in the long term?

Conclusion

The ability to model RUL using machine learning methods was successfully shown in this paper. Features imputed from the charge/discharge operations were successfully used to create models that outperformed the baseline model using only impedance. Additionally, this paper showed that measuring the shape of charge/discharge profiles is useful in modeling, but less effective than using measurements already available in a battery management system. Although the data was fairly limited in its scope, this paper showed the ability to develop machine learning models to predict RUL for a given battery type and use case.

References

1. Saha, B., and K. Goebel. “NASA Battery Dataset.” Www.kaggle.com, 2007, https://www.kaggle.com/datasets/patrickfleith/nasa-battery-dataset. Accessed 10 Apr. 2024.

2. Lazanas, Alexandros Ch., and Mamas I. Prodromidis. “Electrochemical Impedance Spectroscopy–a Tutorial.” ACS Measurement Science Au, vol. 3, no. 3, 8 Mar. 2023, https://doi.org/10.1021/acsmeasuresciau.2c00070.

3. “Victron Energy Lynx Smart BMS 500 Battery Management System (BMS) - M8 Busbar - VOLTAICO.” https://voltaico.com/product/victron-energy-lynx-smart-bms-500-battery-management-system-bms-m8-busbar/. Accessed 10 Apr. 2024.