Predicting the Capacity Fade Curves in Rechargeable Lithium-ion Batteries

Group 5

Abdul Fayeed Abdul Kadir (aa5042), Chris Spitler (cs4178)

EAEEE 4000 - Machine Learning for Environmental Engineering and Science

Prof. Pierre Gentine

December 22nd, 2023

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Table of Contents

List of Figures/Tables

Introduction

Since 2010, the share of renewable energy in the global power has increased by 10%. Energy storage is critical for an efficient and clean electric grid. One of the solutions in improving the use of renewable energy is by using batteries as energy storage systems. In the United States alone, battery storage has increased from 47 MW to 11,071 MW in 2023. Large-scale battery storage capacity is expected to grow to 80 GW in 2030, according to an average forecast done by the American Clean Power Association.

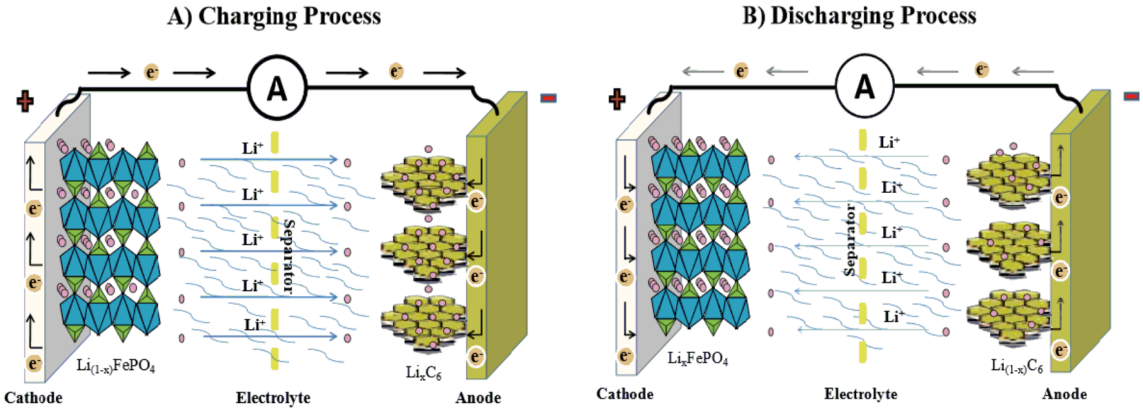

Figure 1. Schematic diagram of a Graphite/Lithium Iron Phosphate (LFP) battery [1].

One of the batteries which is commonly deployed in a wide range of applications is the lithium-ion battery. **Figure 1** shows the schematic diagram of a lithium iron phosphate (LFP)/graphite battery. LFP/graphite batteries are widely used due to their low costs [2], high energy densities, and long lifetimes [3]. As crucial as rechargeable batteries are for energy and environmental solutions, they however will undergo a degradation process during their use. It is important to make an accurate degradation analysis in order to monitor the health of the battery. The battery cycle life or the capacity loss is usually the common health indicator. Having an accurate prediction of the battery lifetime by using the early-life cycle data helps in guiding the process in battery production, validating the manufacturing processes, and meeting the end-users expectations.

However, predicting the battery degradation is not easy due to factors that are unknown, such as complex aging mechanisms and non-linear degradation with cycling. Even so, to determine the capacity fade through experiments requires thousands of charge and discharge cycles, which can take months to years. A different means should be approached to tackle this issue. There is a need to develop methods for battery capacity fade predictions using limited sets of data.

People in the field usually approached this issue through two ways: a) electrochemical approach, where one employs a conventional pseudo two dimensional model with complex partial differential equations to describe the physical phenomena that is happening within the system, and b) data-driven approach, where one would solely rely on little data and incorporate machine learning techniques to predict the capacity fade. Although the electrochemical approach would help in predicting the capacity fade and building a better understanding of the physical processes at play, building reliable physical models requires more information than is typically available in most energy storage systems as well as considerable computational power. With that, machine learning based degradation prediction provides a promising option for use in real-world applications, where data sets and computational power may be limited.

Data Analysis

The cycling dataset was available online, provided by Severson et. al [4]. They specifically used the commercial graphite/$LiFePO_4$ cylindrical batteries in collecting the cycling data. In total, 133 batteries were used with cycle life ranging from 326 to 2284. The nominal capacity of the battery is 1.1 Ah and the nominal voltage is 3.3 V. The cycling was done at an ambient temperature of 30 °C and the batteries were all discharged with the same protocol, which was a constant current constant voltage (CCCV) method, where the batteries were discharge at 4C down to 2 V and held at 2 V until the current meets the cutoff at C/50. Before the cells were discharged, each cell was charged using different protocols, where they were charged with one or multiple constant current steps up to 80% of the state of charge, followed by 1C CCCV charge up to 3.6 V with a charge cut-off at C/50. Aside from the cycle life, voltage, current, charge and discharge capacity data, the temperature (minimum, maximum, average) and internal resistance of each cell were also provided and analyzed. **Figure 2** shows the a) degradation in discharge capacity with cycles for all batteries and b) the variations in cycle life among the datasets.

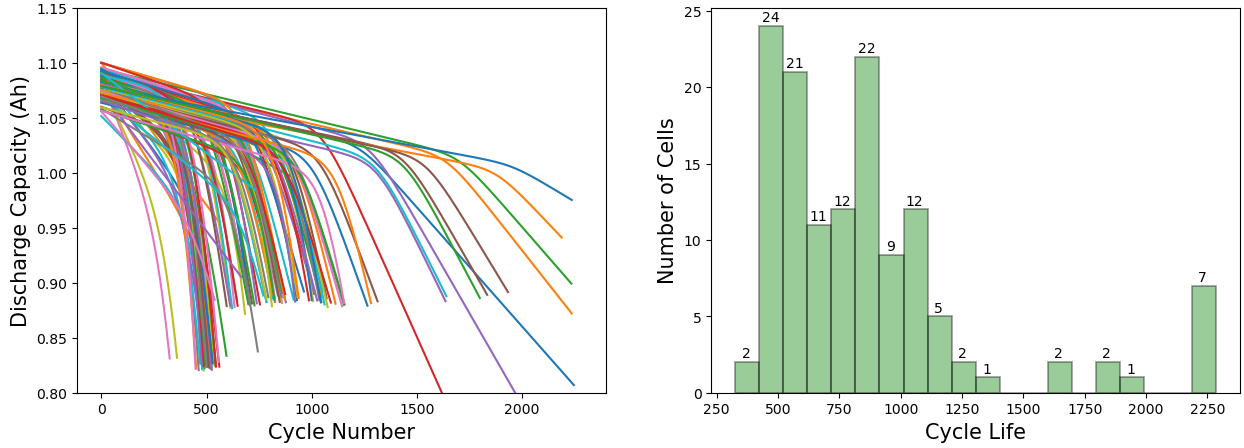

Figure 2. Degradation in discharge capacity (left) and cycle life (right) of the batteries.

The cycling data shown in **Figure 2** represents the original data. They were further filtered by removing the data points with discharge capacity below 80% of the nominal capacity (0.88 Ah). This was done to ensure that all of the datasets are in the same state of discharge (20%), regardless of their actual cycle life. There are 26 cells with cycle life of less than 500 and 12 cells with cycle life greater than 1,500. The remaining cells have a cycle life of between 500 and 1,500. The machine learning model requires training, validation, and testing datasets for model training, model selection/hyperparameter tuning, and model performance evaluation respectively. 79 cells are used for training (60%), 27 cells for validation (20%), and 27 cells for testing (20%).

In this work, multiple machine learning models will be used, as will be further discussed in the next section. Each model requires a different kind of input from each other, although they were all coming from the same dataset explained previously. They include: a) a 100 x 100 data of discharge capacity for the first 100 cycles varying within 2.0 and 3.5 V for each cell, b) a difference in discharge capacity across the set voltage range between the 100^{th} and 10^{th} cycle of each cell, and c) the variance in the difference of the discharge capacity throughout the voltage range recorded between the 100^{th} and 10^{th} cycle. From these inputs, the output was expected to be either a) the capacity fade curve past the 100^{th} cycle, b) the mathematical expression of the capacity fade curve (parameters to be fit), or c) the end-of-life of each battery cell.

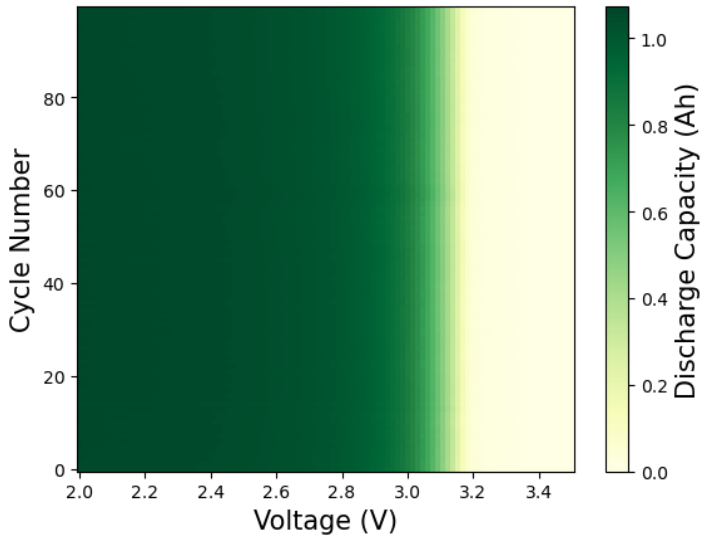

Figure 3. A 100 x 100 discharge capacity data for the first 100 cycles.

Figure 3 shows the 100 x 100 discharge capacity data for the first 100 cycles in one cell. Before obtaining this data matrix, the original data was first filtered from cycles that were noisy. Then, it was further interpolated with the same 100 data points for each cycle within a set voltage range of 2.0 to 3.5 V. This then repeated for all cells, then later split for training/validation/testing in the machine learning model.

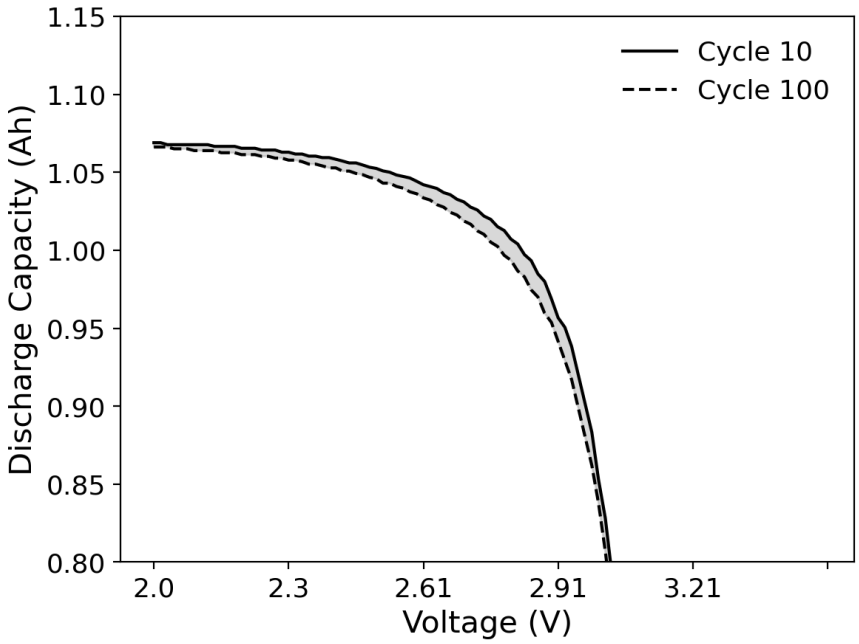

Figure 4. Shift in discharge capacity at 10th and 100th cycle.

An alternative input that is considered in this work is the difference in the discharge capacity between the 100th and 10th cycle. From **Figure 4**, it could be seen that there is a slight shift to the left in the capacity curve, which dictates a capacity loss within the first 100 cycles. From this observation, it seems reasonable to consider the difference in capacity between these two cycles as the input for one of the machine learning models. **Figure 5** shows the difference in the capacity throughout the specified voltage of 2.0 to 3.5 V.

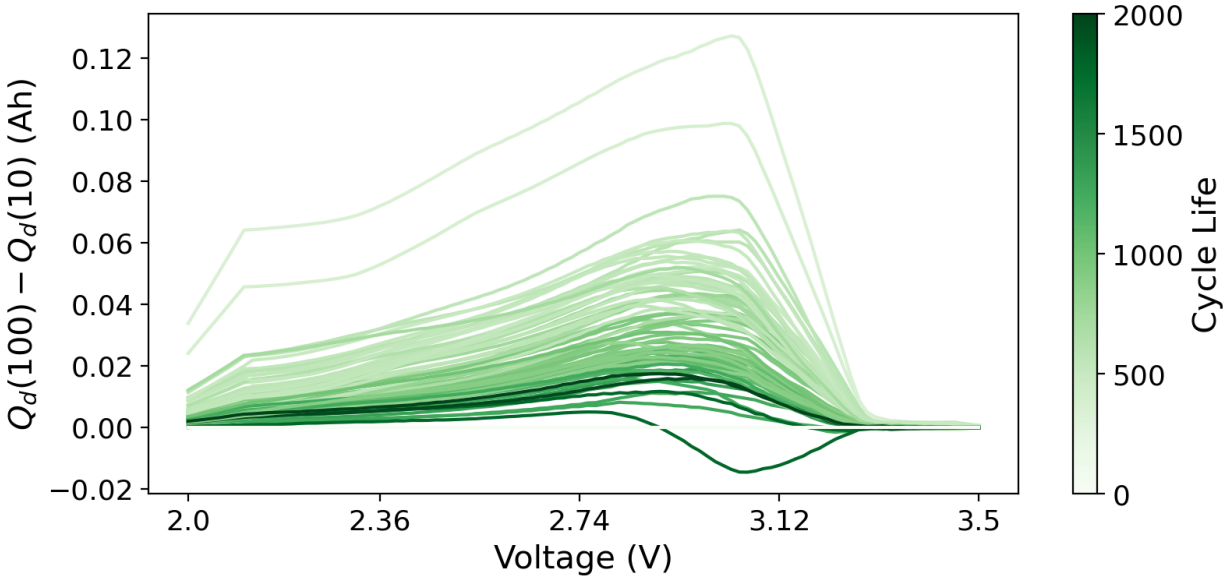

Figure 5. Discharge capacity difference between 100th and 10th cycle.

Other than that, instead of considering the entire difference in discharge capacity throughout the voltage range between the 100th and 10th cycle only as shown in **Figure 5**, the variance in the differences in capacity in the first 100 cycles was determined and used as the input for machine learning models. This is done to make the model simple by just handling a single data point as the data input for the capacity trend, and allows this input to be easily paired with other potential input values, such as temperature, current, and internal resistance. **Figure 6** shows the capacity difference variance for all cells compared with their cycle lives, showing that it may serve as a potential predictor based on the correlation of the two values.

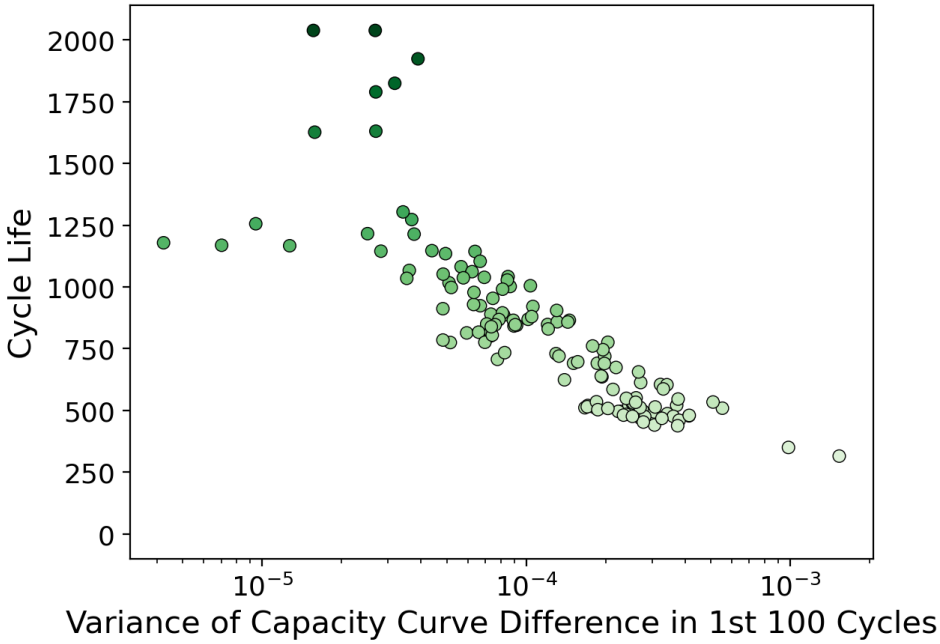

Figure 6. The variance in discharge capacity curve difference in the first 100 cycles.

Moreover, instead of dealing with the cycling data per cycle for each cell, another approach that is relatively simpler would be to consider the capacity fade trend as a whole. Each cell has a different cycle life, thus will have different length in discharge capacity across the cycle life. Most of the data also contain noise that will affect the machine learning model prediction if they were not cleaned up priorly.

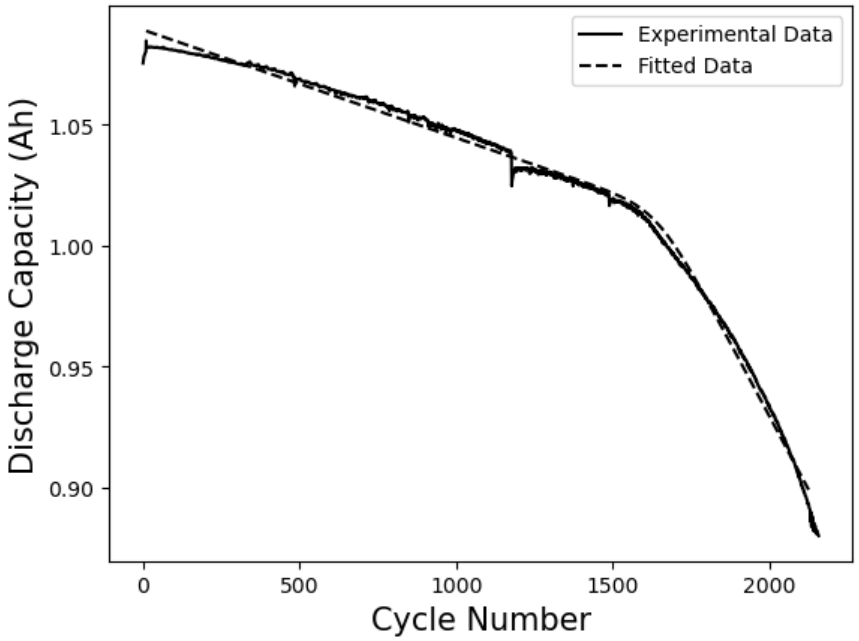

Figure 7. Experimental and fitted capacity fade of one of the battery cells.

$$q = m_0 n + (m_f - m_0) \cdot \delta \cdot ln\left[\frac{exp\left(\frac{n}{\delta}\right) + exp\left(\frac{N_k}{\delta}\right)}{1 + exp\left(\frac{N_k}{\delta}\right)}\right] + q_0 \qquad \textbf{Eq. 1}$$

From **Figure 7**, the experimental data from one of the cells has a sudden drop in discharge capacity around the 1,200th cycle due to a pause during data collection and continued the next day instead. This data can be easily removed by refitting the entire data by following **Eq. 1**, and interpolate within a fixed number of data points, in order to have the same length for each cell in the entire dataset. **Eq. 1** contains 5 main parameters, (m_0, m_f, N_k, q_0, δ) which are the initial slope, failure slope, rollover cycle, initial capacity, and the rollover width of the capacity curve respectively. **Eq. 1** takes the cycle number n to obtain the capacity q at that cycle number. When $n = 0$, which represents the first cycle (starts at 0 following Python indexing), the capacity will be q_0, as expected. To reduce the amount of variables to fit in the equation, δ is set to 50. The other parameters are allowed to vary within the following range:

$$m_0 = [-10^{-3}, 0],\ m_f = [-2 \cdot 10^{-2}, 0],\ N_k = [0, 3000],\ q_0 = [1.05, 1.1]$$

The range was chosen to ensure that it still follows the physical meaning of the variables in the capacity fade curve, where the slopes are varied within the negative range as it is sloping downwards, the rollover cycle is within the positive and hundreds/thousands range, and the initial capacity is within the nominal capacity of 1.1 Ah.

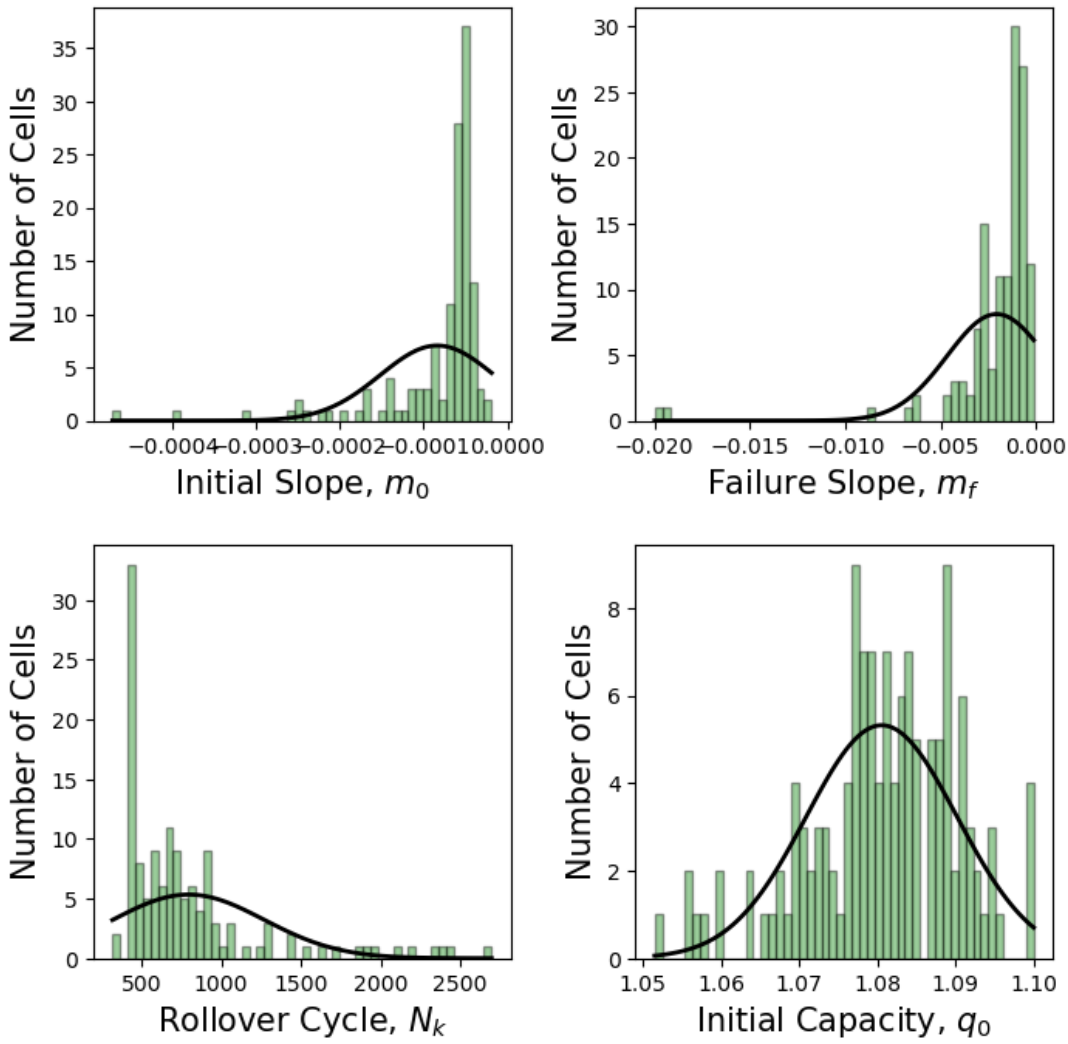

Figure 8. Parameters of degradation in discharge capacity for 133 cells.

Figure 8 summarizes the values of parameters for the discharge capacity fade curve for all cells in the datasets. The black curve lines represent the normal distribution of the parameters within the datasets. These parameters will be as inputs to one of the machine learning models.

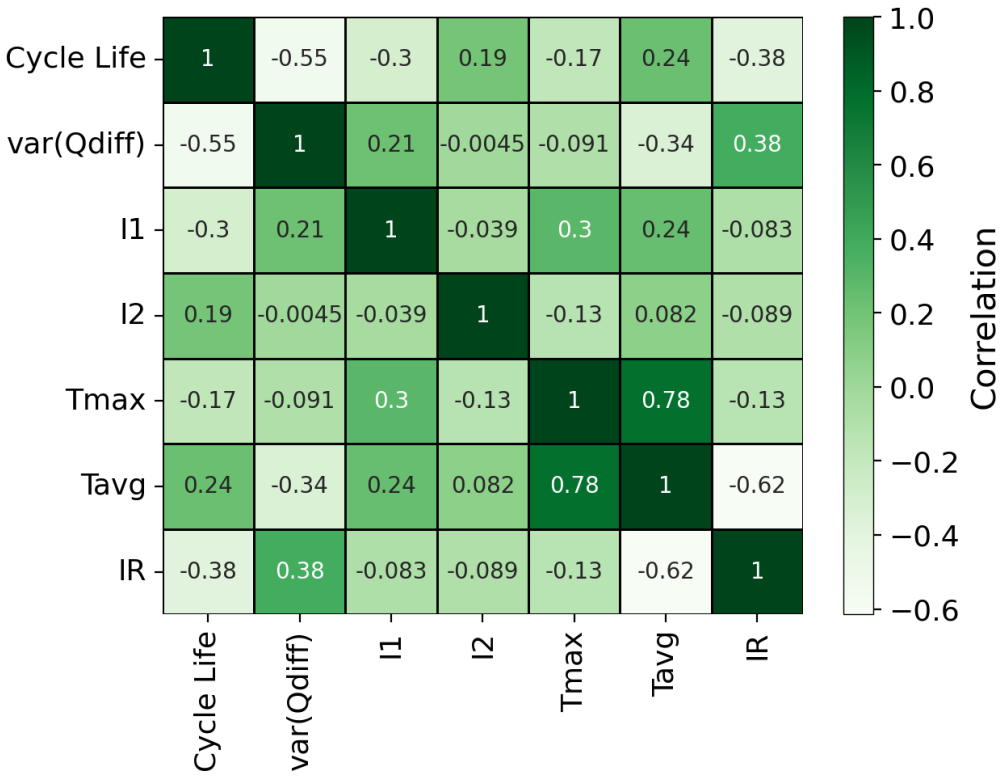

Figure 9. Correlation matrix for potential model input variables.

The correlation of all the available input variables was compared with cycle life and the most significant variable appears to be the variance of the discharge capacity, with internal resistance also having a decent correlation with cycle life. Some of the physical aspects of the system are visible in this diagram, for example internal resistance and average temperature are somewhat correlated, and maximum charge current (I1) and max temperature are also somewhat correlated. These indicate that the average temperature is more representative of the battery's electrical performance, while the max cycle temperature is a result of how quickly the battery is charged.

Methods

Once the data is preprocessed, there are multiple ways it can be fed into the various machine learning models. In all cases, the goal of the exercise is to use data from the first 100 cycles that each battery is operated on in order to predict how many cycles it will continue to operate before it degrades to below 80% of its initial capacity. The preprocessed discharge capacity data comes in the form of one capacity value for 100 different voltage values for each cycle. As the batteries are cycled and they degrade, the voltage at each discharge capacity will also change. These discharge capacity vs. voltage curves (QV curves) can be fed directly into a model as a 100 x 100 array for each cell (**Figure 3**). Alternatively, the difference can be taken between the QV curves

at the 10th and 100th cycles (**Figures 4** and **5**), referred to here as "Qdiff", to get a sense for how the QV curve has changed over the first 100 cycles while allowing a more concentrated amount of data to be fed into the model (100 points per cell). An even more boiled down version of this input can be used in the form of the variance of the Qdiff curve, which reduces the original 100 by 100 array to one data point per cell (**Figure 6**).

In addition to the discharge capacity data, current data from the charge cycles can be fed into the models. All of the cells from the dataset were discharged according to the same protocol, but several different charging routines were used for the different cells which can be a major factor in the rate at which a battery degrades. The two most prominent current values from each cycle are available (I1, I2). The temperature of each cell was recorded as they were cycled, and the maximum and average temperatures of each cycle can be used as model inputs (Tmax, TAv). Finally, the measured internal resistance of the cells at each cycle can be inputted (IR). A correlation matrix was created to determine the relative impact that each of these variables has on cycle life (**Figure 9**), and to examine how they may correlate to one another.

The output data can take one of three forms. A prediction can be made for the discharge capacity at every cycle past the 100th cycle, returning a vector of capacity for as many cycles as desired. Similarly, a function can be used for estimating the capacity at future cycles. The outputs of the machine learning model in this case are parameters of the function which fit the curve to a cell's discharge capacity loss (**Eq. 1**). The simplest option is to merely predict one number for the cycle life of each cell. All three methods were attempted here, and each one lends itself to different machine learning techniques.

Convolutional Neural Net

As the QV data is originally in a 100 by 100 form, it lends itself naturally for use in a CNN model which may extract features of the QV surface using 2D kernels. The first goal for the CNN was to reproduce the results of Saxena et al. [5], who produced a parameterized function for capacity curves based on the same dataset using a CNN with 100 by 100 QV curves as the input. A similar CNN design was used for reducing the 2D array to a vector for the output, yet different hyperparameters for learning rate, activation function, and loss function were used based on which gave the best performance via the Keras Tuner (**Table 1**). Four convolutional layers were used, the first having a non-overlapping kernel of size 5, and subsequent layers using kernels of size 3 and stride 1. Batch normalization was performed after each convolutional layer to normalize the features and stabilize gradients. Max pooling layers following the convolutional layers were used to extract the max value of each feature. Finally, the data is flattened and passed through two dense layers to achieve the desired output shape. Other CNN designs were also tried with fewer convolutional layers and without max pooling layers, however no improvement to performance was seen.

Table 1. Hyperparameters used for CNN with single cycle life value output.

Varied Hyperparameters	Potential Options	Best Value
Minibatch Size	1, 4, 16, 40	1
Epochs	100,200,300	100
Learning Rate	1e-1, 1e-2, 1e-3, 1e-4	1e-3
Conv Activation Function	Relu, Tanh, Sigmoid	Relu
Dense Layer Activation	Linear, Sigmoid	Linear
Loss Function	MSE, MAE	MAE
Fixed Hyperparameters		**Fixed Value**
Kernel 1 Size		5
Kernel 2 Size		3
Max Pool Size		2
Conv Layer Filters (4)		16, 32, 64, 64

The same CNN design was also used to try and predict the entire discharge curve explicitly. The four output parameters were thus replaced with 2048 values for the capacity at each cycle. Finally, a prediction of a single value for the overall cycle life of each cell was attempted. Tuned hyperparameters displayed are for the single value prediction model.

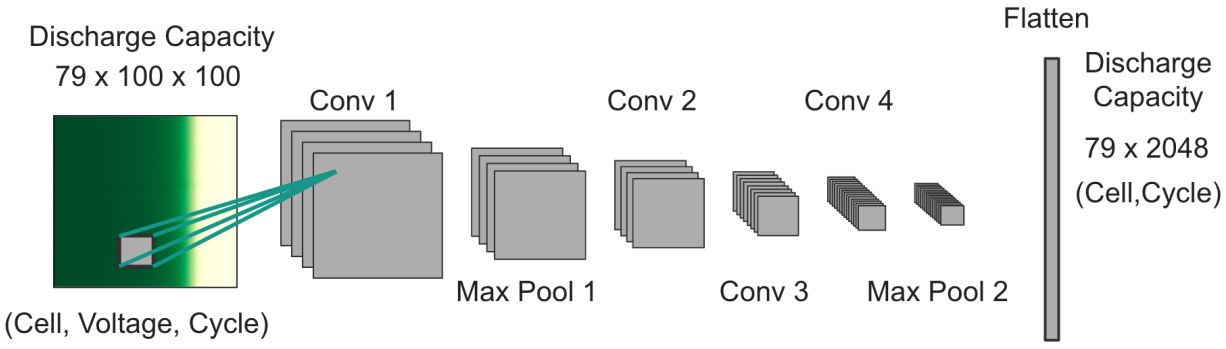

Figure 10. Schematic representation of CNN used to predict full discharge capacity curves. Other versions of the model have the output replaced with parameters for the discharge function (4 data points), or the cell cycle life (1 data point).

Neural Network

A series of neural net designs were attempted for generating the same types of outputs as for the CNN. Different inputs were used for the NNs than for the CNNs, as the NNs require a flat input. First, a flattened version of the 100 by 100 QV curves (10,000 points) was used. Next, the 100 point Qdiff curve described above was tried as an input. Finally, the variance of the Qdiff curve, along with current, temperature, and internal resistance data was used as an input. Hyperparameter tuning was performed, now with varying numbers of dense layers having varying numbers of neurons. Details of the hyperparameter tuning for the version of the model using the variance of Qdiff as an input and the single value cycle life as an output are shown in **Table 2**.

Table 2. Hyperparameters used for NN with single cycle life value output.

Varied Hyperparameters	Potential Options	Best Value
Minibatch Size	1, 4, 16, 40	1
Epochs	100, 200, 300	300
N Layers	2, 3, 4, 5, 6	6
N Neurons	16, 32, 64, 128, 256	16
Learning Rate	1e-2, 1e-3, 1e-4, 1e-5	1e-3
Activation Function	Relu, Tanh, Sigmoid	Relu
Loss Function	MSE, MAE	MAE
Output Layer Activation	Linear, Sigmoid	Linear

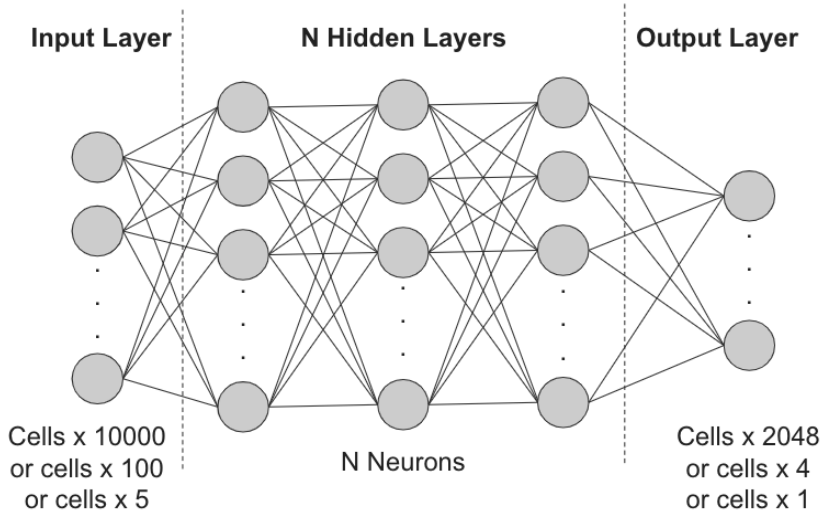

Figure 11. Schematic representation of NN to predict full discharge capacity curves.

Regression: XGBoost

Extreme gradient boosted decision tree regressors (XGBoost) were used for cycle life prediction as well. XGBoost is a popular decision tree method which uses ensemble methods, tree pruning, and regularization for an ideal balance of speed and performance. Models were created both using Qdiff curves as an input and using the variance of Qdiff along with current, temperature, and internal resistance data. The only output for the attempted XGBoost models was the single value cycle life. Hyperparameter tuning was performed using the Hyperopt package and the details for the hyperparameters used for the best performing model are shown in **Table 3**.

Table 3. Hyperparameters used for XGBoost with var(Qdiff), I1, I2, TAv, IR inputs.

Varied Hyperparameters	Potential Options	Best Value
N Estimators	20-100	20
Learning Rate	0.05, 0.1, 0.15, 0.2	0.2
Gamma	1-9	8.76
Max Depth	5, 10, 15, 20	10
Subsample	0.6, 0.7, 0.8, 0.9, 1	0.6
Lambda	0.7, 0.8, 0.9	0.8
Alpha	0.1, 0.2, 0.3	0.3
Evaluation Metric	RMSE, MAE, MAPE, Logloss	RMSE

Results and Discussion

As multiple sets of inputs and outputs were used for the various model types, it is most convenient to compare results which seek to give the same outputs. **Figure 12** shows the loss curves for CNN and NN models used to predict a single value for cycle life using the best hyperparameters found from the Keras tuner. Neither model type was able to converge between training and validation loss.

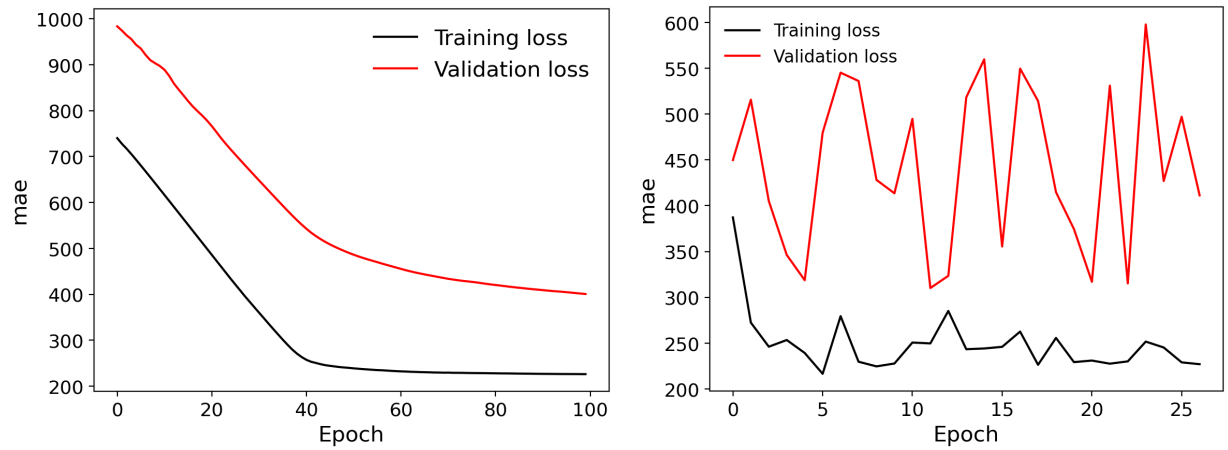

Figure 12. Loss plots for CNN prediction (left) and NN prediction of cycle life (right).

As is expected based on the loss plots, the predictions of the NNs and CNNs did not match the test data very well, as shown in **Figure 13**, and the best performing models of each type simply seemed to be predicting the mean value of the test data in order to minimize loss. Virtually no difference is seen for the predicted cycle life from cell to cell, indicating the models are not functioning properly.

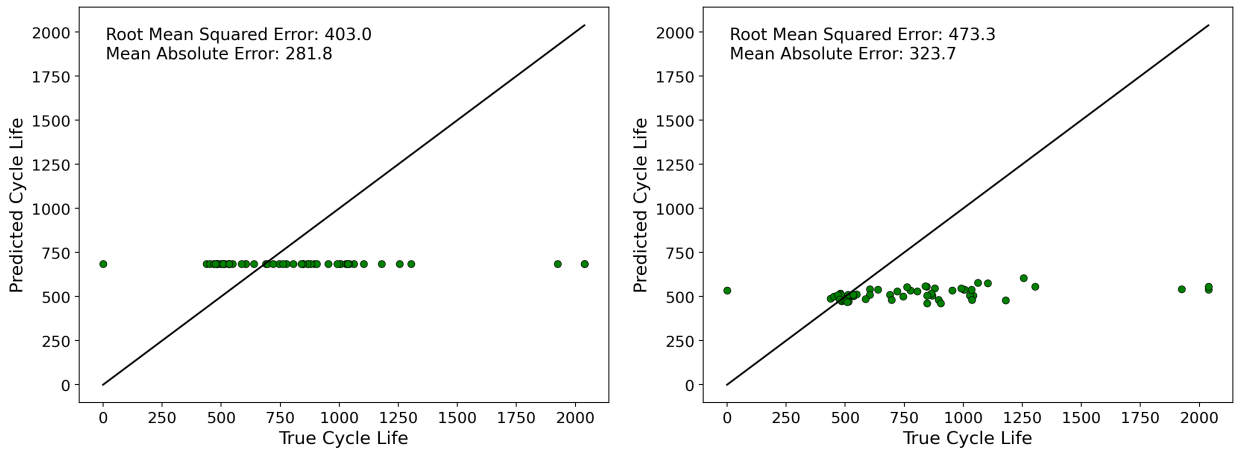

Figure 13. Cycle life predicted by the CNN (left) and NN (right) vs. actual test cycle life.

A loss plot and CNN prediction of a parameterized capacity loss function is shown in **Figure 14**. The same problem is seen in this choice of outputs, where the model simply outputs a value close to the mean of the training data in order to minimize the loss. Only one predicted loss curve is shown in the plot, as they all overlap one another.

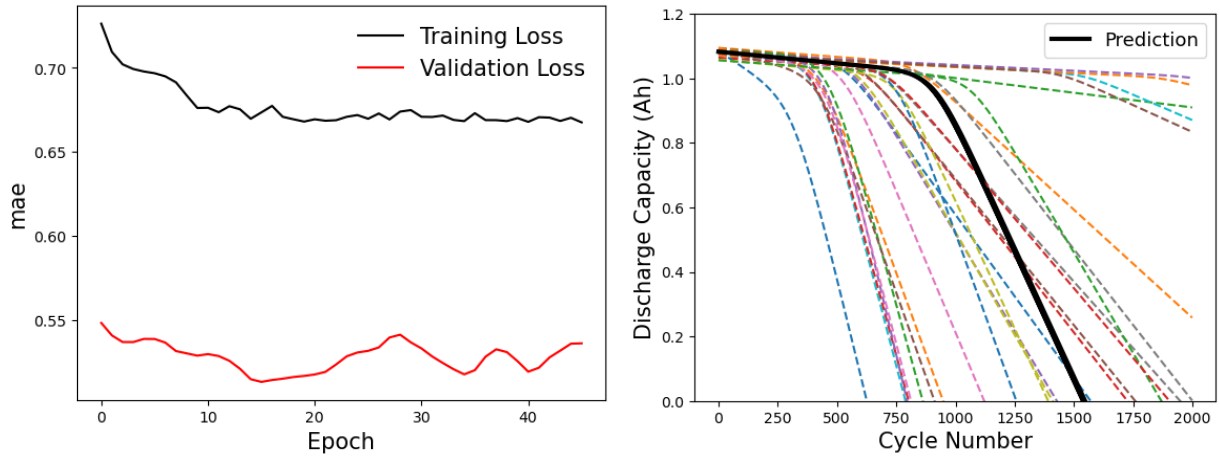

Figure 14. Loss curve (left) and CNN predicted capacity degradation function (right).

Results of XGBoost models are shown in **Figure 15**. XGBoost performs much better than any of the NNs and CNNs attempted thus far. The regression tree approach is able to actually distinguish between cells and give reasonable cycle life predictions which match the test data. The rather noisy data has several outliers which may skew the prediction somewhat, so further refinement of the model may be possible.

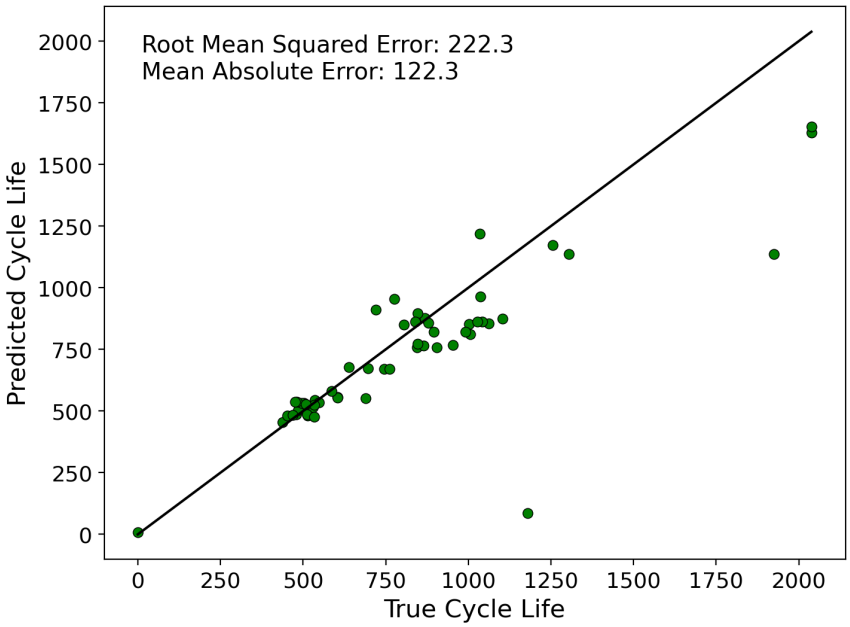

Figure 15. Predicted vs. actual cycle life by best performing XGBoost model.

Conclusion

The objective of this work was to build and train machine learning models which could use data from the first 100 charge and discharge cycles of a battery's use and predict the number of cycles the battery could remain in use until it degrades to its end of life, determined to be 80% of its initial capacity. Various types of input data were attempted, including capacity fade curves, cell temperature, charge current, and internal resistance. Three machine learning methods were used, neural networks, CNNs, and XGBoost regression trees.

Unfortunately, all attempts in creating neural networks and CNNs which could accurately predict battery cycle life or degradation curves based on information from the first 100 cycles of battery operation were unsuccessful. The models were unable to create predictions that differed significantly from cell to cell, and when the structure of the data and model was changed to that the outputs varied across cells, the output was more or less random. In the cases where the loss function was properly minimized, it appears that the models are just predicting the mean of the data. One possible reason for this could be that the training data is being sampled from a poorly representative section of the data, however repeatedly changing the randomizing seed does not produce any significant change in the output. It is also unlikely that the problem stems entirely from the dataset, as other successful works have been published using this exact dataset. Another possibility is that the preprocessing of the data was done incorrectly, as the data was challenging to sort through. This is also unlikely, as the XGBoost model was able to successfully create good predictions of cycle life with the same inputs and outputs as the NN. Thus, the model architecture is likely the cause of the problems, and further refinement may return better results.

The XGBoost model performed fairly well with very few tweaks being required other than the hyperparameter tuning, reaching values of RMSE and MAE of 222 and 122, respectively. Ideally, the NNs and CNNs could be refined to perform similarly well in the future, as they are more memory efficient and portable than regression trees. NNs and CNNs could potentially be used with transfer learning to apply similar models to other battery chemistries or batteries used in different systems, such as in electric vehicles or mobile devices. Future work may also investigate LSTMs applied across cycle data for predicting future performance.

Code Availability

[Abdul's Github] [raw_mat_fileData]

References

[1] S. Sepasi, Adaptive state of charge estimation for battery packs, (2014). https://doi.org/10.13140/RG.2.1.4737.5209.

[2] ACP, Clean Energy Storage Facts, ACP. (n.d.). https://cleanpower.org/facts/clean-energy-storage/

[3] Renewables in Electricity Production | Statistics Map by Region | Enerdata, (n.d.). https://yearbook.enerdata.net/renewables/renewable-in-electricity-production-share.html

[4] K.A. Severson, P.M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M.H. Chen, M. Aykol, P.K. Herring, D. Fraggedakis, M.Z. Bazant, S.J. Harris, W.C. Chueh, R.D. Braatz, Data-driven prediction of battery cycle life before capacity degradation, Nat. Energy. 4 (2019) 383–391. https://doi.org/10.1038/s41560-019-0356-8.

[5] S. Saxena, L. Ward, J. Kubal, W. Lu, S. Babinec, N. Paulson, A convolutional neural network model for battery capacity fade curve prediction using early life data, J. Power Sources. 542 (2022) 231736. https://doi.org/10.1016/j.jpowsour.2022.231736.