
Battery Lifetime Prediction with Limited Cycle Data

Jinpu Cao^{*1} Solomon Kim^{*2} Zewen Zhang^{*3}

Abstract

Accurately predicting the remaining useful lifetime of batteries is critical for accelerating technological development and creating a paradigm shift in battery usage. Data-driven approaches, based on large datasets, provide a physical-model agnostic way to predict the health status of batteries with high accuracy. However, most data-driven methods on battery life prediction are limited by the relatively small dataset available. More importantly, such models often rely on features extracted from a hundred cycles worth of data for a given cell, making it computationally inefficient and incompatible with on-board application. Herein, we augment a battery dataset of 124 cells through a sliding window method to mimic the real-world situation where only recent cycle history is available or could be obtained within a reasonable timeframe. We applied machine-learning models, including linear regression, random forest regression, convolutional neural networks, and recurrent neural networks to make predictions on cell life. Our best model achieve a 7.5% prediction error given the data of only 5 cycles. We varied the number of given cycles to be 5, 10 and 20 allowing us to understand how increasing the number of cycles relate will affect model accuracy. Our results indicated that increasing the number of cycles does not necessarily increase the prediction accuracy of our models. Also, we tested the extreme case with only one cycle available using the best model architecture, and obtained an error of 12.1%.

1. Introduction

The development of clean energy generation, transmission, and distribution technologies that reduce carbon emissions is indispensable in tackling the impending climate crisis and is critical for a sustainable future (Chu et al., 2017). Battery technology emerges to be one of the core pieces in this process. On the one hand, the intermittent nature of renewable energy resources urges the use of batteries as the energy buffer to help the electricity sector decarbonize the grid. On the other hand, electric vehicles are replacing vehicles that use traditional combustion engines. These efforts stimulated a growing demand for long-lived batteries, both for driving and for grid buffering (Berecibar, 2019). However, the long nature of battery lifetime entails delayed feedback of performance, often months to years, making optimization of battery manufacturing and design experimentally almost intractable in time. Reliable methods to diagnose battery health status are becoming increasingly important. Moreover, the advancement of such methods could also provide new opportunities for end users to estimate battery life expectancy, promoting new business models like battery swapping in electrical vehicle industry. As a result, the accurate assessment of battery health is critically important (Ng et al., 2020).

A figure of merit known as state of health (SOH) is used to quantify the ability of a battery to store energy, relative to its initial or ideal conditions. SOH is reported as a percentage: it is 100% when the battery is new, but decreases over time. Generally, when SOH drops to lower than 80% of the initial value, the battery has reached the end of its service life (Berecibar, 2019). The remaining number of charge/discharge cycles until the battery reaches end of life is the remaining useful lifetime of the battery. It represents the period from the observation to the end of life, and is also the most desirable parameter to estimated accurately. Battery degradation, which is highly associated with the remaining useful lifetime, is a complex process controlled by electrochemical reactions, and therefore can be highly non-linear and affected by many factors, such as temperature, charge, and discharge rate (Roman, 2021). Ultimately, these multiple factors complicate battery lifetime prediction.

^{*}Equal contribution ¹Department of Computer Science, Stanford University, California, United States ²Department of Civil and Environmental Engineering, Stanford University, California, United States ³Department of Materials Science and Engineering, Stanford University, California, United States. Correspondence to: Jinpu Cao <jinpu@stanford.edu>, Solomon Kim <solomon3@stanford.edu>, Zewen Zhang <zwwzhang@stanford.edu>.

2. Related Work

Conventionally, battery lifetime has been modelled by mechanistic and semi-empirical methods developed for ideal physical models to describe batteries (Ramadesigan et al., 2012; Waag et al., 2014). Despite success in these early models, accurately predicting full cell batteries in relevant cycling conditions remains challenging due to coupled thermal, mechanical, and chemical heterogeneity in cells.

Increasingly, researchers in the battery space have experimented with data-driven methods to make better analysis on battery lifetime (Nuhic et al., 2013). Despite the progress in developing more accurate and fast models for SOH estimations, there remains a clear trade-off between the computational efficiency and the accuracy of model-based predictions. Recently, data-driven models have drawn much attention. Combined with machine learning techniques, these models are able to make predictions without prior knowledge of the system. Machine learning techniques—including neural network, support-vector machine, random forest and regression techniques—have been applied to predict the RUL of batteries.

Notably, Severson et al. have tackled this challenge by generating a comprehensive data set that characterizes the performance of 124 commercial lithium-ion batteries (more specifically, batteries that use lithium iron phosphate as the cathode material and graphite as the anode) as they undergo cycles of charging and discharging (Severson, 2019). The authors used a variety of fast-charging conditions (but identical discharging conditions) to alter the cycle lives of the batteries—the number of cycles that a battery goes through before it reaches the end of its life. A wide range of cycle lives (from 150 to 2,300) was therefore captured in the data. The researchers then used machine learning to analyse the data, and thereby produced models that can reliably predict battery cycle lives. Notably, the authors analysed data from only the first 100 cycles of each experimentally characterized battery—which was before the batteries showed clear signs of capacity fade. The best model correctly predicted cycle lives for about 90.9% of the batteries in the study.

While the researchers focused on brand-new batteries and predicted their lifetime, there are more areas that we can explore. More specifically, we can diagnose a battery that's already been in use. In addition, the authors used data from the first and the hundredth cycle for the prediction. Our goal is to get accurate results with measurements from much less consecutive cycles, pushing the model to the limit and making the model much more applicable in the real world.

3. Summary of Contributions

Herein, we take advantage of this increasing shift towards machine learning approaches by experimenting with a vari-

ety of models trained on a varying range of cycles to predict the remaining useful lifetime of batteries (Berecibar, 2019). To clarify, we mimic the situation in which you are handed a battery that you have never seen before and are asked to predict its remaining life of battery. Considering that in practice it takes around a day to fully charge and discharge an EV battery, the lower the number of cycles needed for a reasonable estimate on remaining life, the better. Specifically,

- We augmented the dataset by generating many samples from one cell data through a sliding window method.
- We evaluated the dependence of prediction accuracy on the number of cycles given to understand the limit in this problem setting. We tested how increasing the number of cycle data in the sliding window would increase the accuracy of a machine learning model.
- We experimented with a variety of machine learning methods including linear regression (LR), random forest regression (RFR), convolutional neural networks (CNN), and recurrent neural networks (RNN) to validate our results.

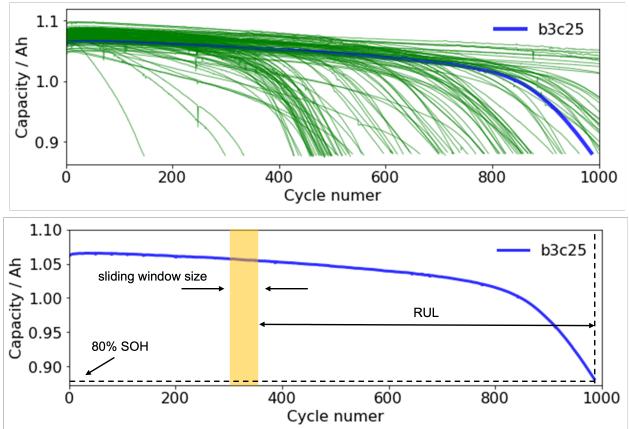


Figure 1. Battery cycle data and the schematic for sliding window method for data augmentation

4. Method

4.1. Dataset

We used the same dataset from Severson et al. Specifically, this dataset contains 124 A123 Systems cells, model APR18650M1A, 1.1Ah nominal capacity, cycled in a temperature-controlled environmental chamber at 30 degrees celsius under various fast-charging conditions but with identical discharging conditions of 4C to 2.0V, where 1C is 1.1A (Severson, 2019). Cycle lives using 72 different

fast-charging conditions were collected. Within these 72 different fast-charging conditions, charging rates ranged from 3.6C to extreme fast-charging (XFC) conditions of 6C. By varying charging conditions, this creates a dataset that has a wide range of cycle lives, averaging cycle life around 806 with a standard deviation of 377 ranging from approximately 150 cycles to 2,300 cycles (Severson, 2019). Temperature controlling these different fast-charging profiles varied by around 10 degrees celsius within the cycle due to the large amount of heat generation during charge and discharge for this dataset (Severson, 2019). Also, Figure 1. highlights the distribution of the data before data augmentation as well as after splitting the data with a sliding window of 5, 10, and 20 cycles. Also, Figure 2. demonstrates capacity as a function of time for various cells and specifically for cell['b3c25'].

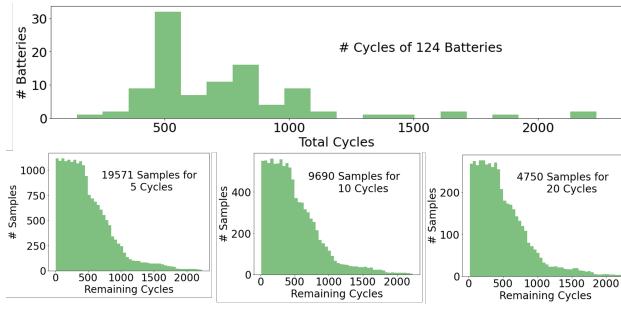


Figure 2. The distribution of cycles in dataset

4.2. Data Processing

Given the noisiness in the dataset, we preprocessed the data in order to remove small outliers as well as cycles that had gaps in time. Additionally, we also applied a savitzky golay filter to additionally smooth out the noise. The savitzky golay filter utilizes convolutions to fit low-degree polynomials to adjacent data points in space. Since this dataset only contains the cycle history from 124 cells, which is not large enough for a typical neural network based method. We augmented the dataset by applying a sliding window method. Instead of using only the first few cycles as the input, we took multiple consecutive charging cycles as input, and use the corresponding remaining life cycles as the label. For example, imagine if k (the size of the sliding window) is 5. Then, given the cell['b01c'] we use the cycle data from the 1st to the 5th cycle as one sample, the cycle data from the 6th to the 10th as another, and the cycle data from the 11th to the 15th as another, etc. We continue extracting windows of data as samples until we hit the $2 * k$ cycle from the end. With this data augmentation method, we greatly increase the size of our dataset to make it suitable for deep neural network models.

Considering that we do not use the overlapping regions in a

cell, our data augmentation approach is legitimate and we therefore do not suffer from the problem of looking into the test set while training the model. We only look into different battery cycle histories and make predictions based off that.

Additionally, we were dealt with 3 time scale features of charge, voltage, temperature and 4 scalar features of internal resistance, total charge, total time, and temperature statistics. For each time scale feature within a cycle, there are 1,000 data points. In other words, given cycle 10 of cell['b01c'], there are 1,000 data values for charge, 1,000 for voltage, and 1,000 for temperature. Also, linear interpolation was used to fill in the gaps within the data. Moreover, the label provided was the remaining number of cycles. Given $k = 5, 10$, or 20 , we iterated through each cell and extracted data from the k number of non-overlapping cycles for all the cells. Thus, we were able to increase the size of our dataset significantly by performing "data augmentation".

For the convolutional neural network approach, we preprocessed the data so that the 3 channels were associated with charge, voltage, and temperature, with each channel having dimensions of k by sample size.

4.3. Model

Here, we experimented with models used in machine learning tasks to determine the best structures to capture the underlying information within one cycle and in between cycles.

In order to benchmark our performance, we first created a simple linear regression model as the baseline. We then compared the baseline model to other algorithms more specifically, RFR, CNN and RNN. We are particularly interested in RNN due to how it incorporates an element of time/sequence which we thought would be valuable given the sequential nature of cycle data. Note that in all our machine learning based models, we use root mean squared errors as the loss function for training.

4.3.1. LINEAR REGRESSION

We start off with a simple linear regression model for our baseline. Given an input X which is an n by p matrix as well as an output Y , where n is the number of samples that have been extracted using the data augmentation approach specified in section 4.2. In our case, p was the number of total features after concatenating T , Q , and V as the input for the LR model.

4.3.2. RANDOM FOREST REGRESSION

The RFR algorithm utilizes an ensemble approach to average the output of multiple trees therefore accurately determining predicted value. Considering that the ensemble aspect of this model helps against noisy data as well as

adds more robustness, we thought that this could be a legitimate way to approach battery lifetime prediction. The RFR algorithm can be used for both classification as well as regression. The algorithm learns features that can best be used to construct the decision points within the tree.

4.3.3. CONVOLUTIONAL NEURAL NETWORK

The design concept of a CNN model for this regression task is that convolutional kernels are capable of capturing certain patterns within channels, we hope to utilize the same methodology to capture certain patterns intra and across measurable features for a battery that might be of great importance to determine the battery RUL.

Given that CNNs operate by feeding in different channels corresponding with some characteristic, we modeled our input data into a multi-channel format. Specifically, we use the temperature (T), discharge capacity (Q), and voltage (V) as the three channels. For each channel, the height of the image is the size of the sliding window, and the width is 100 steps we sampled uniformly from the 1000 step points in one discharge cycle.

For the scalar features, internal resistance (IR), discharge capacity (QD), and discharge time (DT), we concatenate them in a similar fashion in the window direction to produce an array that combines with the feature embeddings of the time-dependent features.

The model we experimented consisted of two parts, a CNN model taking care of the time-dependent feature embedding, and a linear model taking care of the scalar features. We used a stochastic gradient descent optimization during the training. We tuned the kernel size and the structure of our fully connected layers to optimize the model performance. Also, Figure 3. presents the schematics for the CNN-based models.

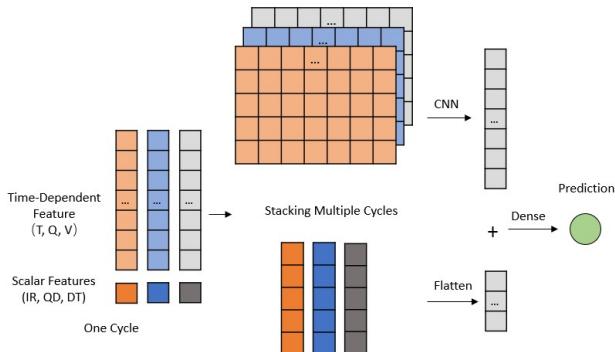


Figure 3. The schematic for the CNN-based model

4.3.4. RECURRENT NEURAL NETWORK

RNNs can be used to capture temporal dynamic behaviors through the use of internal states, i.e., memory, to process sequence inputs of variable lengths (Tealab, 2018). The battery charge and discharge data used in this study shows temporal characteristic both within one cycle and also upon several consecutive cycles. Within a charge-discharge cycle, the data contains the continuous record of discharge capacitance and temperature in a fixed voltage window. For consecutive charge cycles of one cell, the change of certain patterns between adjacent cycles can also be regarded as special sequence data. RNNs can run different programs to process different sequences of inputs since they are theoretically Turing complete (Hyötyniemi, 1996). This study is a multi-input, one output model (remaining cycles). Therefore, a many-to-one RNN model is built with reference to the semantic classification model.

Long short-term memory (LSTM) and Gated Recurrent Unit (GRU) are variants of RNN (Hochreiter & Schmidhuber, 1997), which can solve the vanishing or exploding gradient problem encountered by traditional RNN when processing long-term sequences. Bidirectional recurrent neural networks (BRNN) can link two hidden layers of opposing directions to the same output (Schuster & Paliwal, 1997). With this architecture, the output layer can simultaneously get information from previous and future states. Although the many-to-one problem in this study cannot make full use of the advantage of the BRNN form, it can still improve the fitting ability of the model by increasing the complexity of the model.

Through experiments, for the dataset of this study, the prediction performance of bidirectional LSTM (Bi-LSTM) is better than that of bidirectional GRU, LSTM, GRU and other models. As mentioned earlier, data sets have sequential relationships in both dimensions. Therefore, after testing and comparison, this paper selects the following model architecture. There are two inputs to the model. One is the time-dependent features, such as how T, Q, V change in each cycle. Another input is the scalar features. As opposed to how we inputted the scalar features in the CNN, scalar features are stacked in order to explore the nuance changes between each cycle. Additionally, time-dependent features are resampled to increase the training speed.

After hyper-parameter tuning, the hyperparameters of 1-cycle, 5-cycle, 10-cycle, 20-cycle RNN-based model are as follow: For the 1-cycle model, 1-layer Bi-LSTM with 512 hidden units is used when dealing with time-dependent inputs. However, the Bi-LSTM is not used when dealing with scalar inputs since it is a vector. For the 5-cycle model, 1-layer Bi-LSTM with 1024 hidden units is used when dealing with time-dependent inputs. Also, a 1-layer Bi-LSTM with 40 hidden units is used when dealing with the scalar

inputs. For the 10-cycle model, 2-layer Bi-LSTM with 256 hidden units is used when dealing with time-dependent inputs. Moreover, a 1-layer Bi-LSTM with 128 hidden units is used when dealing with the scalar inputs. For the 20-cycle model, a 1-layer Bi-LSTM with 512 hidden units is used when dealing with time-dependent inputs. Additionally, a 1-layer Bi-LSTM with 512 hidden units is used when dealing with scalar inputs. Several dense layers with 300 hidden units each are used after Bi-LSTM layers in each model. Lastly, in order to get rid of overfitting, dropout layers are used after each layer.

Figure 4. shows the schematics for the RNN-based model.

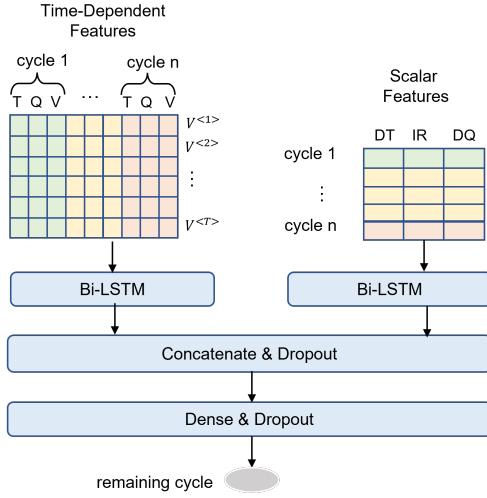


Figure 4. The schematic for the RNN-based model

4.4. Evaluation Metrics

We used the mean absolute percent error (MAPE) to evaluate our model performance:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|$$

A_t represents the actual value while F_t represents the forecast value. A mean absolute percent error reflects on average the accuracy of our models.

Despite that MAPE reflects the overall performance on our prediction accuracy, we would like to know more about how accurate our prediction is for each cell data. In real applications, we might not accept certain levels of error in the prediction. Thus, we define a prediction accuracy with a given acceptable interval. Namely, we report the percentage of predictions where the predictions lie in between the given acceptable interval. For example, given a $\pm 10\%$ tolerance region, we will regard the prediction k as accurate with an actual life remaining t if $0.9t \leq k \leq 1.1t$.

5. Results and Discussions

The results for all four types of models are summarized in Table

Table 1. Results of Models on Different Cycles

Model/ #Cycles	Linear Regression	Random Forest	CNN	RNN
5	456	14.03	23.63	7.5
10	254	16.05	26.5	11.6
20	258	17.9	24.3	11.9

Below, we plotted 3 figures for each model with each figure corresponding with the model's performance on either the 5, 10, or 20 width sliding window. Additionally, each graph contains two bands of different colors, each with a different confidence interval representing two ranges of error that our predictions can fall into. The first band has a range from 0.9-1.1 while the second band has a range from 0.8-1.2. These ranges can be changed depending on the specific needs of the second-use battery user. This allows us to get a better sense of how legitimate a prediction is with respect the true value. In other words, data points that fall out of these band ranges are data points that surpass the expected maximum range value.

5.1. Linear Regression

The MAPE for the linear regression model on 5, 10, and 20 cycles was 456%, 254%, and 258% respectively. In other words, given the nonlinear relationship of the data, attempting to fit the data with a linear model performed poorly. Attached below are the Figures 5-7 for the model performance on 5, 10, and 20 cycles respectively. We plot the predicted remaining cycle life on the y-axis and plot the true remaining cycle life on the x-axis.

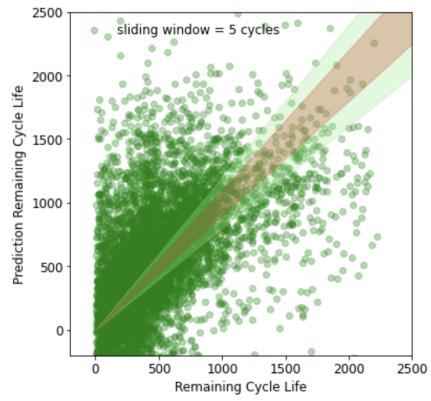


Figure 5. Result of linear regression on 5 cycle

Battery Lifetime Prediction given a Few Cycles

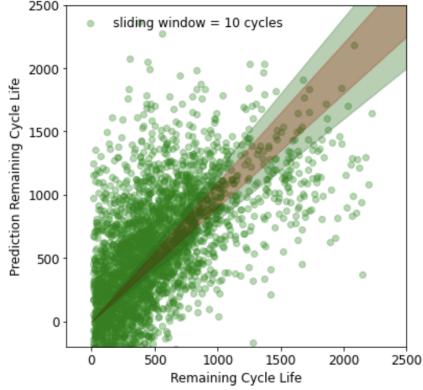


Figure 6. Result of linear regression on 10 cycle

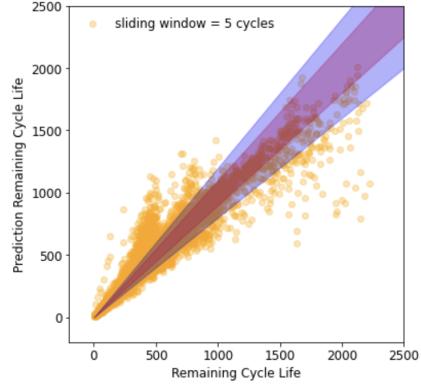


Figure 8. Result of random forest regression on 5 cycle

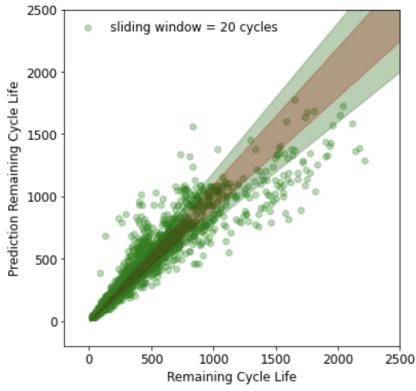


Figure 7. Result of linear regression on 20 cycle

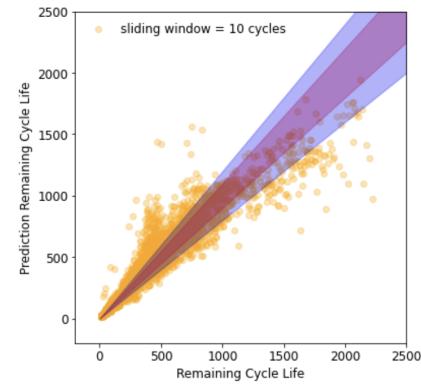


Figure 9. Result of random forest regression on 10 cycle

5.2. Random Forest Regression

The MAPE for the RFR model on 5, 10, and 20 cycles was 14.03%, 16.05%, and 17.9% respectively. Surprisingly, increasing the width of the sliding window from 5 to 20 actually increased the mean absolute percent error. We imagine that this is a result of how increasing the sliding window leads to a decrease in the overall number of samples. Therefore, there is less data for the model to accurately learn representations from at least in the case of the RFR model. Attached below are the Figures 8-10 for the model performance on 5, 10, and 20 cycles respectively. We plot the predicted remaining cycle life on the y-axis and plot the true remaining cycle life on the x-axis.

5.3. Convolutional Neural Network

The MAPE for the convolutional on 5, 10, and 20 cycles was 23.63%, 26.5%, and 24.3% respectively. Attached below are the Figures 11-13 for the model performance on 5, 10, and 20 cycles respectively. We plot the predicted remaining cycle life on the y-axis and plot the true remaining cycle life on the x-axis.

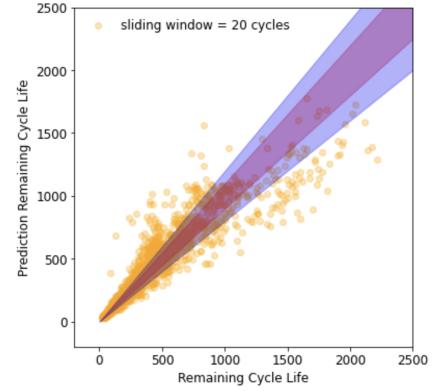


Figure 10. Result of random forest regression on 20 cycle

CNN-based models do not work as well compared to other models experimented regardless of the sliding window size. One possible reason is that the kernel size we use is still limited, with the largest to be (5×5) , therefore only capture the patterns within 5 time steps, which might not be enough for this system.

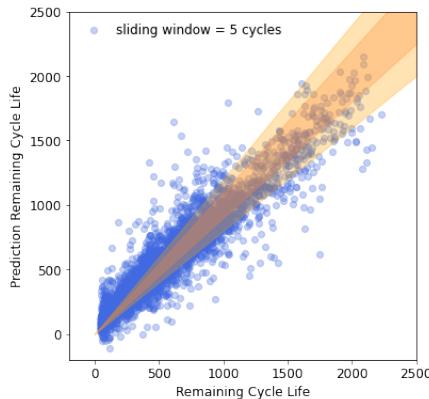


Figure 11. Result of convolutional neural network on 5 cycle

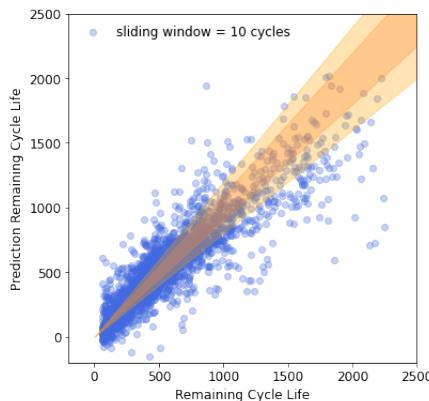


Figure 12. Result of convolutional neural network on 10 cycle

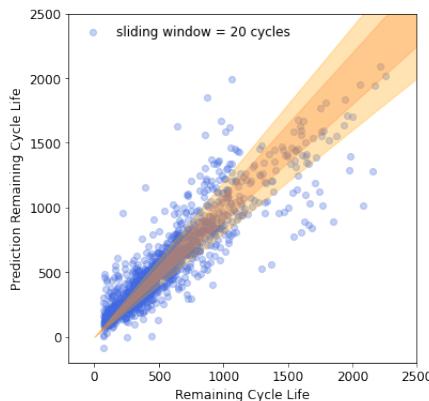


Figure 13. Result of convolutional neural network on 20 cycle

As stated in the method part, we expected to discover patterns across certain features over time within one cycle and over certain consecutive cycles. However, due to the limited kernel size. However, the fact that CNN-based models do not perform well seems to suggest that the underlying pattern across

5.4. Recurrent Neural Network

The mean absolute percent error for the recurrent neural network on 5, 10, and 20 cycles was 7.5%, 11.6%, and 11.9% respectively. Attached below are the Figures 14-16 for the model performance on 5, 10, and 20 cycles respectively. We plot the predicted remaining cycle life on the y-axis and plot the true remaining cycle life on the x-axis.

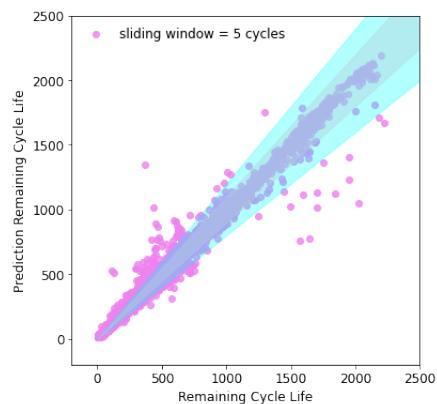


Figure 14. Result of recurrent neural networks on 5 cycle

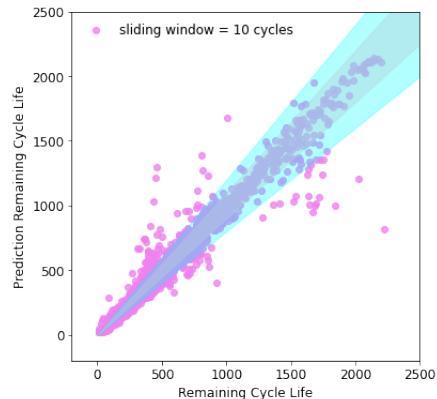


Figure 15. Result of recurrent neural networks on 10 cycle

5.5. One-Cycle Prediction

Ideally, we would like to make predictions based on only one cycle. If successful, such model would be highly desirable for on-board application deployment due to the much reduced computational resources needed. The results from RNN models with 5, 10, and 20 cycles are quite promising. Therefore, we decided to push the limit to see if we could get reasonable accuracy based on just one cycle.

5.6. Confidence Interval

The two tables below highlight the percent of the data that is in the specified band for the 10% range and the 20%

Battery Lifetime Prediction given a Few Cycles

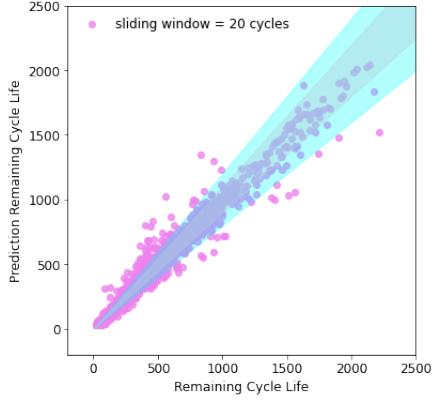


Figure 16. Result of recurrent neural networks on 20 cycle

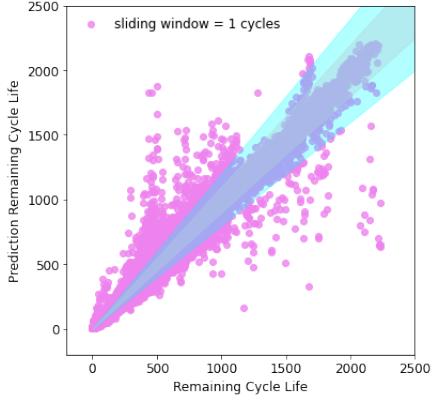


Figure 19. Result of recurrent neural networks on 1 cycle

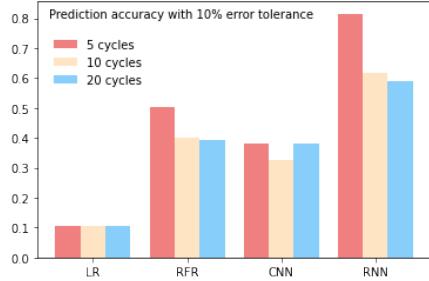


Figure 17. Prediction accuracy with $\pm 10\%$ error tolerance

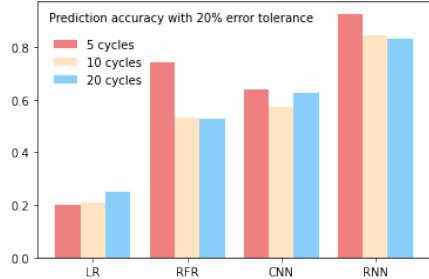


Figure 18. Prediction accuracy with $\pm 20\%$ error tolerance

Table 3. Evaluation with 20% Confidence Interval

#Cycles	Linear Regression	Random Forest	CNN	RNN
5	0.201	0.744	0.638	0.927
10	0.210	0.531	0.573	0.846
20	0.251	0.526	0.626	0.831

6. Conclusion and Future Work

In short, given a specified sliding window of cycles with ranges of 5, 10, or 20, our machine learning algorithms are able to accurately predict the remaining number of cycles without seeing any data from cycles outside that specified sliding window. This paper builds upon already existing research on battery lifetime estimation given the first "n" number of cycles and tweaks the motivation. Given data from one cell, our model performs "data augmentation" in which we can extract multiple sliding windows from one cell thus allowing one cell to now serve as a large amount of data points. This also allowed us to increase the size of our data. We believe that this similar "data augmentation" approach can help battery manufacturers make better prediction on battery lifetime prediction even with a small number of data samples. Our results demonstrated that increasing the size of the sliding window from 5 to 20, does not necessarily increase the mean absolute percent error. We imagine that this is caused by a decrease in the number of overall samples as you increase the sliding window, since extracting non-overlapping windows will decrease the size of the dataset.

Future work could build upon the findings of this paper by using a dataset of the same number of samples even after augmenting the dataset for sliding windows of 5, 10, and 20. This will rule out the possibility of the sample size affecting model performance and instead give better insight on to how increasing the sliding window affects accuracy. Furthermore, future steps could also include decreasing the number of cycles from 5 to an even smaller sliding win-

Table 2. Evaluation with 10% Confidence Interval

#Cycles	Linear Regression	Random Forest	CNN	RNN
5	0.105	0.504	0.382	0.815
10	0.104	0.400	0.326	0.618
20	0.105	0.394	0.382	0.591

range. To clarify, the 0.815 value in the top right box of the first table represents that 81.5% of the data for that specific model is in the 10% error range. In the case of the 81.5% value, that model is the RNN and the width of the sliding window is 5 cycles.

dow range such as 1. Additionally, while this paper found more success using the recurrent neural network approach versus the convolutional neural network, we believe that experimenting with other convolutional neural networks approaches such as the ResNet which utilises skip connections could provide strong results, therefore this creates future areas to potentially explore.

Still our results demonstrated that by utilizing data-driven approaches, battery lifetime estimation can be achieved accurately even with no prior information on a specific cell besides a given sliding window. The implications of this for the used EV space are profound. We envision that for the used car market our machine learning model can be put on a hardware device that will be able to produce the remaining number of cycles given a few charge and discharge cycles. Additionally, companies planning on repurposing used EV batteries for grid scale storage now have better insight on how to assign a warranty to a second-use battery. In conclusion, our research demonstrates that the versatility and efficiency of data-driven approaches in the case of second-use batteries will outperform the alternative significantly.

Software

We used a combination of Jupyter Notebook and Google Colab. All of our models were built and trained locally. The models were implemented in Python and other libraries.

Acknowledgments

Thank you for the CS329P Course Staff for all their assistance and guidance throughout this project. More specifically, thank you to Qingqing Huang, Mu Li, and Alex Smola. Also, thank you to the CS329P Teaching Assistants of Nash Luxsuwong and Alex Boulton.

References

- Berecibar, M. Machine-learning techniques used to accurately predict battery life. *Nature*, 2019.
- Chu, S., Cui, Y., and Liu, N. The path towards sustainable energy. *Nature materials*, 16(1):16–22, 2017.
- Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- Hyötyniemi, H. Turing machines are recurrent neural networks. *Proceedings of step*, 96, 1996.
- Ng, M.-F., Zhao, J., Yan, Q., Conduit, G. J., and Seh, Z. W. Predicting the state of charge and health of batteries using data-driven machine learning. *Nature Machine Intelligence*, 2(3):161–170, 2020.
- Nuhic, A., Terzimehic, T., Soczka-Guth, T., Buchholz, M., and Dietmayer, K. Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. *Journal of power sources*, 239:680–688, 2013.
- Ramadesigan, V., Northrop, P. W., De, S., Santhanagopalan, S., Braatz, R. D., and Subramanian, V. R. Modeling and simulation of lithium-ion batteries from a systems engineering perspective. *Journal of the electrochemical society*, 159(3):R31, 2012.
- Roman, D., S. S. R. V. Machine learning pipeline for battery state-of-health estimation. *Nat Mach Intell*, 2021.
- Schuster, M. and Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.
- Severson, K. e. a. Data-driven prediction of battery cycle life before capacity degradation. *Nat Energy* 4, 2019.
- Tealab, A. Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, 3(2):334–340, 2018.
- Waag, W., Fleischer, C., and Sauer, D. U. Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles. *Journal of Power Sources*, 258:321–339, 2014.