Predicting the degradation of Lithium-Ion Batteries with Linear regression Machine Learning Model.

J. Civile, J. van den Berk, G. Joppe, T. Snoei, S. Uijl, Y. Wierenga

Data Science, Hogeschool Rotterdam, Rotterdam, Nederland

Abstract: The degradation of battery health is a significant concern for many electronic devices, and charging cycles are one of the main factors that contributes to this degradation. The purpose of this study is to investigate the impact of charging cycles on battery health. A sample of 6 out of 177 batteries was used to collect data on the number of charging cycles and the corresponding capacity loss. A prediction model was developed using linear regression to analyse the relationship between charging cycles and battery degradation. The results of the study indicate that there is a significant correlation between the number of charging cycles and capacity loss. The developed prediction model can be used to estimate the capacity loss of a battery after a given number of charging cycles. This research could provide valuable knowledge on the degradation of battery health due to charging cycles and lead to improved methods for preserving battery health.

Keywords: Battery degradation, Machine Learning, linear regression, Lithium-ion Battery.

Table of contents

Introduction

The lithium-ion battery has become a dominant technology since its introduction in 1991 due to its high-energy density and long-life cycle. It is commonly used in a variety of devices, from cell phones to automobiles. However, the efficiency of a lithium-ion battery is dependent on the number of times it is charged, and each charging cycle causes degradation to the battery's health. As the number of charging cycles increases, the battery's capacity decreases, and the performance suffers. This loss of battery health due to charging cycles is a topic of ongoing research in the field of battery science.

Understanding the degradation of battery health caused by charging cycles is a complex task that involves examining the effect on the nominal capacity of the battery that take place during each cycle. This research paper aims to investigate the effects of charging cycles on battery health, and to provide prediction on the speed of degradation.

The hypothesis of this study is that there is a negative correlation between the number of charging cycles and the effective capacity of a battery, resulting in a decline in battery performance. The following questions will be addressed in this study: What are the effects of charging cycles on battery health? How does the number of charging cycles impact the battery's capacity?

This paper will consider these questions in the context of existing literature in the field of battery science and offer insights into the degradation due to charging cycles.

In summary, this research aims to examine the effects of charging cycles on battery health. The study will address questions such as: What are the effects of charging cycles on battery health? How does the number of charging cycles affect the battery's capacity? The results of this research could provide valuable knowledge on the degradation of battery health due to charging cycles and lead to improved methods for preserving battery health.

The study considered a quantitative research method to investigate and predict the degradation of battery health caused by charging cycles. A sample of 6 batteries was selected from 177 batteries. Data was collected using Skoon’s battery monitoring software. The data was analysed using statistical techniques, cleaned and transformed into useful data. The predication model was made using a linear regression. These methods were chosen to provide a comprehensive examination of the topic and data.

Nomenclature

Abbreviation	Definition	Unit
SoC	State of charge	%
DoD	Depth of discharge	%
BMS	Battery Management System	
SOH	State-of-Health	%
EOL	End-of-Life	%
RUL	Remaining useful life	%
NMC	Nickel Manganese Cobalt	
$LiPO_4$	Lithium Polymer	
SVM	Support Vector Machine	
ML	Machine learning	
ANN	Artificial neural networks	
RVM	Relevance Vector Machine	
GPR	Gaussian process regression	
ARMA	Autoregressive–moving-average	
LSTM	Long short-term memory	
MSE	Mean Squared Error	
RMSE	Root Mean Squared Error	
EV	Electric vehicle	
HEV	Hybrid electric vehicle	
E	Energy	kWh
P	Power	kW
T	Time	Hours

Background research

Relevant knowledge sources

There has been a plethora of papers regarding the SOH, RUL, and overall degradation of batteries. Data-driven methods for battery health estimation and prediction are relatively new compared to older methods. The older methods range from simulations of cell behaviour to analogue electrical-circuit models that simulate cell dynamics. The field surrounding these older methods for estimation has been particularly vivid, and therefore several papers on this topic exist [1].

Data-driven methods have gained increasing interest in both academia and industry due to their advantages of flexibility and being model-free, but they require a large set of ageing data as their effectiveness is heavily dependent on the quality and size of the dataset [1]. Older methods using chemistry and/or mechanic-specific models have shown predictive success but developing models that describe full cells cycled under relevant operating conditions remains challenging, given the many degradations models and their couplings to thermal and mechanical heterogeneities within a cell [2]. Advances in computational power and data generation have enabled statistical and machine-learning techniques to accelerate progress for a variety of tasks, which in turn improves the prediction models. These improvements include higher accuracy, earlier prediction, greater interpretability, and broader application to a wide range of cycling conditions [2].

Table 1
An overview of the published literature related to battery SOH estimation and RUL prediction.

Topic	Content	Reference
Health estimation	General review on SOH estimation methods	[3]
	General review on SOH estimation methods and ageing mechanism diagnosis tools	[4]
	Review of SOH estimation techniques for EV and HEV application	[5]
	Battery health estimation and safety management by roughly focusing on physical models, data-driven and fusion methods	[6]
Health prediction	Key functions of the BMS, such as the state of charge estimation, SOH estimation, cell balancing and fault diagnosis	[7]
	Ageing mechanism and SOH estimation for automotive applications	[8]
	Battery state of charge estimation, health monitoring, fault detection, correction, and RUL prediction	[9]
	RUL prediction methods focusing on self-adaptive battery ageing models	[10]
	General review of battery SOH estimation and RUL prediction methods	[11]

Because multiple papers on battery degradation using older and newer data-driven methods exist, there are already semi-established methods and theories. Therefore, the existing information was used to check our findings and confirm that there is a negative correlation between the number of charging cycles and the nominal capacity of a battery, resulting in a decline in battery performance. Most likely no new revelations on the topic of battery degradation will be descoverd, as it has been extensively and professionally researched.

Used knowledge

Current knowledge about batteries and their cycles differs per source, so different sources were looked at and determined what information was best applicable.

A lithium-ion battery works on ion movement between the positive and negative electrodes. In theory such a mechanism should work forever, but cycling, elevated temperature and ageing decrease the performance over time [12]. This degradation in battery capacity and power is inevitable and reduces its performance and service life [13]. Manufacturers take a conservative approach and specify the life of Li-ion batteries in most consumer products between 300 and 500 discharge/charge cycles [12]. Apple estimates the battery life on their newest MacBook products to be around 1000 cycles [14]. As these batteries are all used in consumer-grade electronics, these estimates cannot be applied to the large and industrial batteries used in the dataset. The estimation is that industrial-grade batteries need ten times the number of cycles before any degradation is seen.

How a cycle is calculated also differs per source. Apple was used as the leading source, as they are a market leader in electronics and had sufficient information available about their definition of a cycle. According to Apple, one cycle is completed when an amount that equals 100% has been discharged. But this doesn't necessarily need to happen in one charge [15]. This is better explained in the figure below:

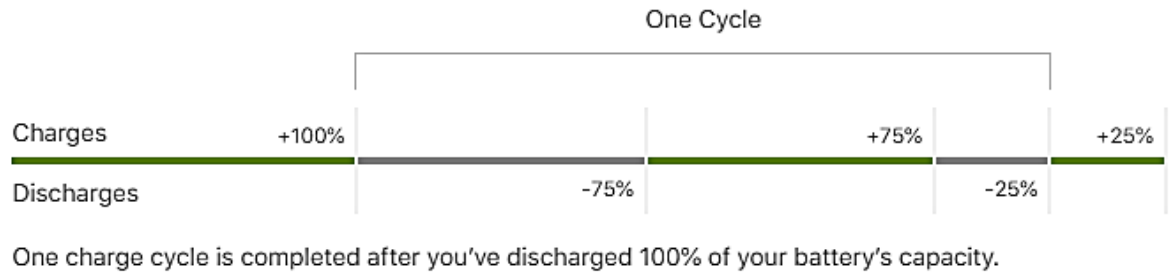

Figure 1. Timeline of one battery cycle. One charge cycle is completed after 100% of the battery's capacity is cumulatively discharged.

The next gap in our knowledge was on the topic of machine learning. First it had to be determined what ML entails in this context and which ML method(s) were of the most use with our data. ML is a method of data analysis that automates analytical model building. It is based on the idea that systems can learn from data, identify patterns, and make decisions or predictions with minimal human intervention [1]. ML approaches are especially attractive for high-rate operating conditions, where first-principles models of degradation are often unavailable [2]. ML algorithms that can be applied to battery degradation modelling are summarized in table 2.

Table 2
Machine learning algorithms for battery degradation modelling.

Methods	Estimation parameter	Content and features	References
ANN	SOH	Uses maximum available capacity to indicate the SOH based neural network.	[16]
	RUL	Uses battery capacity and applies neural networks to estimate the RUL.	[17]
SVM	SOH	Provides battery health capacity model	[18]
	RUL	Provides accurate practical information about the battery's expected life.	[19]
RVM	SOH	Estimates the battery health and remaining capacity using capacity fade trend.	[20]
	RUL	Employed as a time-series prediction model to predict the RUL of the battery.	[21, 22]
Fuzzy Logic	SOH	Establishes the dynamic prediction model and calculates SOH.	[23]
GPR	SOH	Generates the mapping between different factors and forecasts battery SOH.	[24, 25]
	RUL	Realize the RUL prediction by incorporating prognostics with prior information and uncertainty.	[26]
ARMA	SOH	Evaluates SOH by taking battery capacity as the representing parameter.	[27]
	RUL	Uses SOH estimation to characterize RUL.	[28, 29]
LSTM	SOH	Establishes the SOH prediction model-based on a sliding window.	[30]
	RUL	Predicts RUL and EOL with better generalization.	[31]
Bayesian networks	SOH	Devise the battery degradation model over repetitive cycles to estimate SOH.	[32]
	RUL	Evaluates RUL by quantifying the uncertainties in predicting the end-of-life cycles.	[33]
Hybrid methods	SOH	Optimizes the combination of different methods and achieves better performance for accurate SOH estimation.	[34, 35]
	RUL	Predicts the RUL of the battery by combining ML methods with different adaptive techniques.	[36]

Some techniques that stood out to us in the early stages where SVM, ANN and Fuzzy logic. SVM is an effective ML technique which is applied to solve nonlinear issues like battery health estimation, by transforming the data into a higher feature space, where a problem becomes linear [13]. Artificial neural networks (ANN) are particularly suitable for non-linear problems and achieves better accuracy. The structure of ANN consists of artificial neurons arranged in input, output, and hidden layers. Fuzzy logic is a behaviour-based bionic reasoning technique designed to address complicated reasoning problems involving fuzzy phenomena. It is an adaptive ML technique which can be used for SOH estimation [13]. These methods were not suited for our data, as they are either non-linear or overcomplicated. Instead, we turned towards regression. The advantages and disadvantages are shown in table 3.

There are also different types of algorithms used in machine learning; supervised, unsupervised, and reinforced. There is a major difference between supervised and unsupervised learning. In contrast to unsupervised learning, which employs unlabelled datasets, supervised learning uses labelled datasets. By "labelled," we mean that the information has already been marked with the appropriate answer. The supervised learning method in machine learning (ML) makes use of labelled datasets to train algorithms to accurately classify input or predict outcomes. The model evaluates the significance of various features to gradually improve the model fit to the known result using the labelled data. Unsupervised techniques are used to analyse and classify unlabelled datasets without human supervision. These algorithms can uncover hidden patterns in data. As our dataset is labelled, we opted for a supervised learning method.

Table 3.
Advantages and disadvantages of different ML methods

Method	**Advantage**	**Disadvantage**
ANN	Can implement tasks that linear programs cannot Does not require programming Can be executed within any application	Requires training to operate Requires to be emulated Requires high processing time
Fuzzy logic	Simple Easily improved and modifiable	Isn’t always exact Setting accurate guidelines is tough Needs broad testing
SVM	Performs well in high-dimensional space High accuracy Performs well when there is a large gap between classes	Requires long training period Large data sets are not a good fit Poor performance when dataset contains noise
Linear regression	Simple Computationally efficient Easy interpretability of the output	Sometimes overly simplistic Linearity assumption Severely affected by outliers

ML techniques which include regression methods have received increasing attention in Li-ion battery health estimation, because of their model-free characteristics. Generally, the battery health estimation problems are primarily evaluated using the regression techniques as it can be used to supplement existing mathematical approaches with strong potential for battery degradation model development [13]. From the sources we used, we noticed that mostly linear models were used [13, 1]. So that is the method we chose to use as well.

Solution design

This chapter explains how the product was created and what considerations were made. To start exploring this problem the following process was used:

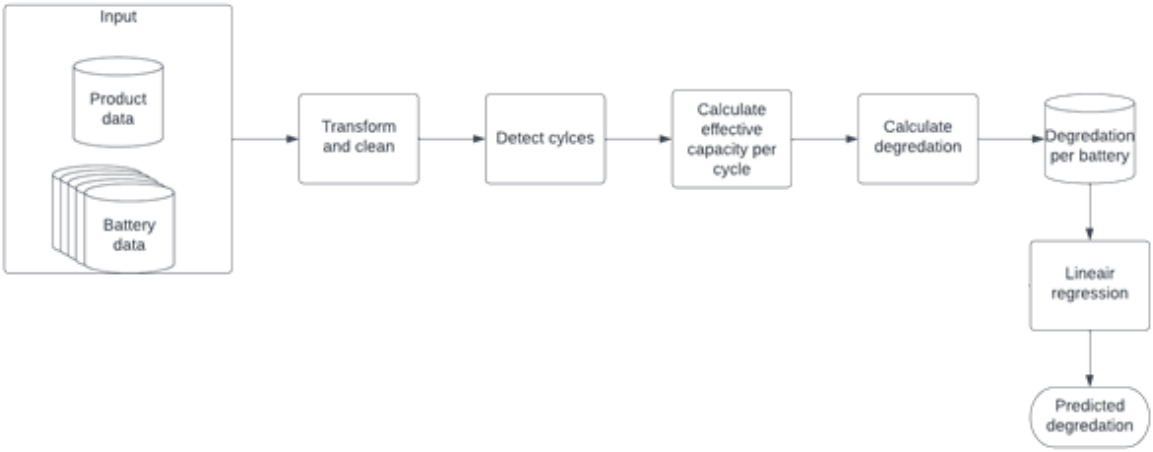

Figure 2. Solution design process

The first step was collecting the data. This data was provided by Skoon and consisted out of product data and battery data. The product data contained static information of the batteries, such as the nominal capacity, type of battery chemistry and maximum continuous discharge. The battery data contained status data such as SoC, Temperature, Power, Current and Voltage. For every battery a CSV file was provided which contained the battery data for every 5 minutes. The next step was to clean and transform the product and battery data. Since the data was sourced from a real company, it also contained sensitive data that had to be anonymized. This was done by using pseudonyms to hide the true identity of the owner and corresponding battery. This code can't be provided due to potential privacy infringement. The next step is to clean the data. Since this is data from a real company, the data is relatively clean. Therefore, the only cleaning that needed to be done was removing the blank values. After cleaning the data all that was left to do was to be combine all the different CSV files into one file [Appendix I].

To gain insight into the lifespan of a battery it is useful to know how much the battery has been degraded as a result of usage. Degradation is the term used when the RUL of the battery decreases due to the irreversible loss of electrolyte in the chemical reaction of the battery. The higher the degradation the lower the RUL will be. A criterion to define the rate of degradation is to plot the degradation percentage against the amount of charging cycles the battery has gone through. You could represent the degradation as function of the effective capacity as:

$$degradation = \frac{Nominal\ capacity - Calculated\ Effective\ Capacity}{Nominal\ Capacity} \quad (1)$$

It was chosen to use the previously discussed Apple standard because it is the most efficient way to calculate a cycle. In short this means that a charging cycle happens when you use all the battery's power. Essentially, this means that by simply counting the number of cycles relative to the maximum number of cycles it can easily be determined what the degradation of a battery is. So first, the data must be converted from status data to cycle data [Appendix II].

After converting the status data to cycle data certain meta-data had to be created for every cycle, such as the amount of energy that went in and out of the battery and the difference in SoC during a cycle [Appendix II]. To calculate the energy that was put in or went out of a system the following equation was used:

$$E = P * \Delta T \tag{2}$$

When the energy of a cycle was calculated, the effective capacity was calculated using the following equation:

$$Effective\ Capacity = \frac{E}{\Delta SoC} * 100\% \tag{3}$$

Based on the Apple standard comparing these cycles with the maximum cycles per battery should determine the degradation of a battery. However, the theory here differs from the problem of Skoon because there are currently no maximum number of cycles defined for the batteries. To determine this, a tool must be created that shows the progression of the degradation over a certain number of cycles.

Based on the background research it was decided to use the linear regression model, because most sources related to battery degradation used regression. This is because the correlation between the cycles and degradation of a battery is scientifically proven. Linear regression is the most used form of regression because it is easy to use, easy to predict, easy to explain and it mostly follows the distribution. of the correlation. A drawback of this method is that it is sensitive to outliers. One source even state that if the battery is new, the degradation could barely be seen. Also, other options of regression were considered but upon examination of the complete discharge pattern of the battery's overall lifespan, the obtained data is sufficiently short to derive that no non-linear correlation can be shown. Therefore, linear regression was chosen to avoid the use of unnecessarily complicated models.

To calculate the degradation of a system, the difference in calculated effective capacity at the beginning of the received data until the last data points was calculated [Appendix III]. Because the calculated effective capacity fluctuates due to inaccuracy of the data, the average capacity over every 10 cycles was used and plotted against the amount of charging cycles.

When the degradation of the batteries was calculated, the correlation with the number of cycles was tested and a linear regression model was made. With this model, it is possible to predict the degradation of a battery when the number of cycles is known. It’s also possible to predict how fast the battery will degrade based on the historic cycles.

A linear regression algorithm necessitates the usage of an x-axis and a y-axis. The number of cycles that a battery has completed serves as the x-axis for this model's application. The y-axis represents the Calculated effective capacity of a battery. The independent variable for this model is the number of cycles the battery has completed. The dependent variable is the

Validation and performance evaluation are carried out using a train-test split. The effectiveness of a machine learning algorithm is assessed using this technique. For this, two subsets of the battery data are generated. The first subset, often known as the training dataset, is used to fit the algorithm's model. The second one, the test dataset, is utilized as a dataset input element and fed into the model. This generates model predictions, which are then compared to the predicted values. The data is split into 80% train data and 20% test data.

The linear regression model is expected to show a decrease in overall capacity when the amount of charging cycle increases. The accuracy of a regression model is tested through residuals. There are multiple ways to implement residual calculation as a measure of accuracy. For a quick representation of the accuracy of the regression model, MSE was used. That is the average squared difference between the estimated values and the actual value. The result of MSE is always a positive value and values closer to zero indicate a better fit. For a more informative measure of accuracy, R^2 will be used to determine the fit of the model. Both RMSE and R^2 Quantify how well a regression model fits the dataset. Since the result of the R^2 is conveniently scaled between 0 and 1 it is preferred over the absolute response value of the RMSE.

Results / testing and evaluation

After we implemented the solution design and tested it, we discovered that the data points are very spread out and that there is little to no correlation between the different batteries. This is shown in figure 3. We checked the calculations and the output of the individual batteries. We discovered that out of 177 batteries, only 50 have a significant number of cycles. This means the other 127 batteries in the dataset had seen almost no use. While we see that 50 batteries have "significant" cycles, this number of cycles ranged only between 300-600.

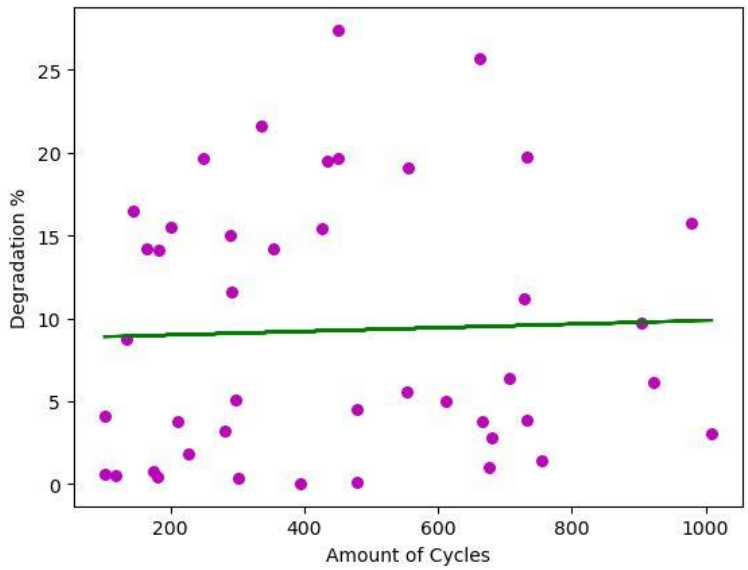

Figure 3. Degradation % & number of cycles over all batteries

The batteries also differ in age, size and charging behaviour, which means that comparing these batteries to each other is not really viable. We do, however, see that individual batteries show a correlation between the number of cycles and the degradation of the battery.

When looking back at the background research, we can see that for consumer-grade products, manufacturers conservatively estimate the cycle count before degradation of batteries to be between 300-500 cycles [12]. Apple states that the batteries in their MacBook products last for 1000 cycles before degradation [14].

Comparing this to the 50 best batteries that have 300-600 cycles, one would say that degradation should be more visible. An important point, however, is that the batteries in the dataset are not consumer-grade but meant for industrial purposes. We estimated that, based on differences in battery capacity and power output between consumer and industrial-grade batteries, that the latter needs ten times as many cycles before degradation occurs.

Also, accurate early life cycle prediction is challenging because of the negligible capacity degradation in early cycles, as well as the relatively small datasets used to date that span a limited range of lifetimes [2]. For example, Harris et al. found a weak correlation (ρ=0.1) between capacity values at cycle 80 and capacity values at cycle 500 for a consumer-grade battery, illustrating the difficulty of early life cycle prediction [37].

Seeing as the dataset mostly consists of newer batteries over the course of a short period, in combination with the previously mentioned points like, difference in age, capacity and charging behavior, we have decided to divert the focus from comparing the multiple batteries, to looking at the degradation and cycle count correlation of individual batteries. Six batteries that contained the most data points were selected for this.

After we implemented this focus on individual batteries, we saw a big fluctuation in capacity degradation. This can be seen clearly in figure 4. This is most likely caused by the data being recorded in 5-minute intervals. To counteract the fluctuation, we calculated the average capacity degradation over the course of 10 cycles and assigned that value to the last cycle ID of those 10. This means that every group of 10 cycles has an average degradation value assigned to it.

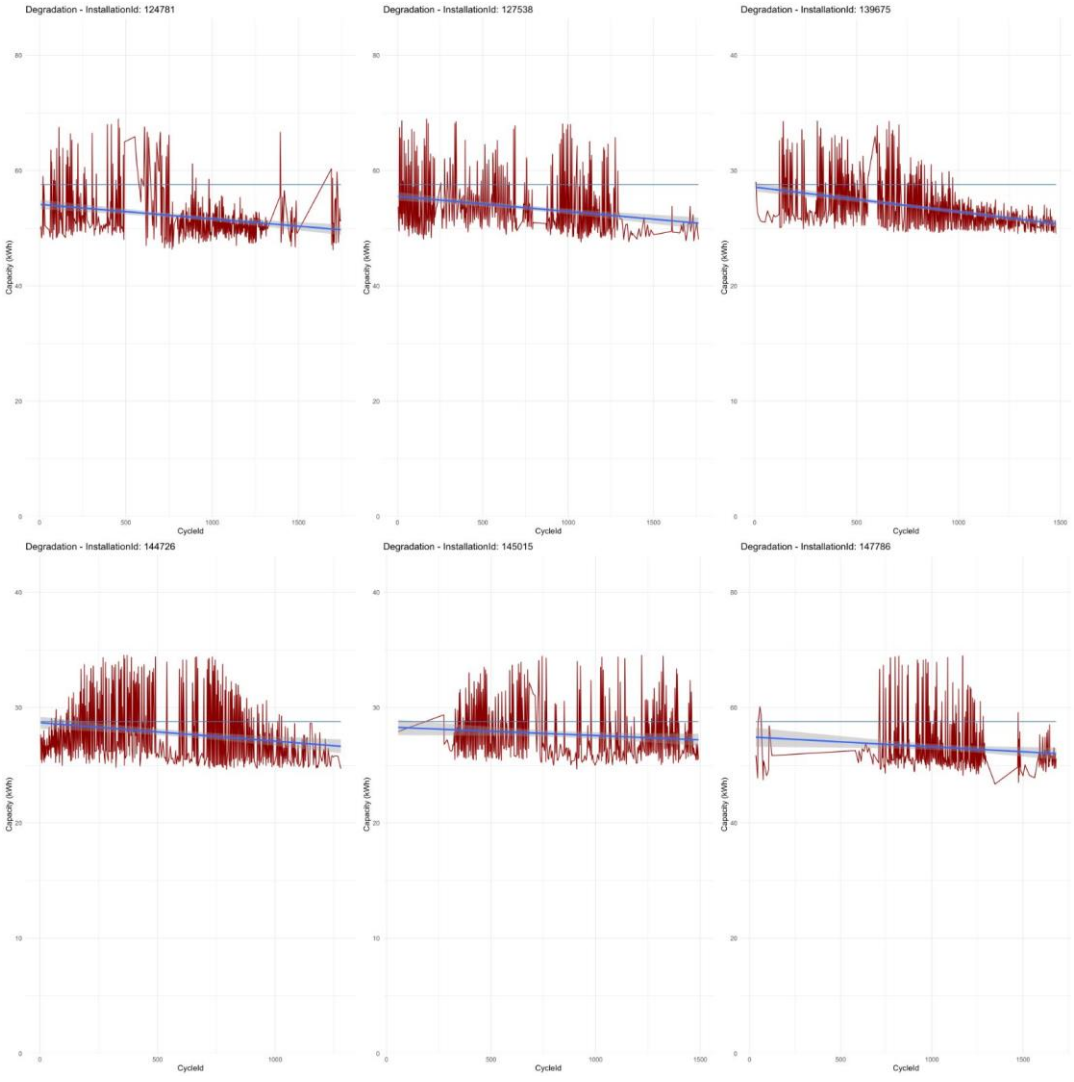

Figure 4. Large amounts of fluctuation in the degradation of the batteries

After we calculated the average degradation over 10 cycles, we plotted this new variable against the number of cycles and implemented the 80/20 of the train-test split. This resulted in better graphs seen in figure 5. We tested the accuracy of these models using R^2 and MSE.

The R^2 ranges from 0.23 to 0.56, with the latter showing that there is indeed a correlation between the degradation and the number of cycles. The MSE is low on all six graphs, ranging from 1.6 to 33.5, which also indicates that there is a correlation between the two variables. The lower values can be explained by the fact that early lifecycle prediction is challenging because of the limited degradation in early cycles and the limited range of lifetimes [2].

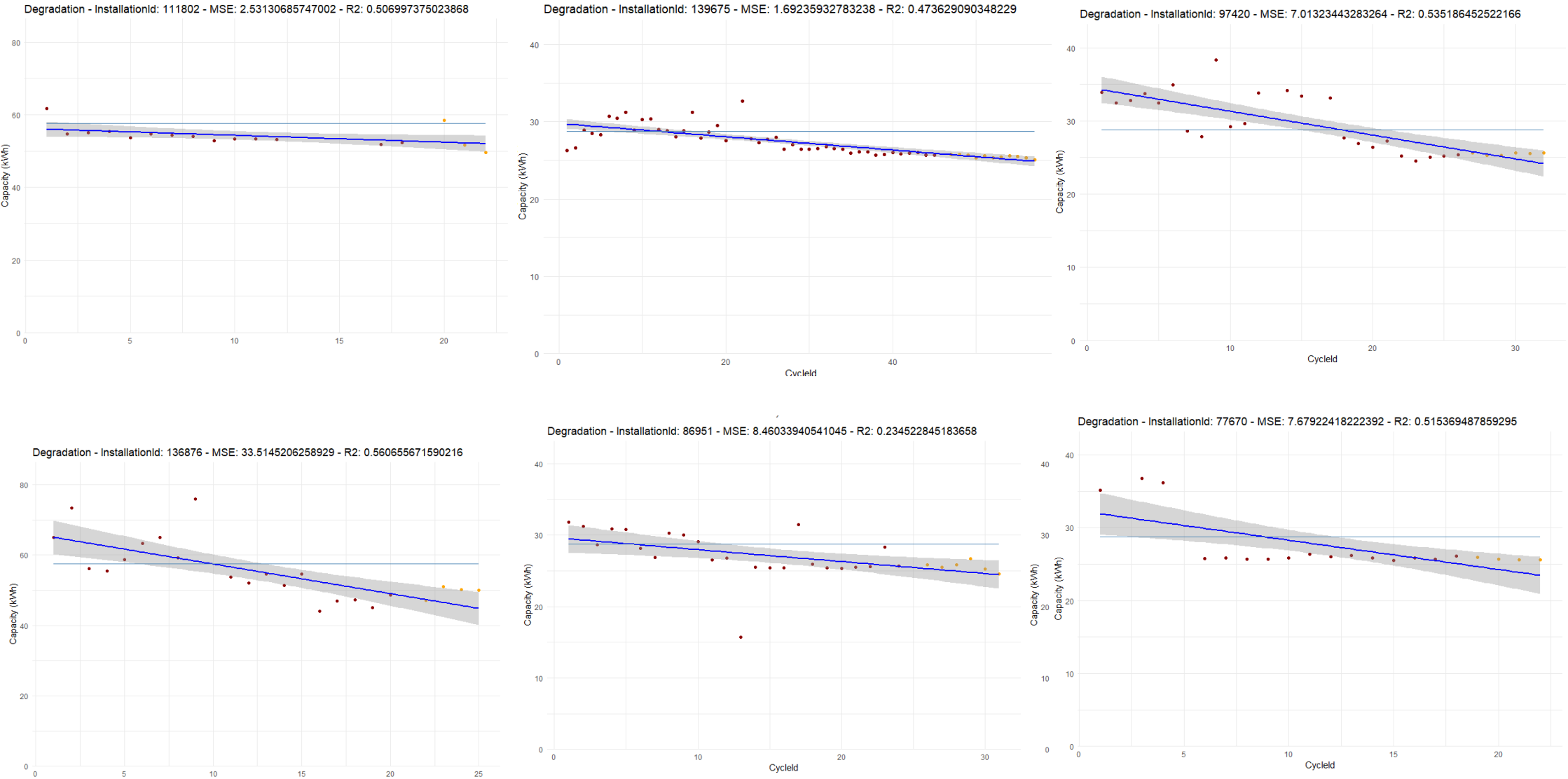

Figure 5. Capacity Model prediction based on Battery cycles

Conclusion

In conclusion, this study aimed to investigate the impact of charging cycles on the degradation of battery capacity, which gives an overview of the SOH of the battery. A sample set of 177 batteries was used to collect datapoints every 5 minutes. 6 batteries with the most datapoints were used for testing the prediction model. In order to be tested, the number of charging cycles were calculated, and the effective capacity was determined using equations 2 and 3.

Based on the pros and cons listed in table 3 the prediction model is based on a linear regression, due to the easy implementation and linear degradation characteristic of batteries in their early life cycles. For training and testing the model an 80-20 split was used. This means 80% training data with a 20% test data.

Figure 5 gives an overview of the prediction model based on the training data. The accuracy of the regression model is tested by using MSE and R^2. As the results shown in Figure 5 that the MSE were calculated with a max of 33 but an average of 5.4 when we ignore the max and the calculated R^2 values ranges from 0.23 to 0.56. Considering the MSE at first glance the model shows a decent fit, hence the MSE values are mostly close to zero. Looking at the R^2 values it could be stated that the predicted outcome does not fit the data that well. As stated by Harris et al, early life cycle detection and capacity loss is difficult to predict due to a small data set because of a limited range of lifetime and negligible capacity loss in the short lifetime.

It’s important to note that while the linear regression model used in this study provides a useful tool for predicting capacity loss based on charging cycles, it is also important to consider other factors that may affect battery health such as temperature and storage conditions.

Additionally, this study had a limited sample size, and further research is needed to validate the findings and to evaluate the effectiveness of the developed prediction model on a larger scale.

Reference List

[1] Y. Li et al, “Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review,” *Renewable and Sustainable Energy Reviews,* vol. 112, pp. 109-254, 2019.

[2] K. Serverson et al, “Data-driven prediction of battery cycle life before capacity degradation,” *Nature Energy,* vol. 4, no. 5, pp. 383-391, 2019.

[3] R. Xiong, L. Li and J. Tian, “Towards a smarter battery management system: A critical review on battery state of health monitoring methods,” *Journal of Power Sources,* pp. 18-29, 2018.

[4] M. Berecibar et al, “Critical review of state of health estimation methods of Li-ion batteries for real applications,” *Renewable and Sustainable Energy Reviews,* pp. 572-587, 2016.

[5] A. Farmann et al, “Critical review of on-board capacity estimation techniques for lithium-ion batteries in electric and hybrid electric vehicles,” *Journal of Power Sources,* vol. 281, pp. 114-130, May 2015.

[6] S. Rezvanizaniani, Z. Liu, Y. Chen and J. Lee, “Review and recent advances in battery health monitoring and prognostics technologies for electric vehicle (EV) safety and mobility,” *Journal of Power Sources,* vol. 256, pp. 110-124, 2014.

[7] L. Lu, X. Han, J. Li, J. Hua and M. Ouyang, “A review on the key issues for lithium-ion battery management in electric vehicles,” *ournal of Power Sources,* vol. 226, pp. 272-288, 2013.

[8] A. Barré et al, “A review on lithium-ion battery ageing mechanisms and estimations for automotive applications,” *Journal of Power Sources,* vol. 241, pp. 680-689, November 2013.

[9] J. Lee and J. Zhang, “A review on prognostics and health monitoring of Li-ion battery,” *Journal of Power Sources,* vol. 196, no. 15, pp. 6007-6014, 2011.

[10] M. Lucu, E. Martinez-Laserna, I. Gandiaga and H. Camblong, “A critical review on self-adaptive Li-ion battery ageing models,” *Journal of Power Sources,* vol. 401, pp. 85-101, 2018.

[11] M. S. H. Lipu et al, “A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles: Challenges and recommendations,” *Journal of Cleaner Production,* vol. 205, pp. 115-133, 2018.

[12] “BU-808: How to Prolong Lithium-based Batteries,” Battery Universuty, 3 11 2021. [Online]. Available: https://batteryuniversity.com/article/bu-808-how-to-prolong-lithium-based-batteries. [Accessed 5 1 2022].

[13] H. Rauf, M. Khalid and N. Arshad, “Machine learning in state of health and remaining useful life estimation: theoratical and technological development in battery degradation modelling,” *Renewable and Sustainable Energy Reviews,* vol. 156, 2022.

[14] Apple, “Determine battery cycle count for Mac notebooks,” 06 Augustus 2019. [Online]. Available: https://support.apple.com/en-us/HT201585.

[15] Apple, “Why Lithium-ion?,” Apple, 2019. [Online]. Available: https://www.apple.com/batteries/why-lithium-ion/. [Accessed 1 12 2022].

[16] G. You, S. Park and D. Oh, “Real-time state-of-health estimation for electric vehicle batteries: A data-driven approach,” *Applied Energy,* vol. 176, pp. 92-103, 2016.

[17] H. Pan, Z. Lü, H. Wang, H. Wei and L. Chen, “Novel battery state-of-health online estimation method using multiple health indicators and an extreme learning machine,” *Energy,* vol. 160, pp. 466-477, 2018.

[18] H. Dong, X. Jin, Y. Lou and C. Wan, “Lithium-ion battery state of health monitoring and remaining useful life prediction based on support vector regression-particle filter,” *Journal of Power Sources,* vol. 271, pp. 114-123, 2015.

[19] T. Qin, S. Zeng and J. Guo, “Robust prognostics for state of health estimation of lithium-ion batteries based on an improved PSO–SVR model,” Microelectronics Reliability,” *Microelectronics Reliability,* vol. 55, pp. 1280-1284, 2015.

[20] J. Lee, D. Kwon and M. Pecht, “Reduction of Li-ion Battery Qualification Time Based on Prognostics and Health Management,” *IEEE Transactions on Industrial Electronics,* p. 1, 2018.

[21] H. Li, D. Pan and C. L. P. Chen, “Intelligent Prognostics for Battery Health Monitoring Using the Mean Entropy and Relevance Vector Machine,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 44, no. 7, pp. 851-862, 2014.

[22] D. Liu, J. Zhou, D. Pan, Y. Peng and X. Peng, “Lithium-ion battery remaining useful life estimation with an optimized Relevance Vector Machine algorithm with incremental learning,” *Measurement,* vol. 63, pp. 143-151, 2015.

[23] H. Tian, P. Qin, K. Li and Z. Zhao, “A review of the state of health for lithium-ion batteries: Research status and suggestions,” *Journal of Cleaner Production,* vol. 261, pp. 120-813, 2020.

[24] D. Liu, J. Pang, J. Zhou, Y. Peng and M. Pecht, “Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression,” *Microelectronics Reliability,* vol. 53, no. 6, pp. 832-839, 2013.

[25] R. R. Richardson, M. A. Osborne and D. A. Howey, “Gaussian process regression for forecasting battery state of health,” *Journal of Power Sources,* vol. 357, pp. 209-219, 2017.

[26] Y. Peng, Y. Hou, Y. Song, J. Pang and D. Liu, “Lithium-Ion Battery Prognostics with Hybrid Gaussian Process Function Regression,” *Energies,* vol. 11, no. 6, p. 1420, 2016.

[27] Z. Yun, W. Qin, W. Shi and P. Ping, “State-of-Health Prediction for Lithium-Ion Batteries Based on a Novel Hybrid Approach,” *Energies,* vol. 13, no. 18, p. 4858, 2020.

[28] B. Long, W. Xian, L. Jiang and Z. Liu, “An improved autoregressive model by particle swarm optimization for prognostics of lithium-ion batteries,” *Microelectronics Reliability,* vol. 53, no. 6, pp. 821-831, 2013.

[29] Y. Zhou and M. Huang, “Lithium-ion batteries remaining useful life prediction based on a mixture of empirical mode decomposition and ARIMA model,” *Microelectronics Reliability,* vol. 65, pp. 265-273, 2016.

[30] J. Qu, F. Liu, Y. Ma and J. Fan, “A Neural-Network-Based Method for RUL Prediction and SOH Monitoring of Lithium-Ion Battery,” *IEEE Access,* vol. 7, pp. 178 - 191, 2019.

[31] K. Park, Y. Choi, W. J. Choi, H. Y. Ryu and H. Kim, “LSTM-Based Battery Remaining Useful Life Prediction With Multi-Channel Charging Profiles,” *IEEE Access,* vol. 8, pp. 20786 - 20798, 2020.

[32] Z. He, M. Gao, G. Ma, Y. Liu and S. Chen, “Online state-of-health estimation of lithium-ion batteries using Dynamic Bayesian Networks,” *Journal of Power Sources,* vol. 267, pp. 576 - 583, 2014.

[33] S. S. Y. Ng, Y. Xing and K. L. Tsui, “A naive Bayes model for robust remaining useful life prediction of lithium-ion battery,” *Applied Energy,* vol. 118, pp. 114-123, 2014.

[34] F. Li and J. Xu, “A new prognostics method for state of health estimation of lithium-ion batteries based on a mixture of Gaussian process models and particle filter,” *Microelectronics Reliability,* vol. 55, no. 7, pp. 1035-1045, 2015.

[35] J. Yu, “State-of-Health Monitoring and Prediction of Lithium-Ion Battery Using Probabilistic Indication and State-Space Model,” *IEEE Transactions on Instrumentation and Measurement,* vol. 64, no. 11, pp. 2937-2949, 2015.

[36] Y. Xing, E. W. M. Ma, K. L. Tsui and M. Pecht, “An ensemble model for predicting the remaining useful performance of lithium-ion batteries,” *Microelectronics Reliability,* vol. 53, no. 6, pp. 811-820, 2013.

[37] S. J. Harris, D. J. Harris and C. Li., “Failure statistics for commercial lithium ion batteries: A study of 24 pouch cells,” *Journal of Power Sources,* vol. 342, pp. 589-597, 2017.

Appendix

Appendix I: ImportProductData.R

```
library(dplyr)
library(tidyverse)

productTypes = read.csv(file.path(basePath, "221216_productTypes.csv"),
                        header = TRUE,
                        colClasses=c("ID"="character"))
pt = productTypes %>% filter(InstallationId != "NULL")

# These properties get imported. All the others get discarded.
useFullBatteryProperties = c("TimestampUTC",
                             "Power",
                             "Current",
                             "Voltage",
                             "StateOfCharge",
                             "Temperature",
                             "TotalPowerIn",
                             "TotalPowerOut")
datePath = "221130"
combinedProductData = list()
for (i in 1:nrow(pt)) {
  productType = pt[i, ]
  batteryPath = file.path(basePath, paste(
    paste(datePath, productType$ID, sep = "_"),
    ".csv",
    sep = ""
  ))
  if (file.exists(batteryPath)) {
    batteryData = read.csv(batteryPath)
    useFullBatteryData = batteryData %>% select(all_of(useFullBatteryProperties))
    useFullBatteryData$TimestampUTC = as.POSIXct(useFullBatteryData$TimestampUTC,
                                         format = "%Y-%m-%d %H:%M:%S",
                                         tz = "UTC")
    combinedData = list(productType, useFullBatteryData)
    names(combinedData) = c("ProductType", "BatteryData")
    combinedProductData[[i]] = combinedData
  } else {
    print(paste(batteryPath, "Does not exist"))
  }
}
combinedProductData = compact(combinedProductData)
```

Appendix II: DetectCyles.R

```
library(data.table)

# Loop through entire combinedProductData and extract all batteryData.
list_batteryData = lapply(combinedProductData, function(x) {
  data.frame(
    InstallationId = x[[1]]$InstallationId,
    NominalCapacity = x[[1]]$NominalCapacity,
    EffectiveCapacity = x[[1]]$EffectiveCapacity,
    BatteryType = x[[1]]$BatteryType,
    Power = as.numeric(x[[2]]$Power),
    PowerIn = as.numeric(x[[2]]$TotalPowerIn),
    PowerOut = as.numeric(x[[2]]$TotalPowerOut),
    Current = as.numeric(x[[2]]$Current),
    Voltage = as.numeric(x[[2]]$Voltage),
    TimeStamp = x[[2]]$TimestampUTC,
    StateOfCharge = as.numeric(x[[2]]$StateOfCharge)
  )
})

list_diffTime = lapply(list_batteryData, function(x){
  df = data.frame(x) %>%
    arrange(TimeStamp) %>%
    mutate(PowerIn = coalesce(PowerIn, 0)) %>%
    mutate(PowerOut = coalesce(PowerOut, 0)) %>%
    distinct() %>%
    mutate(Status = case_when(
      PowerIn > PowerOut ~ "Charging",
      PowerIn == PowerOut ~ "Idle",
      PowerIn < PowerOut ~ "Discharging")) %>%
    mutate(TimeStamp_Diff_Minutes =
             as.numeric(difftime(TimeStamp, lag(TimeStamp, default = first(TimeStamp)), units = "mins"))) %>%
    mutate(HasBeenOffline = (
      TimeStamp_Diff_Minutes >= 12 # If there are more than 11 minutes between datapoints, we consider
the battery offline.
    )) %>%
    mutate(CycleId = cumsum(Status != lag(Status, default = first(Status))) + 1) %>%
    mutate(Diff_Power = Power - (PowerIn - PowerOut)) %>%
    group_by(CycleId) %>%
    add_tally(name = "nr_datapoints") %>%
    mutate(PeakPower = max(abs(Power))) %>%
    mutate(MeanPower = mean(Power)) %>%
    ungroup()
})

diffTime_df = do.call("rbind", list_diffTime)

# Loop through entire list_batteryData. list_cycles will be a list of the cycles by battery.
list_cycles = lapply(list_batteryData, function(x) {
  # Orders batteryData by timeStamp and deletes duplicates.
  # Adds Status column to batteryData.
```

```
  df = data.frame(x) %>%
    arrange(TimeStamp) %>%
    mutate(PowerIn = coalesce(PowerIn, 0)) %>%
    mutate(PowerOut = coalesce(PowerOut, 0)) %>%
    distinct() %>%
    mutate(Status = case_when(
      PowerIn > PowerOut ~ "Charging",
      PowerIn == PowerOut ~ "Idle",
      PowerIn < PowerOut ~ "Discharging")) %>%
    mutate(TimeStamp_Diff_Minutes = as.numeric(difftime(TimeStamp, lag(TimeStamp, default =
first(TimeStamp)), units = "mins"))) %>%
    mutate(HasBeenOffline = (
      TimeStamp_Diff_Minutes >= 12 # If there are more than 11 minutes between datapoints, we consider
the battery offline.
    )) %>%
    mutate(CycleId = cumsum(Status != lag(Status, default = first(Status))) + 1) %>%
    group_by(CycleId) %>%
    add_tally(name = "nr_datapoints") %>%
    mutate(TotalPower = sum(case_when(
      TimeStamp_Diff_Minutes == 0 ~ (0.1 * Power),
      TimeStamp_Diff_Minutes < 12 ~ Power * TimeStamp_Diff_Minutes / 60,
      TimeStamp_Diff_Minutes >= 12 ~ 0))) %>%
    mutate(EnergyIn = sum(case_when(
      TimeStamp_Diff_Minutes == 0 ~ (0.1 * PowerIn),
      TimeStamp_Diff_Minutes < 12 ~ PowerIn * TimeStamp_Diff_Minutes / 60,
      TimeStamp_Diff_Minutes >= 12 ~ 0))) %>%
    mutate(EnergyOut = sum(case_when(
      TimeStamp_Diff_Minutes == 0 ~ (0.1 * PowerOut),
      TimeStamp_Diff_Minutes < 12 ~ PowerOut * TimeStamp_Diff_Minutes / 60,
      TimeStamp_Diff_Minutes >= 12 ~ 0))) %>%
    mutate(PeakPowerIn = max(abs(PowerIn))) %>%
    mutate(PeakPowerOut = max(abs(PowerOut))) %>%
    mutate(MeanPowerIn = mean(PowerIn)) %>%
    mutate(MeanPowerOut = mean(PowerOut)) %>%
    ungroup()

  # Detect when battery has been offline.
  offlineData = data.frame(rbind(df[which(df$HasBeenOffline), ],
                                 lead(df[which(df$HasBeenOffline), ], default = df %>% slice(n())))) %>%
    mutate(Status = "Offline") %>%
    mutate(nr_datapoints = 2)

  # Checks which data point has a different Status than the next one. Save this one.
  differs = df[which(df$Status != lead(df$Status, default = last(df$Status))), ]

  # Combine offline and different status data.
  differs_with_offlineData =
    rbind(differs, offlineData) %>%
    arrange(TimeStamp)

  # Calculate the duration of a cycle by substracting the current and previous timeStamp.
  # In this line you can also very easily add new properties of the cycle.
```

```
  cycles = differs_with_offlineData %>%
    mutate(Duration =
             difftime(
               differs_with_offlineData$TimeStamp,
               lag(
                 differs_with_offlineData$TimeStamp,
                 default = first(df$TimeStamp)
               ),
               units = "hours"
             )) %>%
    mutate(
      SoC_Diff =
        differs_with_offlineData$StateOfCharge - lag(
          differs_with_offlineData$StateOfCharge,
          default = first(df$StateOfCharge)
        )
    ) %>%
    drop_na() %>%
    filter(nr_datapoints != 1)
})

# Create data frame of all the cycles
list_cycles_exp = do.call("rbind", list_cycles)

selection_cycle_data = list_cycles_exp %>% select(
  CycleId,
  InstallationId,
  NominalCapacity,
  TotalPower,
  EnergyIn,
  EnergyOut,
  SoC_Diff,
  StateOfCharge,
  MeanPowerIn,
  MeanPowerOut,
  PeakPowerIn,
  PeakPowerOut,
  Duration,
  Status,
  TimeStamp,
)
```

Appendix III: CalculateDegredation.R

```
discharging_degredation_df = selection_cycle_data %>%
  filter(Status == "Discharging") %>%
  mutate(TotalEnergy = EnergyIn - EnergyOut) %>%
  mutate(Energy_Out_NC = EnergyOut / NominalCapacity * 100) %>%
  mutate(Calc_EffectiveCapacity = TotalEnergy / SoC_Diff * 100 ) %>%
  mutate(Calc_Internal_Resistance = TotalEnergy - TotalPower) %>%
  mutate(DegradationPerc = (NominalCapacity - Calc_EffectiveCapacity) / NominalCapacity * 100) %>%
  group_by(InstallationId) %>%
  filter(Energy_Out_NC >= 50) %>%
  mutate(Index = row_number(InstallationId)) %>%
  add_tally(name = "Amount_Discharging_Cycles") %>%
  filter(Amount_Discharging_Cycles > 200) %>%
  ungroup()

grouped_by_installationId = discharging_degredation_df %>%
  group_by(InstallationId = InstallationId) %>%
  group_map(~.x, keep=TRUE)

grouped_averages = lapply(grouped_by_installationId, function(x){
  df = data.frame(x) %>%
    group_by(avg_index = Index %/% 10 + 1) %>%
    mutate(avg_calc_cap = mean(Calc_EffectiveCapacity)) %>%
    slice(which(Index %% 10 == 1))
})

models = lapply(grouped_averages, function(x) {
  df = data.frame(x)
  model = lm(avg_calc_cap ~ CycleId, data = df)
  rsq = summary(model)$r.squared
  slope = model$coef[2]
  mse = mean((df$avg_calc_cap-predict(model))^2)
  installationId = first(df$InstallationId)
  new_df = data.frame(list(installationId, rsq, slope, mse))
  names(new_df) = c("InstallationId", "R2", "Slope", "MSE")
  new_df
})
valid_batteries = do.call("rbind", models) %>%
  filter(Slope < 0)

ungrouped_averages = do.call("rbind", grouped_averages) %>%
  filter(InstallationId %in% valid_batteries$InstallationId) %>%
  filter(avg_calc_cap > 0 & avg_calc_cap < (NominalCapacity * 1.5))

regrouped_averages = ungrouped_averages %>% group_by(InstallationId) %>% group_map(~.x,
keep=TRUE)

theme_set(theme_minimal())
list_avg_plots = lapply(regrouped_averages, function(x){
  df = data.frame(x) %>%
    select(
```

```
    NominalCapacity,
    avg_calc_cap,
    avg_index,
    InstallationId
  )
 drop_df = data.frame(slice_tail(df, n =as.integer(nrow(df) * 0.9)))
 training_df = data.frame(slice_head(drop_df, n =as.integer(nrow(drop_df) * 0.8)))
 test_df = slice_tail(drop_df, n = as.integer(nrow(drop_df) * 0.2))
 installationId = first(training_df$InstallationId)
 model = lm(training_df$avg_calc_cap ~ training_df$avg_index, data = training_df)
 rsq = summary(model)$r.squared
 mse = mean((training_df$avg_calc_cap-predict(model))^2)
 slope = model$coef[2]
 y_limit = first(training_df$NominalCapacity) * 1.5
 title = paste("Degradation", paste("InstallationId", installationId, sep = ": "), paste("MSE", mse, sep = ": "),
paste("R2", rsq, sep = ": "), sep=" - ")
 plot = ggplot(NULL) +
  geom_point(data = training_df, aes(x = avg_index, y = avg_calc_cap), color = "darkred") +
  geom_point(data = test_df, aes(x = avg_index, y = avg_calc_cap), color = "orange") +
  geom_smooth(data = drop_df, method = lm, aes(avg_index, avg_calc_cap), color = "blue") +
  geom_line(data = drop_df, aes(x = avg_index, y = NominalCapacity), color = "steelblue") +
  labs(title = title, x = "CycleId", y = "Capacity (kWh)") +
  scale_y_continuous(expand = c(0, 0), limits = c(0, y_limit))
})

do.call("grid.arrange", c(list_avg_plots, ncol = 3))
```