CS230

Predicting the aging trajectory of Li-ion batteries under different charging protocols by neural networks models

CS 230 Project

Lude Rong[1], Jihan Zhuang[2], and Benson Zu[3]

[1] ***M.S. Dept.of Civil & Environmental Engineering (luderong@stanford.edu)***
[2] ***Ph.D. Dept.of Material Science & Engineering (jihan123@stanford.edu)***
[3] ***M.S. Dept.of Civil & Environmental Engineering (zuyeyang@stanford.edu)***

Abstract

As demand for Lithium-ion batteries grows rapidly in commercial applications such as electric vehicles and energy storage systems, it is increasingly important to efficiently and accurately predict the useful lifespan of Li-ion batteries. Existing battery degradation prediction models tend to be limited by their costly input requirement or sample size. We utilize a neural network (NN) model, with a weighted loss function, to forecast battery degradation curves. Given initial battery configurations and cycling conditions, our multi-output NN model yields regression coefficients that define battery degradation trajectories. Trained with approximately 100 charging protocols, the NN model achieved a $R^2 = 0.91$, significantly outperforming the baseline model.

1 Introduction

Li-ion batteries are widely used in both electric vehicles and energy storage systems, and it is vital to precisely project their lifetime performance to maximize utilization.[2]. Current research models are designed to predict batteries' Remaining Useful Life (RUL) to a given capacity. Because each prediction requires experimental data of 5 to 100 initial cell cycles as inputs, predicting the whole degradation trajectory of batteries in those models are complex and inefficient [1][10][4].

To reduce operational complexity and better visualize battery lifetime performance, we build a Neural Network (NN) model to forecast battery degradation curves without running any testing experiments. More specifically, our NN of Multi-Output Regression model calculates the coefficients of empirical equations that approximate the battery aging curves. Those coefficients are then used to plot battery degradation trajectories[7]. The project goal is to accurately forecast cell aging trajectories using only initial battery configurations (cell capacity and resistance) and cycling conditions (voltage, current, and charge/discharge time). We deploy and compare the different NN layer and neuron settings. Trained with approximately 100 combinations of battery configurations and cycling conditions, the NN model is able to predict battery degradation curves for new parameter combinations.

This approach sheds light on battery degradation prediction because: 1) it is new and holistic – to our knowledge, no published methods predict the whole aging trajectory curve; 2) it is simple – in operation, it predicts the aging trajectory of a new cell with minimal experimental measurements; 3) it is more economic comparing to traditional RUL predictions, which depend on costly cell-specific experiments.

2 Related Work

In recent years, machine learning (ML) models are garnering increasing interest in both academia and industry due to their high flexibility and efficiency in fitting function, even without underlying physical knowledge. By using advanced machine learning techniques such as support vector regression [8], Gaussian process regression [5], and neural network (NN) [9], people can precisely predict SOH and RUL of Li-ion batteries. While ML methods can be applied to both SOH and RUL estimation, there is a big difference between these two applications in terms of input features and desirable output. The input features for SOH estimation should be extracted from the BMS during operation and the outputs are the estimated capacity at a given time. In contrast, the input features for RUL prediction generally require the estimated or measured SOH information, such as the capacity values, to predict remaining lifetime or cycles. For example, Wu et al. [6] built an RUL prediction model

CS230: Deep Learning, Fall 2021, Stanford University, CA.

based on feed forward neural network (FFNN). The input features in this model were the recorded voltages during the battery charging process in each cycle. The output was the current cycle number of the battery. With the total cycle number determined by when the battery comes to its end of life (EOL), the RUL can be calculated by subtracting the current cycle number from total cycle number.

As for SOH estimation, Yang et al. [10] used a three-layer neural network model (NNM), which was combined with first-order equivalent circuit model (ECM) to predict the SOH within 5% error. Given a cycling profile, the ECM could simulate the current and voltage of the battery and the output of ECM was used as the input to the NNM to predict the SOH of Li-ion batteries. However, SOH and RUL estimation could not fully represent the degradation behavior of the Li-ion batteries due to the existence of a point (typically called the aging knee) at which degradation rate is increased significantly. Thus, predicting the whole aging trajectory of Li-ion batteries has become a new option. Li et al. [3] proposed a deep learning model with multiple long short-term memory (LSTM) layers in both encoder and decoder blocks to predict the entire capacity degradation trajectory in one shot. This model could learn the initial cell variation and make accurate predictions with only the first 100 cycles of data as the input. It kept updating the input with more cycle data and the model performances improved correspondingly. The best-case median prediction error over the lifetime was 1.1% with normal data and 1.3% with noisy data. Additionally, this model could also predict the EOL point and the aging knee point. However, this model is built on a single cycling condition, so the application of this model is limited and is not suitable for all use cases of Li-ion batteries.

Although those works mentioned above developed machine learning models with good performances, their experimental data sets were relatively small, mostly containing cycling data collected from a limited number of cells. Therefore, the reliability of those models is questioned when they are used for a large number of cells with initial variations. Attia et al. [1] tested 48 batteries simultaneously and used the cycling data to build a closed-loop optimization system for batteries using machine learning methods. This system combined an early-prediction model with a Bayesian optimization algorithm, and optimized parameters over 224 unique six-step, ten-minute fast-charging protocols to find charging protocols with high cycle life. With a larger experimental data set, the result of this paper is more reliable and valid for a broader battery usage. We also collaborated with their group and obtained even more cell cycling data (385 Tesla cells) from Stanford Linear Accelerator Center (SLAC).

3 Datasets

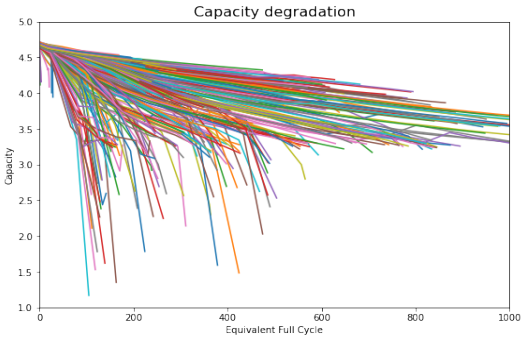

Figure 1. Degradation curves of all empirical cells data.

3.1 Experimental data description

The experimental data come from SLAC (Prof. William Chueh's group). The raw dataset consists of battery cycling data from 385 Tesla cylindrical cells, which use Nickel-Cobalt-Aluminum (NCA) oxides as cathode materials and silicon oxides with graphite as anode materials. The cycling protocols for the aging experiment have more than 100 different combinations of cycling parameters. There are six variable cycling parameters, including charge current in two different stages, discharge current, charge cutoff voltage, discharge cutoff voltage, and constant voltage charge time. Aside from the cycling protocols for each cell, two additional parameters, initial battery cell capacity and resistance values, are also recorded and used as input features. It is noteworthy that the battery cells were run for different numbers of cycles and ended up with different degradation levels. In general, cycling stopped when either of the following two conditions were met: 1) cells degraded to below 80% of their initial capacity (some cells were further degraded to below 70% of their initial capacity); 2) experiments (including charging and discharging) were run for longer than 3 months or a cell was run for more than 5,000 cycles.

3.2 Raw Data Preprocessing

To predict and plot the aging curve, it needs to be defined quantitatively. For a battery degradation trajectory, the independent variable X of the fitting formula is the battery cell equivalent full cycle number, and the dependent variable Y is the cycle-specific battery capacity. Based on existing literature, quadratic equations prove to be best fitting formulas because they are relatively easy to fit and exhibit satisfactory accuracy[7]. Equation 1 is a basic quadratic equation, where C_n stands for the cell capacity of each cycle and n represents the cycle number. The fitting curve coefficients α_1, α_2, and α_3 are outputs generated from our NN model.

$$Equation\ 1: \ C_n = \alpha_1 n^2 + \alpha_2 n + \alpha_3$$

During data cleaning, the cell data with fitting errors higher than 5% (RMSE) are removed for better training results. This leaves us with 290 cells from the original dataset. Figure 2 shows that the quadratic equation fits well with experimental data. Although the number of data points are very limited, the

%RMSE values of different curve shapes remain small.

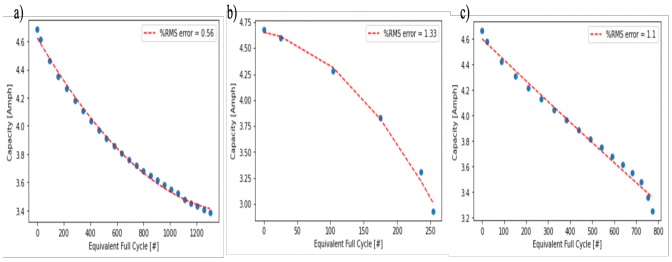

Figure 2. Fitting results of different cells

3.3 Input and Output Data Preparation

Input Argumentation: To better account for polynomial relationship of our input data, two times more parameters are generated by raising the eight parameters to the second power and third power. Tuning the data augmentation level also changes the model prediction accuracy. The tuning process and results are discussed in later sections.

Output Normalization: To optimize the battery degradation prediction, our model minimizes the errors among predicted $\hat{\alpha}$ versus measured coefficient α. Inspecting the output data distribution from the left plot in Figure 3, it has high discrepancies among different coefficient. To bring all variables to the same range, each coefficient is normalized in two ways by adapting [Equation 2], where α_j is the j^{th} coefficient, which is normalized by data range (Equation 2.1) and z-score (Equation 2.2). The normalized results are shown in the center (range) and right (z-score) plot in Figure 3.

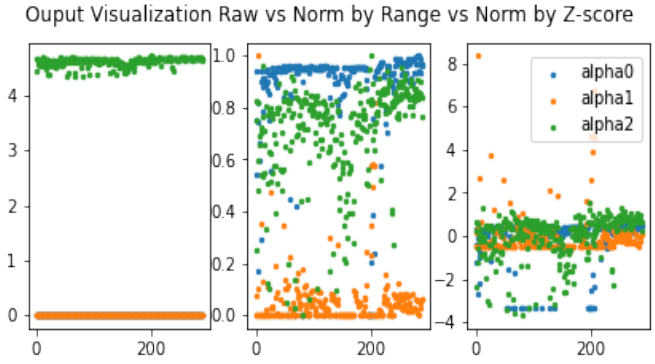

Figure 3. Distribution of 3 curve coefficients. (α_1,α_2,α_3)

$$Equation\ 2.1 : \alpha_{j-norm-range} = \frac{\alpha_j - \alpha_{j\,\min}}{\alpha_{j\,\max} - \alpha_{j\,\min}}$$

$$Equation\ 2.2 : \alpha_{j-norm-zscore} = \frac{\alpha_j - \tilde{\alpha_j}}{\sigma_{\alpha_j}}$$

After range normalization, both α_1 and α_2 end up with relatively concentrated distributions, while α_3 is distributed much more sparsely. This is because both α_1 and α_2 are small numbers (mean of α_1 is around 10^{-6} and mean of α_2 is around 10^{-3}) that fit the experimental data, but α_3 represents the initial capacity of each cell, which falls within a narrow range between 4 to 5. As a result, $\alpha_1$or α_2 are more vulnerable to outliers. Therefore, z-score normalization will be also applied, which usually mitigates the outliers effect.

4 Methodology

4.1 Neuron Network Structure

The eight basic input parameters and their augmented forms are fed into a neural network of L hidden layers and each hidden layer has $n^{[l]}$ number of neurons, where l is a positive integer between 1 and L. Both L and $n^{[l]}$ need to be tuned to optimize model accuracy (Table 1). The activation function between layers are all *ReLu*. The output layer yields the coefficients of the battery degradation curve, as shown in Figure 3 below.

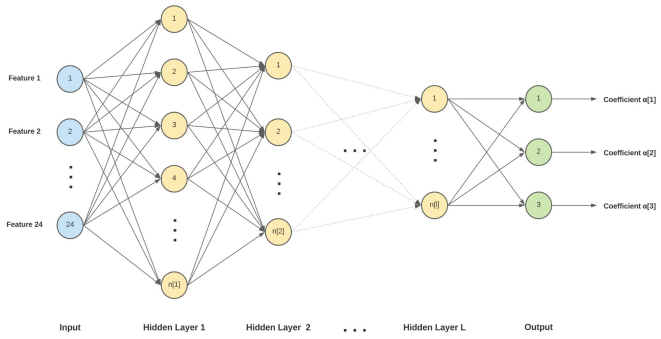

Figure 4. NN architecture assuming no correlation among coefficients

4.2 Hyperparameter Tuning

The dataset split is 75% Training, 10% Validation, 15% Testing. To optimize hyperparameter setting, 2640 different models are tested and their settings are summarized in the Table 1.

Argu	Loss Fun	Class Weight	Neuron Layer	Optimizer	Epochs
1	MSE	1:1:1	[40,10]	Adam	150
2	MAE	2:1:1	[70,10]	RMSprop	300
3		4:1:1	[70,40]		
		6:1:1	[100,40]		
		8:1:1	[100,70]		
		1:3:1	[100,10]		
		2:3:1	[100,70,40]		
		4:3:1	[100,70,10]		
		6:3:1	[100,40,10]		
		8:3:1	[70,40,10]		
			[100,70,40,10]		

Table 1. All tuning settings, including 1. Argumentation of input data(i^{th} polynomial terms generated); 2.Loss Function; 3.Class Weights (Weights of each output in evaluation metrics); 4.Neuron Layers (the i^{th} number indicates the neurons number in i^{th} hidden layer); 5.Optimizer;6.Epochs

Loss Function: We customize the loss function of the Mean Absolute Error (MAE) to that of the Mean Square Error (MSE), which is typically used for regression losses since larger deviation are penalized more by MSE compared to MAE. The constant coefficient β_j is added to indicate the relatively weights for each α_j, and it will be tuned as

parameter class weight based on Table 1. to achieve the best solution.

$$Equation\ 3.1 : MAE = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{3}\beta_j\left|\alpha_j^i - \alpha_{jtrue}^i\right|$$

$$Equation\ 3.2 : MSE = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{3}\beta_j\left(\alpha_j^i - \alpha_{jtrue}^i\right)^2$$

5 Results and Discussions

5.1 Baseline Model

Our NN model first uses loss function in Equation 3.1 by weighing α_j equally ($\beta_j = 1.$) Figure 5 displays six examples of original model prediction versus empirical results. In the bottom three examples, although predicted capacities tend to deviate slightly more from measured values as the cycle number gets larger, errors remain in an acceptable range. In comparison, model predictions in the top three examples poorly match empirical outcomes at the beginning of the aging curves, and the gaps widen as more cycles are run. To make matters worse, some prediction trajectories end up concave up, which drastically differ from the fitting curves of empirical results. For example, in Figure 5b, the coefficients of the fitting curve (positive α_1 and a negative α_2) and the predicted curve (negative α_1 and positive α_2) are of opposite signs. These initial results show that there are some discrepancies between the predicted curves and real curves. Thus, an error analysis was conducted to identify the root causes.

5.2 Error Analysis

It was found that the initial normalization method by range (making all values range between 0 and 1) introduced significant error when calculated coefficients were scaled back up for fitting. To solve this problem, coefficients were instead normalized by their z scores (making the mean of all values is 0 and the standard deviation is 1) in 5e. Comparing 5e to 5b, the improvement was noteworthy. Table 2 lists the absolute errors of three curve coefficients (α_1,α_2,α_3). The model prediction over α_1 has much higher error when compared with other two coefficients. Mathematically in a quadratic function, the coefficient of x square plays a more defining role than other coefficients. Therefore, the deviations of the predicted curves likely result more from error in α_1.

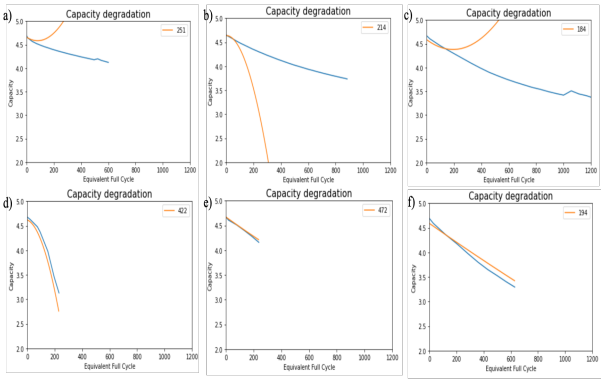

Figure 5. Prediction results of six individual cells. Subfigures a-c: three examples of poor model prediction. Subfigures d-f: three examples of good model prediction.

	Predicted Value			Fitting Value			Absolute Error (%)		
cell sequence	251	214	184	251	214	184	251	214	184
alpha1	1.14e-05	-2.78e-05	5.55e-06	9.94e-07	5.85e-07	7.08e-07	1000	4852	684
alpha2	-1.84e-03	1.15e-05	-2.14e-03	-1.42e-03	-1.53e-03	-1.92e-03	29.6	101	11.4
alpha3	4.658	4.641	4.587	4.642	4.638	4.646	0.345	0.065	1.27

Table 2. Error Summary of 3 curve α predictions.

Based on the above error analysis, the unsatisfactory pilot model outcome may have resulted from inadequacy in the original loss function. More specifically, it fails to accurately represent the error between the predicted aging trajectory versus the experimental degradation curve. Specifically, as the cycle number n increases, coefficient α_1 weighs more and more compared to α_2 in the C_n calculation (Equation 1). Mathematically, error in α_1 impacts C_n more significantly than error in α_2 does. Because our current loss function does not scale up the weight of α_1, it does not fully capture the degradation properties.

5.3 Hyperparameter Tuning Results

To improve model accuracy, different class-weights for α_1, α_2 and α_3 are introduced to the loss function. Figure 6 shows the prediction results, measured by R^2 values, when the considering different class-weights of the loss function and augmentation method. The x-axis represents the data augmentation method (e.g., one means the model only inputs original parameters, while three means the model inputs original parameters along with all the second power and third power of original parameters). The lines with different colors represent the model R^2 performances using different class-weights. Generally speaking, the prediction R^2 value increases as the class-weight ratio of α_1, α_2 and α_3 increases. Under both loss function choices, settings that weigh α_1 more heavily outperform the alternatives. However, more data augmentation doesn't seem to improve the prediction performances – the R^2 value actually drops when more augmented data are added in all three cases. Hence, we conclude that tuning the class weights of the loss function plays a critical role in allowing the NN model to produce better results.

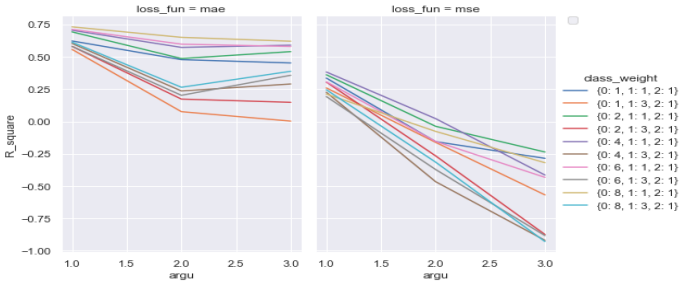

Figure 6. Class-weights tuning results

Aside from class weights, other tuned hyperparameters include the optimization method, the loss functions, the number of neural network hidden layers, the number of neuron in each hidden layers, and the number of epochs.

Figure 7 shows the summary of tuning results. Figure 7 (a) reveals that RMSprop performs slightly better than Adam. Figure 7 (b) shows the effect of tuning the number epochs, and more epochs apparently helps the model improve prediction R^2 values. Figure 7 (c) clearly demonstrates that MAE has higher prediction R^2 values than MSE does in all data augmentation cases. Finally, Figure 7(d) indicates that 3-layer models perform better, which addresses the problem of underfitting in the baseline model. As a result, our best model result with settings combination below:

Argu	Loss Fun	Class Weight	Neuron Layer	Opt	Epochs
2	mae	8:1:1	100:40:10	Adam	150

,which result in a best R^2 of 0.91.

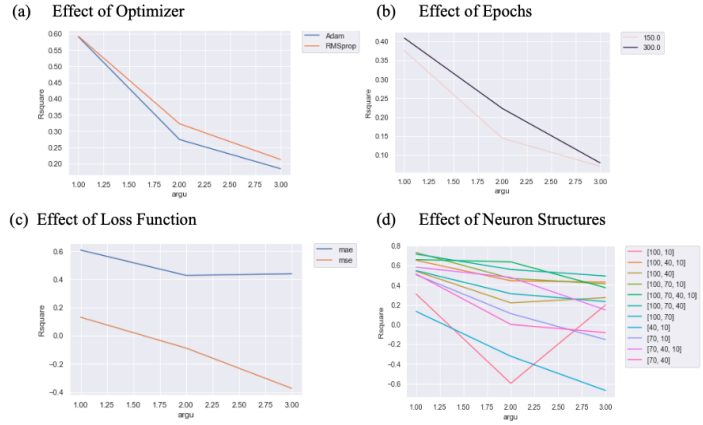

Figure 7. Other hyperparameters tuning results. a) tuning result of optimization method. b) tuning result of loss function. c) tuning result of number of epochs. d) tuning result of number of hidden layers and neurons.

In conclusion, the model performance is more sensitive to the class-weight, loss function, and neurons of layers structures, while other hyperparameter like optimizer and epochs have much less impact on prediction errors. Tuning hyperparameter helps our model improve accuracy.

5.4 Prediction Results

After adjusting the loss function and tuning the hyperparameters, our updated NN model predicts battery degradation trajectory curves much more accurately than the original model. Figure 8 shows that compared to the original model, predicted degradation trajectories of the updated model look more similar to the experimental data. Although for certain cells there are still some noticeable prediction deviations, the overall R^2 value at 0.91 means our model explains 91% of the variance.

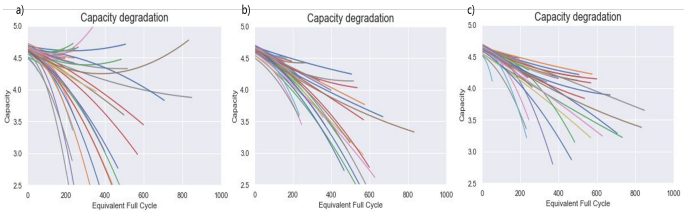

Figure 8. Degradation prediction of original model(left), updated model(middle) and actual result(right)

A closer look at the prediction curves for individual cells reveals model improvements. Figure 9 compares predictions from the original model to those of the updated model. In summary, the original model makes three types of mistakes (shown in Figure 9, subfigures (a-c)): 1) predicts α_1 with positive value while the true value is negative; 2) predicts α_1 with negative value while the true value is positive; 3) predicts α_1 with larger positive value than true value. Fortunately, our updated model is able to tackle all of the three error cases (shown in Figure 9, subfigures (d-f)). The prediction performance improvement from modifying the original NN model validates the point that our updated model accomplishes the goal of using cycling conditions to predict cell degradation trajectories with high accuracy.

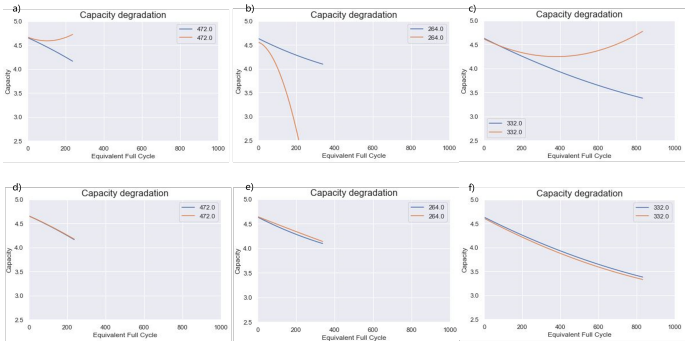

Figure 9. Prediction improvements on test cells

6 Conclusion and Future Work

In conclusion, we built a neural network model for predicting the aging trajectory of Li-ion batteries under different charging protocols. Our model takes different cycling conditions as input and outputs the coefficients of the aging curve for each individual cell. The primary results show that our model has a $R^2 = 0.91$ in regression fitting.

Our NN model works very well for certain battery configurations and cycling conditions. As for the future work, we will 1) collecting more cell data from different manufacturers to test the generality of our model; 2) inspecting deeper in the error analysis in batteries itself, diagnosing those batteries with unpredictable performance.

7 Code Availability

All code is available at `https://github.com/zuyeyang/Stanford_Battery_Project.git`. Feature Process and Machine Learning were performed in python.

8 Contribution

All members of the team contributed to all aspects of the project in general, but each member was more involved in certain areas than others. Jihan Zhuang took the lead on

researching the initial topic, coding for data pre-processing, and conducting error analysis. Benson Zu played a primary role in building the NN model and tuning the hyperparameters. Lude Rong was responsible for quantitative model metrics and refining the writing.

Bibliography

[1] Peter M. Attia et al. "Closed-loop optimization of fast-charging protocols for batteries with machine learning". In: *Nature* 578.7795 (Feb. 20, 2020), pp. 397–402. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-020-1994-5. URL: http://www.nature.com/articles/s41586-020-1994-5 (visited on 10/06/2021).

[2] Xiaosong Hu et al. "Technological Developments in Batteries: A Survey of Principal Roles, Types, and Management Needs". In: *IEEE Power and Energy Mag.* 15.5 (Sept. 2017), pp. 20–31. ISSN: 1540-7977. DOI: 10.1109/MPE.2017.2708812. URL: http://ieeexplore.ieee.org/document/8011541/ (visited on 10/06/2021).

[3] Weihan Li et al. "One-shot battery degradation trajectory prediction with deep learning". In: *Journal of Power Sources* 506 (Sept. 2021), p. 230024. ISSN: 03787753. DOI: 10.1016/j.jpowsour.2021.230024. URL: https://linkinghub.elsevier.com/retrieve/pii/S0378775321005528 (visited on 11/29/2021).

[4] Kailong Liu et al. "A Data-Driven Approach With Uncertainty Quantification for Predicting Future Capacities and Remaining Useful Life of Lithium-ion Battery". In: *IEEE Trans. Ind. Electron.* 68.4 (Apr. 2021), pp. 3170–3180. ISSN: 0278-0046, 1557-9948. DOI: 10.1109/TIE.2020.2973876. URL: https://ieeexplore.ieee.org/document/9040661/ (visited on 10/01/2021).

[5] Kailong Liu et al. "Modified Gaussian Process Regression Models for Cyclic Capacity Prediction of Lithium-Ion Batteries". In: *IEEE Trans. Transp. Electrific.* 5.4 (Dec. 2019), pp. 1225–1236. ISSN: 2332-7782, 2372-2088. DOI: 10.1109/TTE.2019.2944802. URL: https://ieeexplore.ieee.org/document/8853281/ (visited on 11/29/2021).

[6] Ji Wu, Chenbin Zhang, and Zonghai Chen. "An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks". In: *Applied Energy* 173 (July 2016), pp. 134–140. ISSN: 03062619. DOI: 10.1016/j.apenergy.2016.04.057. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306261916304846 (visited on 10/06/2021).

[7] Yinjiao Xing et al. "An ensemble model for predicting the remaining useful performance of lithium-ion batteries". In: *Microelectronics Reliability* 53.6 (June 2013), pp. 811–820. ISSN: 00262714. DOI: 10.1016/j.microrel.2012.12.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S0026271412005227 (visited on 10/06/2021).

[8] Zhiwei Xue et al. "Remaining useful life prediction of lithium-ion batteries with adaptive unscented kalman filter and optimized support vector regression". In: *Neurocomputing* 376 (Feb. 2020), pp. 95–102. ISSN: 09252312. DOI: 10.1016/j.neucom.2019.09.074. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925231219313426 (visited on 11/29/2021).

[9] Boyuan Yang, Ruonan Liu, and Enrico Zio. "Remaining Useful Life Prediction Based on a Double-Convolutional Neural Network Architecture". In: *IEEE Trans. Ind. Electron.* 66.12 (Dec. 2019), pp. 9521–9530. ISSN: 0278-0046, 1557-9948. DOI: 10.1109/TIE.2019.2924605. URL: https://ieeexplore.ieee.org/document/8752268/ (visited on 11/29/2021).

[10] Duo Yang et al. "A Neural Network Based State-of-Health Estimation of Lithium-ion Battery in Electric Vehicles". In: *Energy Procedia* 105 (May 2017), pp. 2059–2064. ISSN: 18766102. DOI: 10.1016/j.egypro.2017.03.583. URL: https://linkinghub.elsevier.com/retrieve/pii/S1876610217306355 (visited on 10/06/2021).