Korean J. Chem. Eng., 40(8), 1850-1862 (2023)
DOI: 10.1007/s11814-023-1476-1

pISSN: 0256-1115
eISSN: 1975-7220

Feature construction for on-board early prediction of electric vehicle battery cycle life

Junseop Shin*, Yeonsoo Kim,†, and Jong Min Lee*,†**

*School of Chemical and Biological Engineering, Institute of Chemical Processes, Seoul National University, Gwanak-ro 1, Gwanak-gu, Seoul 08826, Korea
**Department of Chemical Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 01897, Korea
(*Received 22 December 2022 • Revised 7 April 2023 • Accepted 17 April 2023*)

Abstract−As the worldwide environmental crisis worsens, electric vehicles (EVs) are establishing themselves as eco-friendly alternatives to conventional fossil fuel vehicles. Lithium-ion batteries (LIBs) are a typical source of energy for EVs, but it is important to predict their life in order to ensure safe and optimal operation. However, because LIBs degrade in a nonlinear fashion and their state of health varies depending on operating conditions, achieving fast and accurate cycle life prediction has been a challenge. More importantly, on-board estimation is necessary because even the identical battery cells manufactured by the same company vary in their cycle lifetimes and operational characteristics, which we cannot specify in advance. In this paper, we propose a set of novel features that enable on-board battery cycle life prediction while maintaining high memory efficiency and low calculation complexity. The features' performances were evaluated using a variety of machine learning models, ranging from simple linear elastic nets to nonlinear neural networks.

Keywords: Lithium-ion Battery, Cycle Life Prediction, Feature Construction, Machine Learning

INTRODUCTION

With the invention of the internal combustion engine, the fast expansion of fossil fuel consumption over the 20th century has resulted in serious global warming. As a result, global demand for environmentally friendly power systems is increasing, and electric vehicles (EVs) and hydrogen vehicles are emerging as viable alternatives to meet this demand [1,2]. Lithium-ion batteries (LIBs) are the primary energy source of EVs due to their low standard reduction potential resulting in high power, and their light weight resulting in a high specific energy as well as a high gravimetric and volumetric energy density [2,3].

The comparably compact size enables LIBs to be packed in series, matching their wide range of power requirement from portable electric devices to heavy-duty vehicles [4,5]. The advantages of low energy costs, long cycle life, and environmental protection also support their widespread use [6]. To maximize these benefits and to ensure safe and reliable battery operation, it is necessary to combine multifaceted battery state management with long-term scheduling [7-9]. However, due to the complex chemical and physical degradation mechanisms occurring in each part of the LIB [10-13], it is challenging to accurately monitor the state of health (SOH) of LIB and forecast its end of life. In addition, those deterioration characteristics of batteries make their remaining useful life (RUL) vary even when they are manufactured in the same factory and the same specifications. Because it is impossible to anticipate those random variances, effective test sessions should be in high demand. Since LIB will deteriorate even during the test, it will be required to diagnose its condition quickly in the early stages of operation as possible with only a modest quantity of data.

Along with these purposes, many studies related to the battery management systems (BMS) in EVs try to monitor the capacity decay ratio, which is defined as an indicator of the battery's SOH or the RUL [14,15]. Doyle et al. [16] propose a generic battery model for cells having a lithium anode, polymer electrolyte, and insertion cathode. Arora et al. [17] develop a mathematical model based on lithium deposition and overcharge behavior. Li et al. [18], as well as Song et al. [19], conduct research on empirical models.

Given the difficulties in expressing complicated degradation mechanisms using first principles, researchers have extensively explored data-driven modeling based on machine learning (ML) techniques. Various ML techniques have been studied, including support vector regression (SVR) [9,20], naive Bayes (NB) method [8], elastic net (Enet) [21], gradient boosting (GB) method [9], Bayesian regression [22], random forest regression (RFR) [23], and Gaussian process regression (GPR) [24,25].

To deal with the nonlinearity of degradation mechanisms, several studies have employed neural network-based techniques. Liu et al. [26] used recurrent neural networks (RNNs), while Peng et al. [6] proposed a fusion method that modifies long short-term memory network (LSTM) for battery SOH prognostics. Ren et al. [22] and Yang et al. [25] predicted battery RUL using a multi-layered deep neural network (DNN) and a convolutional neural network (CNN), respectively. Furthermore, hybrid approaches that use two or more techniques have also appeared. Alipour et al. [27] used SVR and GPR, Tang et al. [28] used CNN and LSTM, Ansari et al. [29] used RNN and particle swarm optimization (PSO) technique, and Yao et al. [30] achieved a high level of SOH prediction by utilizing CNN, LSTM, and the recently issued graph neural network (GNN) concept together.

†To whom correspondence should be addressed.
E-mail: kimy3@kw.ac.kr, jongmin@snu.ac.kr

What to use as an input feature also has a significant impact on the prediction performance as much as the ML technique. Many studies simply select or generate features based on the most representative factors of battery operation, such as voltage, current, impedance, and operation temperature. Along with utilizing those basic measurements, some perform additional calculations on them to ascertain more efficient features. Maddikunta et al. [27] use principal component analysis (PCA) with data encoding method to reduce feature dimensions. To sort out significant features, Ren et al. [22], Xu et al. [28], and Tong et al. [29] propose autoencoder-based feature extraction algorithms. These methods seem to have merit in that they do not require extensive background knowledge of battery systems to deduce some unique features that are difficult to come up with intuitively. However, the features obtained through these numerical algorithms frequently take complex forms, making them difficult to calculate and intuitively understand their physical meaning.

Thus, we propose a novel and simple feature set in this study that enables reliable battery cycle life prediction using a small amount of early cycle data. Operation data of commercial LFP (lithium iron phosphate)/graphite cells [21] is used for analysis. By analyzing representative factors in the battery system, new features that can simply indicate the degradation of battery were derived. To evaluate predictive performance of the proposed feature set, several ML techniques were introduced to validate the efficacy of the proposed features. Then, a quantitative comparison was conducted between our proposed model and the results obtained in relevant previous studies [21,27-29,32,34-38]. We further analyzed from the simple linear correlation between each feature and battery cycle life to the feature importance they represent in each ML models.

The new feature set proposed in this study shows improved cycle life prediction performance while reducing computational cost and memory demand. And the prediction is possible in the early period compared to the entire battery lifespan. In addition, its simple form makes it easy to intuitively understand the physical meaning of itself.

The remainder of this paper is organized as follows. Section 2 discusses some fundamental knowledge regarding the LIB system and its degradation mechanisms. Section 3 describes the process of constructing our feature sets and the various learning methods we employed to analyze prediction performance. Section 4 presents the predictions of multiple ML models using the proposed feature set, along with correlations between each attribute and battery cycle life. Additionally, a comparison of the outcomes of other studies is included. Finally, Section 5 summarizes our work and provides concluding remarks.

BACKGROUND

1. Battery Degradation Mechanisms

Batteries consist of two electrodes, electrolytes, and some porous separators between the two electrodes. Active materials are often coated on the electrode surface where the redox reaction with electrolyte mainly occurs [4,39]. Battery degradation takes place in all of these structures. The causes can be categorized as chemical and mechanical degradation with further subgroups for each type of mechanism [12].

The generation of solid electrolyte interphase (SEI) is often considered as the dominant factor [40,41]. An undesired side reaction of the electrolyte generates some solid by-products which tend to stack on the surface of the electrode, and they even float in the electrolyte as they increase. The stacked SEI inhibits the explosive reaction on the electrode surface, resulting in some initial stability; however, as this layer thickens, it blocks the least amount of red-ox reaction, resulting a gradual decrease in battery power [42,43]. From a mechanical standpoint, the intercalation of SEI through the gap between stacked electrode layers or gas generation from unwanted side reactions leads to the exfoliation of the layered structure, resulting in severe physical stress [11,44].

During the progression of above degradation, the following characteristics are typically observed: the voltages under the same conditions gradually decrease, as do the charge and discharge capacities. In addition, as the by-products floating around the electrolytes increase over time, resistance of ion transport increases, resulting in a high impedance and operating temperature [11].

In the very early stages of a battery's cycle life, the above phenomena occur rapidly and are highly variable. During this period, the battery's operation not only has degrading effects, but also has some positive effects in terms of preventing abrupt reactions on the surface of electrodes. After this period, battery cell capacities tend to stabilize [45]. This is the formation cycle period, which can be distinguished from the subsequent degradation period.

2. Battery Cycle Features

The degradation tendencies are often similar, but the degree of those changes varies from battery to battery, and even cycle-to-cycle within a single battery. Not only the inherent conditions, but also the external operating conditions can affect those patterns. To accommodate overall states, it is crucial to extract comprehensible features from the unique changing patterns of the aforementioned factors.

Statistical operations such as average, variance, minimum, and maximum values based on varying temperature, internal resistance, and charging and discharging capacities have been commonly tested in several studies [21,32,35]. Ren et al. [22] selected features from specific intervals on the aforementioned curves, and carried out some new dimensional features using autoencoder and feature fusion process. Xu et al. [32] fit the capacity fade curve to a linear model, and utilized its slope and intercept values as prediction parameters. Similarly, Fei et al. [35] attempted to fit the curve into several other forms and obtained related parameters. Yang et al. [24] obtained slope values from multiple locations along the constant current (CC) charging curve and use them for RUL prediction.

Other attempts were made to gain insight from the correlation between two factors rather than relying solely on the changing pattern of a single factor over time [21]. One of the correlations is the discharge curve, a plot of discharge capacity as a function of voltage, illustrated in Fig. 1. The function is notated as Qd(V), where Qd and V indicate discharge capacity and voltage. As the cycle number increases, the discharge capacity corresponding to the same voltage decreases, causing the area under the discharge curve to decrease. The voltage range corresponding to the main operating session, where the largest curve shrinkage occurs, decreases simultaneously. This behavior represents the discharge energy dissipation resulting from the degradation of battery.

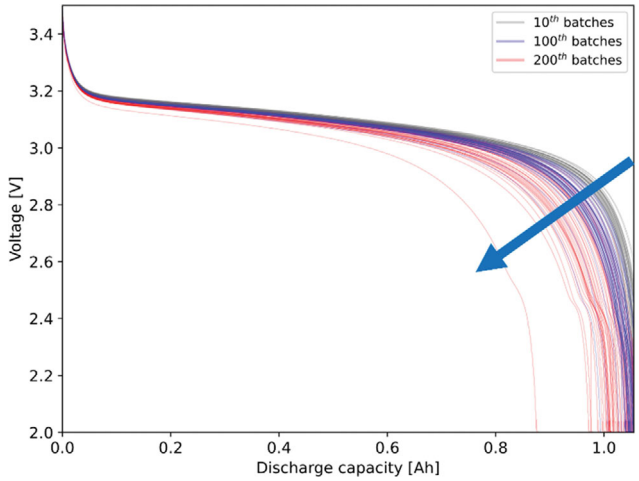

Fig. 1. Discharge curves of train and test data. The curves of 10th, 100th, and 200th cycles corresponding to each experimental batch are plotted. As the cycle number increases, the curve shrinks down towards the arrow direction.

Under galvanostatic conditions, the amount of discharged energy dissipation depends on the decreased area under the discharge curve and the decrease in Qd(V) linearly. For this reason, the cycle-to-cycle variation of Qd(V), denoted as ΔQd(V), and its mean and variance values are suggested as informative features for evaluating battery RUL [21]. Not only for the LIBs, but also for other types of batteries [46,47], the area under the discharge curve gradually decreases from cycle-to-cycle. We also focused on this curve, anticipating that it would provide a crucial indicator for predicting battery cycle life.

METHODOLOGIES

In this study, we utilized experimental data from an academic-industrial partnership between MIT, Stanford, and Toyota Research Institute (TRI). Specifically, commercial LFP/graphite cells (A123 systems, model APR 18650M1A, 1.1Ah nominal capacity) were cycled in a temperature-controlled environmental chamber maintained at 30 °C. The cells were subjected to varying fast-charging conditions while the discharging conditions remained identical. The end of a battery's cycle life was determined as the time when its performance falls below 80% of nominal capacity, as defined in a previous study [21]. The utilized battery cycle data are 41 train batches and 43 test batches whose cycle lives are distributed up to about 2000 cycles (Fig. 2).

During data processing, one of the test batches was excluded because it reached the 80% level of nominal SOH significantly faster than the others, and exhibited entirely different patterns [21]. Furthermore, neither the secondary test data was considered in this study. Since the secondary test results were obtained after one year of aging, their tendencies were significantly different from the train and test data. As one of the primary goals of this study is to achieve early life prediction, dealing with aging data is viewed as a distinct matter. Therefore, only the train and primary test data were used for feature construction and subsequent analyses.

1. Feature Construction

Severson et al. [21] examined the discharge curve from each cycle as a function for Qd of V, and focused on the cycle-to-cycle variation of this function, ΔQd(V). The difference in Qd(V) between 100th and 10th cycles was calculated at several voltage points. Using statistical summary tools, features such as mean, variance, skewness and kurtosis of ΔQd(V) values were obtained. Note that the spline interpolation for Qd(V) should be conducted first in this approach to calculate the differences of Qd(V) (ΔQd(V)) between two cycles on each same voltage point, because the raw sampled voltage points cannot always be identical and evenly distributed. However, this interpolation process necessitates the storage of all sampled points during the specified two cycles, and wastes computing time for the calculating over whole cycle range. Moreover,

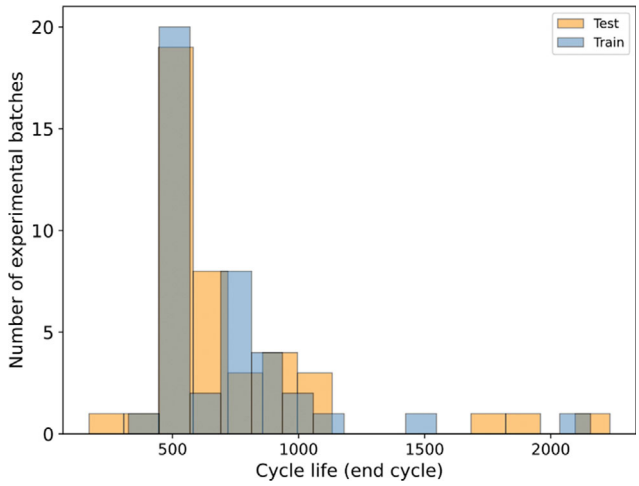

Fig. 2. The distribution of battery cycle life data used in this study. They are primarily concentrated in the small range between 500 and 1000 cycles, but some of them go up to around 2000 cycles.

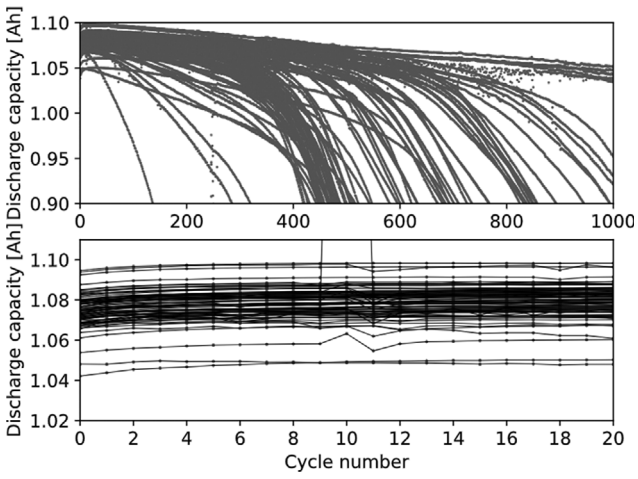

Fig. 3. Discharge capacity plots along with increasing cycle number. Fluctuating behaviors were observed around the 10th cycle, indicating that the formation cycle period may exist under that region.

the reliability of resulting ΔQd(V) values highly depends on the accuracy of interpolation. To address those issues, we tried a pointwise approach instead of exploiting the entire curve data points.

First, we set a criterion for the earliest cycle of data to be used. Examining the experimental data, perturbations of discharge capacity near the first 10 cycles were observed frequently, as depicted in Fig. 3. It implies that the influence of the formation cycle period has extended to that point; thus, the 10th cycle was chosen as the beginning of cycle data to be used for feature construction. The upper boundary criterion was set as a similar or slightly reduced cycle number based on prior studies, as it would be difficult to expect an accurate prediction with an unconditionally small amount of data. In studies that used the same experimental data, Severson et al. [21] utilized data up to the 100th cycle, while Yang et al. [34] used data up to approximately the 250th cycle, and Xu et al. [32] used data up to the 300th cycle. Zhou et al. [37] and Lin and Chai [38] even utilized data from the first 30-70% cycles in their studies. It was clear that the amount of data usage and the computational efficiency would have a trade-off correlation; thus, we initially considered comparably early cycles of the 100th and 200th as candidates for the upper bound, and compared both results.

The feature construction scheme is described in Fig. 4. The following scheme is described with the case of the upper bound of the 200th cycle as a representative. The cycle number at which a factor or a feature was obtained will be indicated in a superscript. In all cases where the base of the exponent is the name of a factor or feature, the superscript does not indicate that the value is squared.

To reflect the trends of both Qd and V factors, we focused on the slope value of the discharge curve and the tangent value on the curve where the curvature changes most. To utilize the tangent at the specific point without exhausting interpolation and differentiation process, we just set two specific points on either side of the curve. Then the slope value of the tangent at the target point on the curve could then be approximated by the mean value theorem: the slope value of the straight line connecting the aforementioned two points. As for the two points, the voltage values at which the cycle-to-cycle ΔQd(V)s appear the largest and the smallest on the discharge curve were selected. This is reasonable because the obtained slope value accommodates the region where the greatest cycle-to-cycle variation occurs around 3 V (Fig. 5(a) and Fig. 5(b)).

Next, the difference of Qd(V) between the upper and the lower bound cycle curves was calculated. The voltage values corresponding to the maximum and minimum ΔQd(V)s are subsequently referred to as HV (higher voltage value corresponding to the maximum ΔQd(V)) and LV (lower voltage value corresponding to the minimum ΔQd(V)). When processing the test data into features, it is possible to simply get Qd values corresponding to the HV and LV values derived from the train data without any iteration of interpolation process.

Since it is difficult to precisely measure voltage, the mean Qd values in the range of ±0.01 V around each HV and LV points were utilized. Then the straight lines could be drawn, connecting the two points corresponding to the HV and LV values on each cycle's discharge curve as shown in Fig. 4. The slopes of the lines obtained from each curve are notated as Qd^{10} and Qd^{200}, and defined as follows:

$$Qd^{10} = \frac{\left|Qd^{10}_{HV} - Qd^{10}_{LV}\right|}{HV - LV}, \tag{1}$$

$$Qd^{200} = \frac{\left|Qd^{200}_{HV} - Qd^{200}_{LV}\right|}{HV - LV}, \tag{2}$$

where the Qd^{i}_{HV}s and Qd^{i}_{LV}s refer to the discharge capacity values corresponding to the HV and LV points on the discharge curve

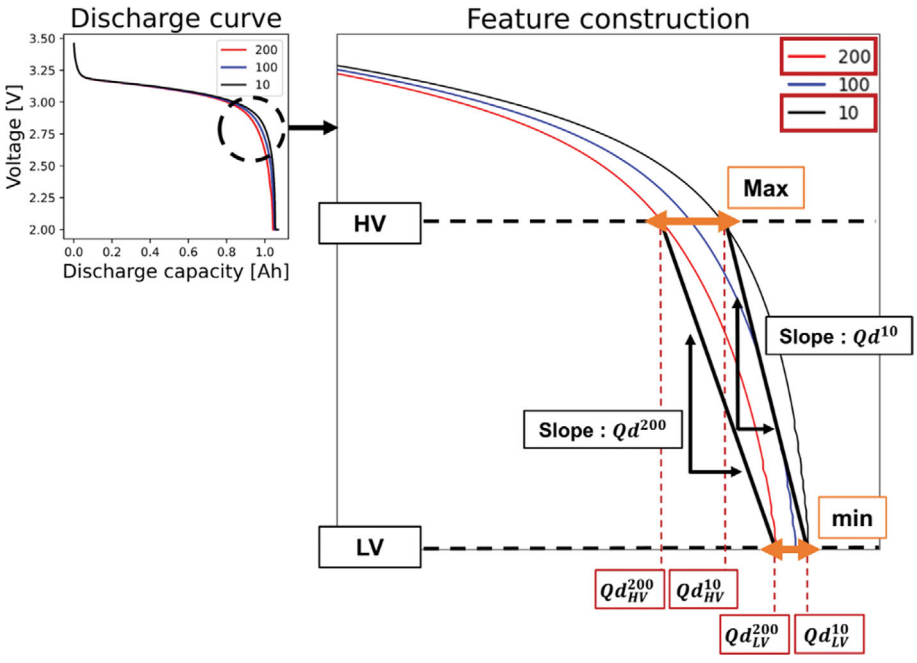

Fig. 4. Description of feature construction process.

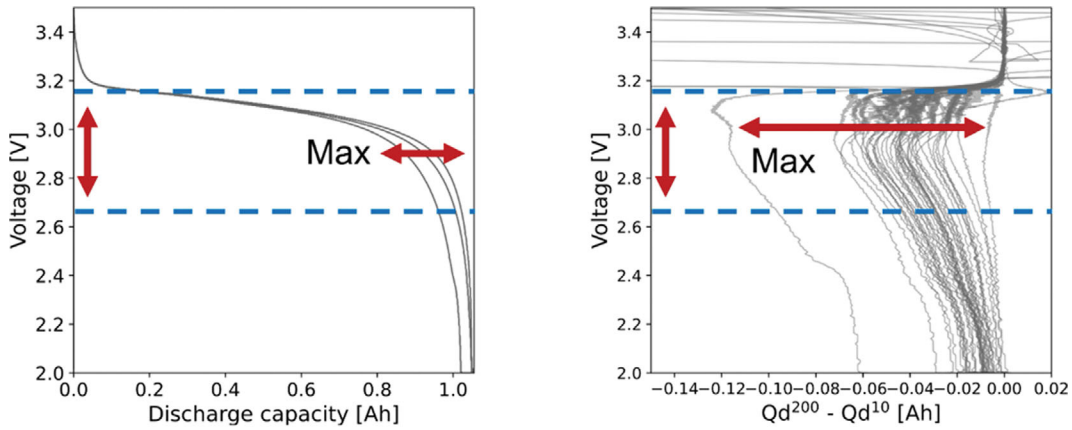

Fig. 5. Comparison between (a) discharge curve and (b) ΔQd(V) plot on the axis of voltage. Three curves of the 10th, 100th, and 200th cycles obtained from the first train batch were plotted as representatives in (a). The voltage range corresponding to the highly varying peak in (b) matches the voltage range where the maximum difference of Qd(V) appears in (a).

of i^{th} cycle.

As mentioned in Section 2.2, Battery cycle features, the variance value of ΔQd(V) can be a crucial factor for battery cycle life prediction due to the linear relationship between the change in ΔQd(V) value and the battery's energy dissipation. Therefore, a new feature was finally constructed to approximate the meaning of the variance of ΔQd(V).

Note that the HV value exists around 3 V in Fig. 5(a), which is nearly the same voltage region as the peaks appearing in Fig. 5(b). As the height of the peak increases, the range of ΔQd(V) around the peak forms a broader distribution, resulting in a greater variance value. Since the difference between the two slope values Qd^{200} and Qd^{10} captures the shrinkage of the discharge curve's cycle-to-cycle variation, it can be used to approximate the magnitude of the peak. Thus, we efficiently considered only the peaked region where the data varies the most, which is closely related to the variance of ΔQd(V), and further, the battery dissipation.

Since the ΔQd(V) obtained at the LV point is negligibly small while distinctly being large at the HV point, the difference ratio between the two slope values can approximate the curve's shrinking behavior toward the bottom-left properly. We suggest this ratio value between (1) and (2) as a new feature, notated as Δm^{200}, defined as

$$\Delta m^{200} = \frac{Qd^{200} - Qd^{10}}{Qd^{10}} = \frac{\left|Qd^{200}_{HV} - Qd^{200}_{LV}\right| - \left|Qd^{10}_{HV} - Qd^{10}_{LV}\right|}{\left|Qd^{10}_{HV} - Qd^{10}_{LV}\right|}. \quad (3)$$

In the same way, Qd^{100} and the Δm^{100} feature can be calculated for the upper bound of 100^{th} cycle case.

In Fig. 5(b), as the voltage value on the y axis approaches both extremes, ΔQd(V) does not show meaningful variation. The cycle-to-cycle behaviors are similar. Therefore, it is inefficient to incorporate the entire ΔQd(V) which includes the edge part ranging from approximately 3.2 to 3.5 V. It would reduce the ΔQd(V)'s variance pointlessly, while increasing the load of computation. On the contrary, the range of approximately 2.6 to 3.2 V marked with blue dashed lines in Fig. 5 covers the region where the ΔQd(V) varies significantly. The HV and LV points closely match this range, focusing only the region where the informative variance occurs. Thus, the proposed feature can efficiently reflect the informative changes.

Not only is the selection of the meaningful voltage range straightforward, but so is the calculation of the feature itself. It requires only four simple calculations of HV and LV points in each 10^{th} and 200^{th} cycles' curves, respectively, and has a significantly lower computational cost than other previously studied statistical summary-related features, which require many calculations including from two to four square, roots, summations of large data over the entire cycle time, and even mixed calculations of these.

Since the feature Δm is a ratio derived from the slopes between two trend lines, the calculated ratio may be the same even if the actual discharge curves have distinct shapes in detail. Considering this case, the Δm feature alone may not be sufficient for accurate prediction. Therefore, we propose using the slope values obtained from each single cycle's discharge curve, such as Qd^{10}, Qd^{100} and Qd^{200}, as auxiliary features comprising an input feature set along with Δm. These auxiliary features describe the unique characteristics in each single cycle curve, thereby enhancing the accuracy of predictions. Moreover, they have the benefit of requiring no additional computational cost, since they are already accounted for during the calculation of Δm.

2. ML Model

To evaluate the cycle life prediction performance of the proposed new features, seven ML models were applied to the problem: deep neural network (DNN), elastic net (Enet), support vector regression (SVR), kernel ridge regression (KRR), Gaussian process regression (GPR), random forest regression (RFR), and extreme gradient boosting (XGB). They are widely employed in supervised learning fields, particularly for regression problems. For the remainder of this paper, we will refer to the 'DNN' as 'NN'.

For NN, a well-implemented package, *keras* (version 2.4.3) in *Python* (version 3.7.9), *tensorflow* (version 2.4.0) was used. The remainder were implemented using *scikit-learn* (version 0.23.2)

Table 1. Neural network model settings

	Model summary
Neural network	Number of hidden layers=3, Number of nodes for each layer=9, Epoch=1000, Activation function=ReLU, Initializer=kernel : He normal/bias : zero, Regulation=L1 (coefficient~0.001), Optimizer=RMSprop (0.001), Loss function=mean squared error (MSE)

package in *Python*. The primary hyperparameters of each ML models were properly tuned by grid search and five-fold cross validations (Table 1 and Table 2) (more detailed descriptions of each ML model and the hyperparameter tuning process are provided in Supplementary Note 1).

3. Feature Analysis Methods

In addition to the prediction based on ML models, two feature analyses were conducted to determine the direct relationship between the features and battery cycle life. First, Pearson correlation coefficients were calculated to determine the linear correlations between the feature and battery life, as well as between each feature. Instead of nonlinear correlations, training and prediction accuracy would benefit more from clear linear correlations. Then, a feature importance analysis was conducted to determine how significantly each feature influences the cycle life prediction of ML models. To this end, permutation importance (PI) of features was analyzed. Supplementary Note 2 details the aforementioned analysis schemes.

RESULTS AND DISCUSSION

In this section, the prediction performance of each ML model and feature described in this section is mainly evaluated using the

Table 2. List of ML models and hyperparameter settings for each model

Type	Methods		Hyperparameters [Search range]
Linear	-	Enet	*alpha* ∈ [0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0] *l1_ratio* ∈ [0.0001, 0.001, 0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0]
Nonlinear	Kernel	GPR	kernel='Matern5/2', *noise* ∈ [0.00001, 0.0001, 0.001, 0.01, 0.1], *n_restarts_optimizer*=[0, 5, 10]
		KRR	kernel='rbf', 'poly', 'linear', *alpha* ∈ [0.00001, 0.0001, 0.001, 0.01, 0.1, 1], *degree* (for 'poly')=3
		SVR	kernel='rbf', 'poly', 'linear', *epsilon* ∈ [0.001, 0.01, 0.1, 1.0, 10, 100], *C* ∈ [1, 10, 100], *degree* (for 'poly')=3
	Ensemble	RFR	*n_estimators* ∈ [50, 100, 150, 200]
		XGB	*booster*='gbtree', *n_estimators* ∈ [50, 100, 150, 200], *max_depth* ∈ [5, 6, 7, 8, 9], *subsample* ∈ [0.7, 0.75, 0.8, 0.85, 0.9], *colsample_bytree* ∈ [0.8, 0.9, 1], *learning_rate* ∈ [0.05, 0.06, 0.07, 0.08, 0.09, 0.1]

Table 3. Summary of feature sets used in this study

Feature sets	Configuration
'*discharge*' set [21]	min(ΔQd(V)), Var(ΔQd(V)), Skew(ΔQd(V)), Kurt(ΔQd(V)), Qd(cycle=2), $\max_{cycle}$Qd(cycle)−Qd(cycle=2).
'*with* Qd^{10}' set	Qd^{10}, Qd^{100}, Qd^{200}, Δm^{200}.
'*without* Qd^{10}' set	Qd^{100}, Qd^{200}, Δm^{200}.
'*all features*' set	min(ΔQd(V)), Var(ΔQd(V)), Skew(ΔQd(V)), Kurt(ΔQd(V)), Qd(cycle=2), $\max_{cycle}$ Qd(cycle)−Qd(cycle=2), Qd^{10}, Qd^{100}, Qd^{200}, Δm^{200}.

root mean square error (RMSE) and mean absolute percentage error (MAPE) metrics, defined as

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}, \quad (4)$$

$$\mathrm{MAPE}=\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i-\hat{y}_i|}{y_i}\times 100(\%). \quad (5)$$

The data used in this study was scaled to the range of 0 to 1 by Min-Max normalization. And for the results, the RMSE values noted without parentheses are derived from descaled values, with the unit of 'cycles'. Additionally, the RMSE based on the scaled values is presented in parentheses.

For simplicity, we named the feature set consisting of Qd^{10}, Qd^{100}, Qd^{200}, and Δm^{200} as *'with Qd^{10}'* set, and the feature set with three features, Qd^{100}, Qd^{200}, and Δm^{200} as *'without Qd^{10}'* set. Table 3 summarizes each feature set.

All analyses were performed using *Python* (version 3.7.9) on a desktop with an Intel(R) Core(TM) i7-11th Generation 11700k (Rocket Lake S) CPU running at 3.6 GHz (8core, 16thread) and 8 GB DDR4-3200 RAM.

The cycle life prediction performances of the proposed features were mainly evaluated by adopting the results of [21] as a benchmark. Severson et al. [21] presented the best prediction results using the variance of ΔQd(V) between 10th and 100th cycles as the main feature, and it was used in several other studies [28,30] as a performance evaluation criterion (see Supplementary Data, Table. 1, for more detail of the benchmark features).

1. Arguments about Feature Generation Strategies

To determine the upper boundary of cycle number, both Δm^{100} and Δm^{200} related features were applied to the NN prediction model as a representative. Their results are summarized in Table 4. Whether it is a single feature or a feature set, prediction errors related to Δm^{200} are smaller by more than half. Since the 200th cycle can also be considered an early cycle in light of the fact that the cycle life ends even around the 2000th cycle and compared to other previous studies [32,35], the upper bound was set to 200th cycle instead of the 100th cycle. The subsequent results were obtained using the Δm^{200} feature.

Through the strategy in Section 3.1, Feature construction, the HV and LV values were obtained as 2.82774 V and 2.05978 V. To evaluate the strategy for defining meaningful HV and LV points, each HV and LV value was subjected to some perturbations in the 0.1-0.2 V magnitude and the proposed features were derived and tested again. The prediction errors for the unperturbed HV and LV values are significantly smaller than those for the perturbed cases (Table 5); thus, the intention to reflect meaningful variations in the variance of ΔQd(V) from the range between HV and LV, has been properly implemented in the aforementioned strategy.

2. Cycle Life Prediction Results

Seven ML models were used to predict cycle life, and their corresponding cycle life prediction results are shown in Table 6 and Table 7.

The input feature set *'without Qd^{10}'* and the NN model of ML exhibit superior prediction performance. The prediction error of this model, which is 77.05 cycles (with an RMSE of 0.042 and MAPE

Table 4. Comparison of training and test error for cycle life prediction from proposed features and NN model. The effects of Δm^{100} and Δm^{200} are mainly compared

Features		Train		Test	
-		Δm^{100}	Δm^{200}	Δm^{100}	Δm^{200}
Δm^{n} single	RMSE	122.20 (0.097)	88.88 (0.069)	221.92 (0.128)	108.69 (0.068)
	MAPE	16.21	10.79	20.73	10.46
Qd^{10}, Qd^{100}, Qd^{200}, Δm^{200}	RMSE	69.85 (0.053)	65.11 (0.044)	177.59 (0.087)	103.22 (0.058)
	MAPE	8.35	7.16	11.52	9.83
Qd^{100}, Qd^{200}, Δm^{200}	RMSE	83.13 (0.059)	65.33 (0.046)	141.64 (0.065)	77.05 (0.042)
	MAPE	9.47	7.43	9.01	6.63

Table 5. Comparison of training and test error (RMSE, MAPE) for cycle life prediction obtained from perturbed HV (2.82774 V) and LV (2.05978 V) values. The case of 'LV-0.2V' was disregarded because it deviates from the region of smooth curvature on the discharge curve depicted in Fig. 1

Feature sets			HV LV	HV+0.1 LV	HV−0.1 LV	HV+0.1 LV+0.1	HV−0.1 LV+0.1	HV+0.2 LV	HV−0.2 LV	HV+0.2 LV+0.1	HV−0.2 LV+0.1
Train	*'with Qd^{10}'*	RMSE	65.11(0.044)	84.36(0.058)	86.46(0.050)	83.66(0.058)	79.87(0.054)	84.64(0.055)	75.21(0.045)	72.01(0.055)	80.85(0.051)
		MAPE	7.16	8.55	7.93	9.05	8.97	8.66	7.09	8.83	8.08
	'without Qd^{10}'	RMSE	65.33(0.046)	65.67(0.049)	69.24(0.049)	74.50(0.056)	67.49(0.050)	76.42(0.058)	75.97(0.053)	70.19(0.055)	74.99(0.056)
		MAPE	7.43	7.40	7.91	8.60	7.83	9.28	8.33	8.61	9.10
Test	*'with Qd^{10}'*	RMSE	103.22(0.058)	113.47(0.062)	116.71(0.057)	124.06(0.071)	123.56(0.071)	129.11(0.067)	125.78(0.063)	113.95(0.069)	123.72(0.062)
		MAPE	9.83	10.13	8.64	12.20	11.61	10.82	10.04	10.84	9.99
	'without Qd^{10}'	RMSE	77.05(0.042)	99.65(0.059)	101.16(0.053)	110.56(0.067)	97.75(0.056)	109.87(0.076)	113.21(0.069)	105.38(0.067)	114.17(0.077)
		MAPE	6.63	9.86	7.84	11.44	9.03	11.71	10.75	10.34	12.08

Table 6. Prediction errors (RMSE, MAPE) obtained from seven ML models, namely, (a) NN, (b) Enet, (c) GPR, (d) KRR, (e) SVR, (f) RFR, (g) XGB, and 6 input features and feature sets

Train		Var(ΔQd(V))	Δm^{200}	discharge	with Qd^{10}	without Qd^{10}	all features
NN	RMSE	118.34(0.096)	88.88(0.069)	68.27(0.058)	65.11(0.044)	65.33(0.046)	66.38(0.190)
	MAPE	15.22	10.79	7.77	7.16	7.43	7.11
Enet	RMSE	124.51(0.098)	96.30(0.081)	62.62(0.053)	61.93(0.047)	62.72(0.048)	53.67(0.040)
	MAPE	16.06	12.66	7.60	7.99	7.87	6.33
GPR	RMSE	119.83(0.096)	86.81(0.070)	36.67(0.032)	42.11(0.031)	44.41(0.033)	35.17(0.026)
	MAPE	25.48	13.54	7.01	7.34	6.71	3.94
KRR	RMSE	124.88(0.098)	98.74(0.081)	48.15(0.042)	52.73(0.038)	57.06(0.040)	19.65(0.014)
	MAPE	26.75	18.37	9.26	9.69	9.77	2.13
SVR	RMSE	116.75(0.094)	84.14(0.067)	87.39(0.071)	48.28(0.035)	50.03(0.036)	34.30(0.025)
	MAPE	22.42	11.56	5.20	5.65	14.88	2.55
RFR	RMSE	83.13(0.051)	71.56(0.037)	41.29(0.033)	71.68(0.032)	60.71(0.030)	69.01(0.033)
	MAPE	11.23	6.94	5.27	5.09	6.74	4.39
XGB	RMSE	24.87(0.017)	18.27(0.010)	5.14(0.005)	19.63(0.006)	19.97(0.007)	19.41(0.006)
	MAPE	3.38	2.27	0.67	1.01	0.67	0.52
Test		Var(ΔQd(V))	Δm^{200}	discharge	with Qd^{10}	without Qd^{10}	all features
NN	RMSE	122.57(0.090)	108.69(0.072)	145.48(0.116)	103.22(0.058)	77.05(0.042)	119.22(0.052)
	MAPE	15.74	10.46	13.70	9.83	6.63	10.24
Enet	RMSE	122.51(0.091)	119.39(0.070)	86.08(0.082)	89.80(0.048)	100.76(0.051)	127.17(0.054)
	MAPE	15.63	10.73	7.48	7.77	10.76	8.46
GPR	RMSE	119.82(0.089)	114.41(0.065)	107.05(0.103)	90.59(0.052)	87.20(0.049)	109.15(0.046)
	MAPE	25.62	19.23	14.54	13.60	12.07	7.1
KRR	RMSE	123.15(0.091)	119.12(0.070)	97.81(0.110)	99.50(0.052)	103.41(0.052)	149.98(0.099)
	MAPE	24.96	20.96	15.48	14.42	11.95	12.67
SVR	RMSE	117.04(0.088)	109.29(0.067)	92.55(0.106)	89.80(0.059)	87.88(0.059)	131.11(0.053)
	MAPE	22.42	19.86	19.37	19.19	19.60	6.64
RFR	RMSE	160.46(0.110)	136.82(0.083)	187.08(0.139)	137.45(0.065)	119.14(0.066)	179.60(0.074)
	MAPE	27.97	23.89	18.68	18.97	29.45	9.52
XGB	RMSE	174.22(0.126)	118.67(0.083)	160.11(0.134)	109.59(0.069)	102.42(0.066)	105.67(0.070)
	MAPE	37.07	25.53	19.84	20.22	31.58	10.14

Table 7. Train and test results summary comparing main features and feature set from this study and the benchmark study [21]. The listed metrics are the best cases in Table 6, corresponding to each feature and feature set. Both proposed feature and feature set showed better performance than the benchmark results. The best performance (error of 77.05 cycles) was 10.49% improved compared to the best result of benchmark (error of 86.08 cycles)

			Train	Test
Single feature	Var(ΔQd(V))	RMSE	124.51 (0.098)	122.51 (0.091)
		MAPE	16.06	15.63
	Δm^{200}	RMSE	88.88 (0.069)	108.69 (0.072)
		MAPE	10.79	10.49
Feature set	*'discharge'* set	RMSE	62.62 (0.053)	86.08 (0.082)
		MAPE	7.60	7.48
	'without Qd^{10}' set	RMSE	65.33 (0.046)	77.05 (0.042)
		MAPE	7.43	6.63

of 6.63%), is significantly better than the best performance reported in the benchmark study, which is 86 cycles.

Table 7 summarizes the best results from our study and the benchmark study. The *'without Qd^{10}'* feature set improved the bench-

Table 8. Summary of literature survey on studies using the same battery dataset and focusing on the early prediction of RUL. The proposed feature sets are listed at the top

References	Prediction errors		Method and used cycles
-	Cycles	MAPE	-
'without Qd^{10}' (Δm^{200})	77.05	6.63	Proposed method with NN (200 cycles)
'without Qd^{10}' (Δm^{100})	141.64	9.01	Proposed method with NN (100 cycles)
Severson et al. [21]	86	10.1	Enet model (100 cycles)
Yang et al. [34]	82.79	~7	Gradient boosting regression tree (GBRT) model (From 30 to 100-250 cycles)
Xu et al. [32]	119.62	-	Stacked Denoising Autoencoder (SDAE) method (300 cycles)
Fei et al. [35]	115	-	Support vector machine (SVM) model (100 cycles)
Zhou et al. [36]	-	8.6	Hierarchical Bayesian linear model (100 cycles)
Alipour et al. [27]	152	8.2	Hybrid model of Linear support vector regression (LSVR) and Gaussian process regression (GPR) (100 cycles)
Tang et al. [28]	145.70	13.40	PCLN model with convolutional neural network (CNN) and long short-term memory (LSTM) (100 cycles)
Ansari et al. [29]	12	-	Hybrid model of recurrent neural network (RNN) and particle swarm optimization (PSO) (Entire data used)
Zhou et al. [37]	64	4.557	Res-CNN model (First 30% of entire cycle)
Lin and Chai [38]	41	-	Two phase Wiener process with measurement errors (TPWPME) degradation model (First 30% of entire cycle)

mark results by 10.49%. Moreover, but even the proposed feature Δm^{200} itself demonstrated a satisfactory performance. The benchmark study reported a prediction error of 138 cycles for the single feature Var(ΔQd(V)). However, across the seven ML methods, the errors of single Δm^{200} are all smaller than that, with the smallest error being 108 cycles, a 21.74% improvement.

Table 8 confirms that our proposed feature set remains competitive compared to other relevant studies. Since many studies utilized features extracted from the first 100 cycles for RUL prediction, we included our results using the Δm^{100} feature based on the first 100 cycles for comparison. With an RMSE of 141.64 cycles and MAPE of 9.01%, our Δm^{100} feature's performance is comparable to other studies' results, including the benchmark. Our proposed method also demonstrated strong performance against more specific deep learning models, such as CNN, RNN, LSTM, and their hybrid models.

While the performance of our proposed method is slightly inferior to those of Ansari et al. [29], Zhou et al. [37], and Lin and Chai [38], it is worth noting that models such as CNN and RNN typically have more model parameters than vanilla NN. These models require a large amount of input data to perform balanced training without overfitting issue, as well as considerable computing resources during the learning and calculation processes. Therefore, from an efficiency standpoint, the proposed method may be considered preferable.

With only the exception of Enet and KRR, our proposed feature sets show more accurate prediction (Fig. 6). In particular, *'without Qd^{10}'* set frequently showed smaller test errors than *'with Qd^{10}'* set.

Since, the 10th cycle is so close to the unstable formation cycle period, the Qd^{10} feature can contain some irregular volatility, which is not very informative for cycle life prediction, and lowers the prediction performance of *'with Qd^{10}'* set. Consistently, as shown in

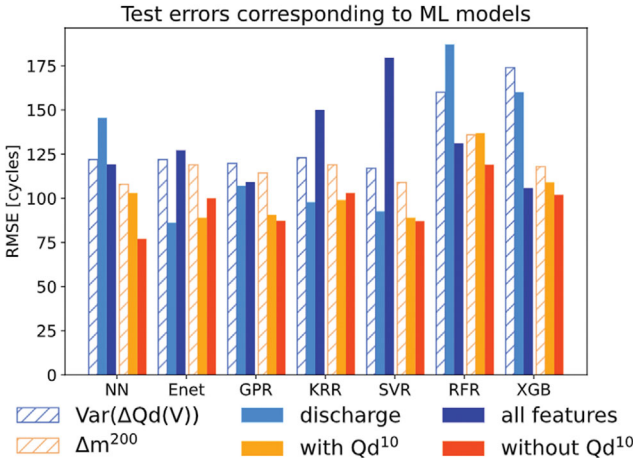

Fig. 6. Error plots of six input features in seven ML models, obtained from test data. The proposed features are colored in orange series, and the others are colored in blue series. Single features, namely Δm^{200} and Var(ΔQd(V)), are indicated as shaded bars distinctive to other feature sets. The orange bars generally show fewer errors than the blue bars, demonstrating that the proposed features have superior predictive ability and robustness in comparison to previously studied features.

Fig. 7 in Section 4.4.1, Feature correlation analysis, the Qd^{10} feature shows weaker correlation than others. Similarly, the *'all features'* set showed poor predictions on test data compared to training results. Further investigation revealed that the features included in the *'all features'* set, namely Skew(ΔQd(V)), Kurt(ΔQd(V)), Qd(cycle=2), and $\max_{cycle}$Qd(cycle)−Qd(cycle=2), were weakly correlated with the cycle life that we aimed to predict. These insignificant predictors increased the number of model parameters, affected almost irrelevant variation on model tuning and prediction, and

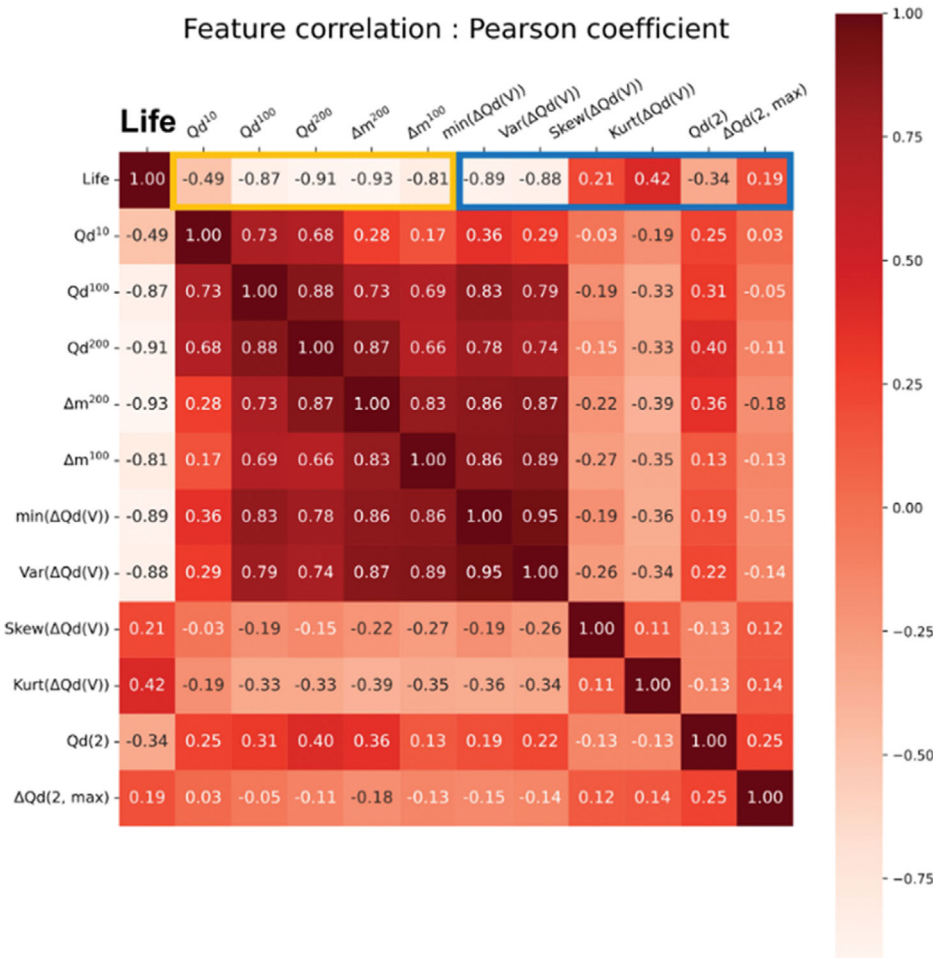

Fig. 7. Pearson correlation coefficient heatmap for all the features considered in this study. The closer the absolute value of the coefficient, ρ, is to 1, the stronger the correlation. The features in the orange box on the top are the proposed features in this study, while the blue box contains the *'discharge'* set [21]. Qd(cycle=2) and $\max_{cycle}$Qd(cycle)−Qd(cycle=2) features are shortly indicated as Qd(2) and ΔQd(2, max). The correlation coefficient values for all combinations of ten features are calculated after taking their logarithms.

led to overfitting issues [48,49]. Therefore, the fact that the *'without* Qd^{10}*'* set consisting of only three highly correlated factors, showed the best performance suggests that successful feature selection was executed.

3. Quantitative Analyses on Efficiency Improvement

A quantitative analysis of efficiency improvement, the most emphasized point in this study, was also conducted. The main feature of the benchmark study, Var(ΔQd(V)), and the Δm feature of this study were compared in terms of the time and capacity consumed for feature computation, on 43 test datasets. See Supplementary Note 3 for more detailed measurement conditions.

For the benchmark feature, the feature computation time for each of the 43 test datasets showed an average of 6.28E-3 seconds, with a maximum of 1.32E-2 seconds and a minimum of 3.38E-3 seconds. However, the method proposed in this study demonstrated computing speed approximately 63.72 times faster than the benchmark feature, as evidenced by an average computation time of 9.86E-5 seconds, with a maximum of 1.88E-4 seconds and a minimum of 8.23E-5 seconds across the 43 test datasets.

Furthermore, storing all cycle data necessary for benchmark feature computation required 1,633,229 bytes of storage capacity when stored in a pickle format, which is the same as the public dataset. However, storing only information for HV and LV points at 10^{th} and 100^{th} cycles in line with the proposed method resulted in memory usage of just 33,558 bytes, demonstrating an approximately 48.67 times higher efficiency in memory management.

Considering only the aspects related to feature calculation processes, the proposed method seems sufficiently effective, making it more desirable in on-board situations where computing resources are limited.

4. Discussions on the Proposed Features

4-1. Feature Correlation Analysis

Fig. 7 shows the linear (Pearson) correlations between all the features, including our proposed features and previously studied features. Recall that the feature sets including Δm^{200} demonstrated better prediction performance than *'discharge'* set related ones, as shown in Table 6. The features we propose consistently show stronger correlations with the battery cycle life, and the Δm^{200} feature shows the strongest correlation of $\rho \approx -0.93$. Although the Var(ΔQd(V)) feature in *'discharge'* set used in this study contains a reproduction

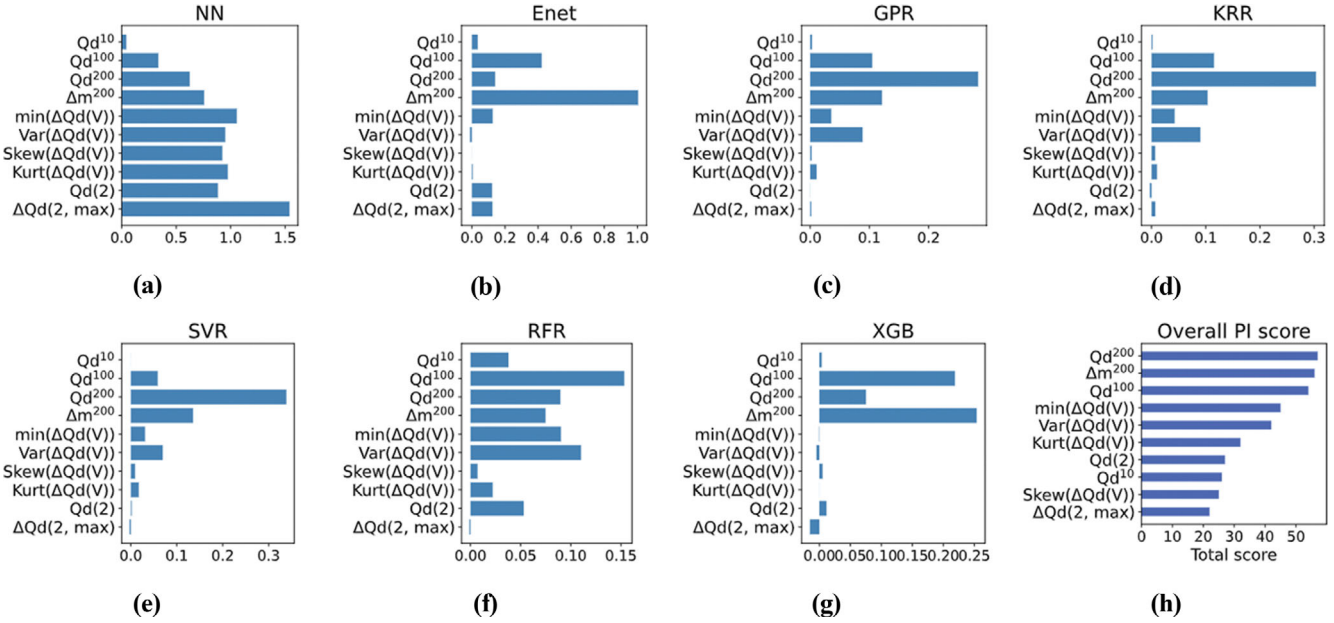

Fig. 8. Comparison of PI scores for each proposed feature, when they are applied to ML models ((a) NN, (b) Enet, (c) GPR, (d) KRR, (e) SVR, (f) RFR, (g) XGB). (h) shows overall PI scores combined from (a)-(g) results

error of about 4.3%, the Δm^{200} still indicates higher correlation than its original value of ρ=−0.92 [21] (see Supplementary Data, Fig. S2, for more details).

The correlations between Var(ΔQd(V)) and Δm^{200} and between Var(ΔQd(V)) and Δm^{100} are ρ=−0.87 and ρ=−0.89, which are also quite high. It clearly supports that the feature construction strategy in Section 3.1, Feature construction, to make the Δm features imply the variance of Qd(V), works well.

Among the *'discharge'* set features, the min(ΔQd(V)) and Var (ΔQd(V)) the features exhibit high values of ρ=−0.89 and ρ=−0.88, respectively. Because the values of kurtosis and skewness are quite small, they seem less informative regarding cycle life. This is also consistent with the result that the *all features* case was not excellent in prediction performance even though it contained the richest input data.

4-2. Feature Importance Analysis

The contribution of the features to the prediction accuracy was accessed by PI scores resulting from each of the seven ML models (Fig. 8).

Some negative PI scores observed in Fig. 8 indicate that the prediction performance is rather improved compared to the reference score by random permutation on some features. Such features that heavily depend on the randomness can be regarded as insignificant in the prediction process.

For the ten tested features, a score from 1 to 10 was assigned. The higher the score, the greater the significance in the prediction process. As shown in Fig. 8(h), when the results from seven ML models were combined, the top five features were ranked in the order of Qd^{200}, Δm^{200}, Qd^{100}, min(ΔQd(V)), and Var(ΔQd(V)) from the highest score.

In Fig. 7, the Pearson correlation coefficients between each feature and cycle life are listed in the order of Δm^{200} (ρ=−0.93), Qd^{200} (ρ=−0.91), min(ΔQd(V)) (ρ=−0.89), Var(ΔQd(V)) (ρ=−0.88), and Qd^{100} (ρ=−0.87) from the highest magnitude. Despite a minor reordering, the top five features were maintained. The fact that features with high Pearson coefficients also had high PI scores provides strong evidence that these features play a crucial role in predicting battery cycle life. Above all, Δm^{200} and Qd^{200} features consistently take the lead among them and seem to have predominant potential to interpret the battery cycle life.

CONCLUSIONS

We developed a new feature, Δm, to predict the end of LIB's cycle life. We focused on the variance of ΔQd(V) and the time-varying area under the discharge curve, which is linearly related to the energy dissipation of the battery. The region where the dramatic variation of the variance of ΔQd(V) appears is captured concentrically by selecting the points at which the cycle-to-cycle change of the discharge curve is the largest and smallest, and applying the mean value theorem between the two points. This pointwise approach requires only two data points on each cycle's discharge curve for measuring the variation of the ΔQd(V), which leads to a significant reduction in computational cost and data memory demand. There is no need to store whole cycle data, interpolate entire curve, and calculate complex statistical indicators using all of the sampled data. The advantage of being able to perform prediction in the early cycle below 200^{th} cycles along with this simplicity will contribute to on-board estimation in actual driving as well as test session in battery manufacturing. The proposed features were tested with seven ML models, NN, Enet, GPR, KRR, SVR, RFR, and XGB, and showed a 21.74% improvement in predictive performance compared to the benchmark result. Finally, through Pearson correlation and PI score analyses, it was qualitatively confirmed that the proposed features are not only convenient but also have close correlations with the meaning of the variance of ΔQd(V) and

the battery cycle life. The shrinking behavior of the discharge curve, which is the main focus in this study, is a general characteristic not only to LIBs but also to other types of batteries. Thus, the proposed features based on the discharge curve's variation are expected to be widely used for cycle life evaluation of other batteries.

Further research should consider the disturbance which can be encountered in the real operation of LIBs in EVs. Since this study is based on experimental data collected under controlled conditions [21], the performance of suggested features can differ in the presence of stochastic noise. The verification and supplement of the proposed feature construction scheme against potential noise are desirable for future work.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) (NRF-2016R1A5A1009592). The Institute of Engineering Research at Seoul National University provided research facilities for this work.

ABBREVIATIONS

BMS : battery management system
CC : constant current
CNN : convolutional neural network
DNN (NN) : deep neural network
Enet : elastic net
EV : electric vehicle
GB : gradient boosting
GPR : gaussian process regression
KRR : kernel ridge regression
LFP : lithium iron phosphate
LIB : lithium-ion battery
LSTM : long short term memory
MAPE : mean absolute percentage error
ML : machine learning
NB : naïve bayes
PCA : principal component analysis
PI : permutation importance
PSO : particle swarm optimization
Qd : discharge capacity
RFR : random forest regression
RMSE : root mean square error
RNN : recurrent neural networks
RUL : remaining useful life
SDAE : stacked denoising autoencoder
SEI : solid electrolyte interphase
SOH : state of health
SVR : support vector regression
TPWPME : two phase wiener process with measurement errors
XGB : extreme gradient boosting

SUPPORTING INFORMATION

Additional information as noted in the text. This information is available via the Internet at http://www.springer.com/chemistry/journal/11814.

REFERENCES

1. L. Cai, J. Meng, D.-I. Stroe, G. Luo and R. Teodorescu, *J. Power Sources*, **412**, 615 (2019).
2. D. E. Newton, Fracking: A Reference Handbook: A Reference Handbook, ABC-CLIO (2015).
3. J. Lee, D. Lee, C. B. Shin, S.-Y. Lee, S.-M. Oh, J.-J. Woo and I.-C. Jang, *J. Korean Inst. Chem. Eng.*, **59**, 393 (2021).
4. L. Gaines, *Sustainable Mater. Technol.*, **1**, 2 (2014).
5. X. Lu, K. Sun, J. M. Guerrero, J. C. Vasquez and L. Huang, *IEEE Trans. Ind. Electron.*, **61**, 2804 (2013).
6. X. Peng and Y. Li, *J. Electrochem. Energy Convers. Storage*, **19**, 3 (2021).
7. D. Wang, F. Yang, Y. Zhao and K.-L.Tsui, *Microelectron. Reliab.*, **78**, 212 (2017).
8. S. S. Ng, Y. Xing and K. L. Tsui, *Appl. Energy*, **118**, 114 (2014).
9. S. S. Mansouri, P. Karvelis, G. Georgoulas and G. Nikolakopoulos, *IFAC-PapersOnLine*, **50**, 4727 (2017).
10. S. C. Nagpure, B. Bhushan and S. Babu, *J. Electrochem. Soc.*, **160**, A2111 (2013).
11. D. E. Demirocak and B. Bhushan, *J. Power Sources*, **280**, 256 (2015).
12. X. Han, M. Ouyang, L. Lu, J. Li, Y. Zheng and Z. Li, *J. Power Sources*, **251**, 38 (2014).
13. P. Arora, R. E. White and M. Doyle, *J. Electrochem. Soc.*, **145**, 3647 (1998).
14. X. Xu, C. Yu, S. Tang, X. Sun, X. Si and L. Wu, *IEEE Access*, **7**, 105186 (2019).
15. X. Shu, G. Li, J. Shen, W. Yan, Z. Chen and Y. Liu, *J. Power Sources*, **462**, 228132 (2020).
16. M. Doyle, T. F. Fuller and J. Newman, *J. Electrochem. Soc.*, **140**, 1526 (1993).
17. P. Arora, M. Doyle and R. E. White, *J. Electrochem. Soc.*, **146**, 3543 (1999).
18. X. Li, Y. Ma and J. Zhu, *Measurement*, 109935 (2021).
19. D. Song, C. Sun, Q. Wang and D. Jang, *Energy Power Eng.*, **10**, 10 (2018).
20. L. Cai, J. Meng, D.-I. Stroe, G. Luo and R. Teodorescu, *IEEE Trans. Power Electron.*, **35**, 11855 (2020).
21. K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggedakis, M. Z. Bazant, S. J. Harris, W. C. Chueh and R. D. Braatz, *Nat. Energy*, **4**, 383 (2019).
22. L. Ren, L. Zhao, S. Hong, S. Zhao, H. Wang and L. Zhang, *IEEE Access*, **6**, 50587 (2018).
23. Y. Li, C. Zou, M. Berecibar, E. Nanini-Maury, J. C.-W. Chan, P. Van den Bossche, J. Van Mierlo and N. Omar, *Appl. Energy*, **232**, 197 (2018).
24. D. Yang, X. Zhang, R. Pan, Y. Wang and Z. Chen, *J. Power Sources*, **384**, 387 (2018).
25. Y. Yang, *Appl. Energy*, **292**, 116897 (2021).
26. J. Liu, A. Saxena, K. Goebel, B. Saha and W. Wang, Defense Technical Information Center (2010).
27. M. Alipour, S. S. Tavallaey, A. M. Andersson and D. Brandell,

ChemPhysChem, **23**, e202100829 (2022).
28. Y. Tang, K. Yang, H. Zheng, S. Zhang and Z. Zhang, *Measurement*, **199**, 111530 (2022).
29. S. Ansari, A. Ayob, M. H. Lipu, A. Hussain and M. H. M. Saad, *J. Energy Storage*, **56**, 106050 (2022).
30. X.-Y. Yao, G. Chen, M. Pecht and B. Chen, *J. Energy Storage*, **58**, 106437 (2023).
31. P. K. R. Maddikunta, G. Srivastava, T. R. Gadekallu, N. Deepa and P. Boopathy, *IET Intelligent Transport Systems*, **14**, 1388 (2020).
32. F. Xu, F. Yang, Z. Fei, Z. Huang and K.-L.Tsui, *Reliab. Eng. Syst. Saf.*, **208**, 107396 (2021).
33. Z. Tong, J. Miao, S. Tong and Y. Lu, *J. Cleaner Prod.*, **317**, 128265 (2021).
34. F. Yang, D. Wang, F. Xu, Z. Huang and K.-L. Tsui, *J. Power Sources*, **476**, 228654 (2020).
35. Z. Fei, F. Yang, K.-L. Tsui, L. Li and Z. Zhang, *Energy*, **225**, 120205 (2021).
36. Z. Zhou and D. A. Howey, arXiv:2211.05697 (2022).
37. Z. Zhou, Y. Liu, M. You, R. Xiong and X. Zhou, *Green Energy. Intell. Transp.*, **1**, 100008 (2022).
38. W. Lin and Y. Chai, *Meas. Sci. Technol.*, **34**(5), 055018 (2023).
39. M. Kabir and D. E. Demirocak, *Int. J. Energy Res.*, **41**, 1963 (2017).
40. D. Aurbach, E. Zinigrad, Y. Cohen and H. Teller, *Solid State Ionics*, **148**, 405 (2002).
41. J. Vetter, P. Novák, M. R. Wagner, C. Veit, K.-C. Möller, J. Besenhard, M. Winter, M. Wohlfahrt-Mehrens, C. Vogler and A. Hammouche, *J. Power Sources*, **147**, 269 (2005).
42. C. R. Birkl, M. R. Roberts, E. McTurk, P. G. Bruce and D. A. Howey, *J. Power Sources*, **341**, 373 (2017).
43. P. Ramadass, B. Haran, P. M. Gomadam, R. White and B. N. Popov, *J. Electrochem. Soc.*, **151**, A196 (2004).
44. A. Mukhopadhyay and B. W. Sheldon, *Prog. Mater. Sci.*, **63**, 58 (2014).
45. M. Palanisamy, M. H. Parekh and V. G. Pol, *Adv. Funct. Mater.*, **30**, 2003668 (2020).
46. Z. M. Salameh, M. A. Casacca and W. A. Lynch, *IEEE Trans. Energy Convers.*, **7**, 93 (1992).
47. B. Paxton and J. Newman, *J. Electrochem. Soc.*, **144**, 3818 (1997).
48. D. M. Hawkins, *J. Chem. Inf. Comput. Sci.*, **44**, 1 (2004).
49. X. Ying, *J. Phys.: Conf. Ser.*, **1168**, 022022 (2019).

Supporting Information

Feature construction for on-board early prediction of electric vehicle battery cycle life

Junseop Shin*, Yeonsoo Kim,†, and Jong Min Lee*,†**

*School of Chemical and Biological Engineering, Institute of Chemical Processes, Seoul National University, Gwanak-ro 1, Gwanak-gu, Seoul 08826, Korea
**Department of Chemical Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 01897, Korea
(*Received 22 December 2022 • Revised 7 April 2023 • Accepted 17 April 2023*)

SUPPLEMENTARY DATA

The Δm^{200} and the Δm^{100} features were calculated and scattered with their cycle life on a log scale as shown in Fig. S2. In this study, Var(ΔQd(V)) feature used in the benchmark study [1] was reproduced and compared in Fig. S2(a).

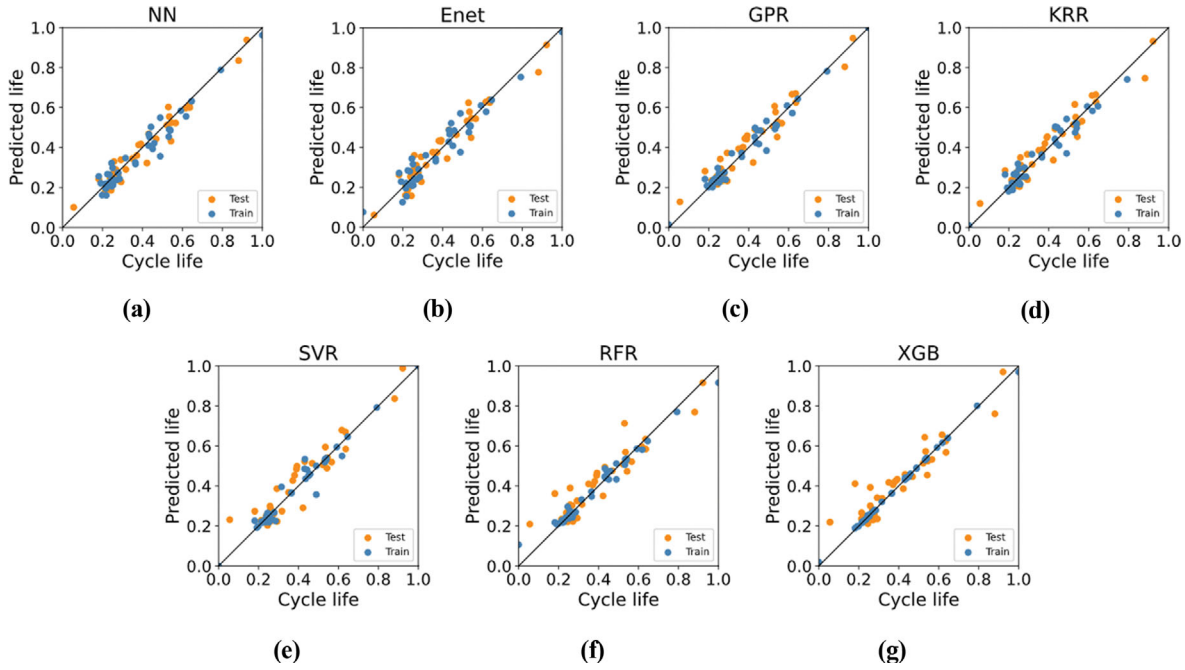

Fig. S1. Regression plots of the best prediction results, obtained through various ML methods and *'without Qd^{10}'* set.

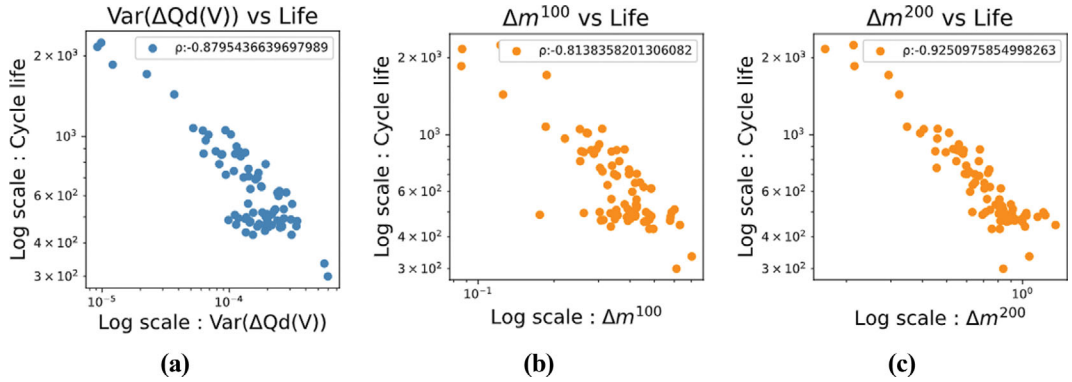

Fig. S2. Linear correlations between features and battery cycle life: (a) Var(ΔQd(V)) vs cycle life, (b) Δm^{100} vs cycle life, (c) Δm^{200} vs cycle life. Both x and y axis are on the log scales.

Table S1. The six features make up the *'discharge'* set [1]. They are reproduced in this study and compared with the proposed features including Δm. The subscript *i* indicates each experimental battery batch and the bar symbol over a letter indicates an average value found in its distribution

Features	Configurations
Minimum	$\min(\Delta Qd(V))$
Variance	$\left\lvert\frac{1}{N-1}\Sigma_{i=1}^{N}(\Delta Qd(V)_i-\overline{\Delta Qd(V)})^2\right\rvert$
Skewness	$\left\lvert\frac{\frac{1}{N}\Sigma_{i=1}^{N}(\Delta Qd(V)_i-\overline{\Delta Qd(V)})^3}{\sqrt{\Sigma_{i=1}^{N}(\Delta Qd(V)-\overline{\Delta Qd(V)})^2}^3}\right\rvert$
Kurtosis	$\left\lvert\frac{\frac{1}{N}\Sigma_{i=1}^{N}(\Delta Qd(V)_i-\overline{\Delta Qd(V)})^4}{\left(\frac{1}{N}\Sigma_{i=1}^{N}(\Delta Qd(V)-\overline{\Delta Qd(V)})^2\right)^2}\right\rvert$
Discharge capacity at 2th cycle	$Qd(cycle=2)$
Max discharge capacity - discharge capacity at 2th cycle	$\max_{cycle}Qd(cycle)-Qd(cycle=2)$

Each of the three features has a negative correlation with the cycle life, as indicated by the minus sign. All of them showed absolute ρ values greater than 0.8, which indicates quite strong linear correlations to the cycle life when taking the log on both axis.

The ρ values in Fig. S2(a), Fig. S2(b) and Fig. S2(c) represent the Pearson correlation coefficients, which illustrate the linear relationships between two factors. The ρ values are −0.88, −0.81, and −0.93, respectively. The original ρ value of Var(ΔQd(V)) was −0.92 [1], but an error of about 4.3% was included in the process of reproducing from the raw data. The analyses of the results are carried out taking this into account.

The six referenced benchmark features are listed in Table 1, that compose the *'discharge'* set [1]. Based on the between 10th and 100th cycles, this feature set demonstrated the highest predictive accuracy in the benchmark study [1]. These types of statistical features are also utilized in several other studies [2,3], so they were considered as a reliable benchmark for evaluating the performance of new features proposed in this study.

SUPPLEMENTARY NOTES

1. Settings of Machine Learning Models

1-1. Deep Neural Network

A deep neural network (NN) maps input data into output data through calculations on its hidden layers. Once a network structure is fixed and training data is selected, a parameter such as weight and biases on each node is optimized to perform proper mapping. The NN can work as a surrogate model which can describe complex, nonlinear tendencies inside the given data, which are difficult to be defined numerically. It has recently been in the spotlight in the fields of machine learning and data science, and has been further used in stochastic modeling, offline simulation, classification, optimization, and prediction [4,5]. In this study, the data-driven approach based on NN was thought to be the most efficient prediction tool due to its ability to describe complex and nonlinear correlations between features and the battery cycle life.

The number of hidden layers and nodes were set heuristically. Just one hidden layer is often enough for a domain problem which can be defined by standard nonlinear equation [6]. Since there exist some numerical models of the LIB system defined already [7], the number of hidden layer do not considered to be large. In addition, the more nodes and layers, the more computational cost and overfitting problems could occur; thus trial and error tests for 1 to 4 layered structures were conducted to find best number of layers, which are not too large compared to the input data dimension. Since the dimension of our feature sets are totally under 10, the number of nodes was tested in top down from the 10 nodes case, and the best case was selected. For the same reason, L1 regularization was also adopted with properly tuned coefficient. The activation function also selected as ReLU by trial and error. To prevent collapsing issue where most of the weight distribution goes to zero, the *he-normal* initialization was introduced: $w\sim N\left(0, \sqrt{\frac{2}{n_{in}}}\right)$, where n_{in} is the number of nodes in the front layer, and w is the weight of each layer. Training epoch was set to 1000, loss function and optimizer were set to mean squared error (MSE) and root mean square propagation (RMSprop), respectively. Numerical hyperparameters such as regularization coefficient, regularization ratio, and learning rate were selected by grid searching at 10-scale intervals.

1-2. Elastic Net

Among the ML models used in this study, elastic net (Enet) only performs linear regression. Let the input data be x and output be y, the following formulation holds,

$$\hat{y}_i=\hat{w}_j^T x_i+b, \tag{1}$$

where $\hat{y}_i$ means predicted output value, x_i means input feature vector, and $\hat{w}$ means matrix form of linear model's coefficient.

Enet basically minimizes the residual sum of error squares (RSS) between y_{label} and $\hat{y}$, and Ridge and Lasso regulation terms are added to prevent overfitting. The ratio of the two regulation terms can be adjusted by the weighting coefficients λ_1 and λ_2. As a result, the given regression problem is formulated as follows,

$$\min_{\hat{w}_1,\, i=1,\,\ldots,\,n}\left\{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\lambda_1\sum_{i=1}^{n}\hat{w}_i^2+\lambda_2\sum_{i=1}^{n}|\hat{w}_i|\right\},$$

$$=\min_{\hat{w}_1,\, i=1,\,\ldots,\,n}\{||y_i-\hat{y}_i||_2^2+\lambda_1||\hat{w}_i||_2^2+\lambda_2||\hat{w}_i||_1\}. \quad (2)$$

λ_1 and λ_2 can be manipulated through hyperparameters α and *l*1_*ratio*, where $\alpha=\lambda_1+\lambda_2$ and *l*1_*ratio*$=\frac{\lambda_1}{\lambda_1+\lambda_2}$. Usually, Enet is known to show better performance compared to vanilla ridge and lasso regressions, which use only regulation of either L1 or L2, when there exist higher linear correlations between features and more abundant training data.

1-3. Gaussian Process Regression

Gaussian process regression (GPR) finds out the distribution of a function by considering each parameter as a probability density function, rather than finding out deterministic parameters that make up the function like Enet. Thus, mean function and covariance function describe the distribution of function parameters in GPR. Kernel function K(x, x) with input variable vector x, and gaussian noise of ε~N(0, σ^2) define prior distribution of observed output variable y as follows,

$$y\sim N(0,\, K(x,\, x)+\sigma^2 I), \quad (3)$$

where I represents unit matrix. By maximum likelihood method, hyperparameters such as σ and parameters contained in the kernel function are tuned to maximize the log marginal likelihood (LML) using gradient based methods in common. For every step, kernel function and its hyperparameters are tuned, and the way they are tuned has a significant impact on the posterior prediction [8]. In this study, the *GaussianProcessRegressor* of *scikit-learn* package in *python* was used to implement above steps [9]. Since LML can fall into local optima, hyperparameter *n_restarts_optimizer* is set to repeat tuning process to reach the global optima. Because the appropriate kernel function or noise level varies depending on the system and the *n_restarts_optimizer* value does not have to repeat too much, the optimal settings were found through some trials. Here, we chose the Matérn 5/2 kernel defined as follows, which showed good performances across the board. For the two points apart from distance unit d, Matérn kernel defines their covariance as

$$C_\nu(d)=\sigma^2\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{d}{\rho}\right)^{\nu}K_\nu\left(\sqrt{2\nu}\frac{d}{\rho}\right), \quad (4)$$

where Γ is the gamma function, K_ν indicates the modified Bessel function of the second kind, ρ and v are covariance parameters. In this Matérn 5/2 case, v=5/2.

1-4. Kernel Ridge Regression

Kernel ridge regression (KRR) also uses kernel function like GPR to find out target function. KRR maps nonlinear functions from the original space to some spaces where they are transformed into linear functions by kernel tricks (ϕ=K(X)). For the mapped variable ϕ, the following minimization problem of the loss function is performed in grid search approach [10].

$$\min_{\hat{w}_1,\, i=1,\,\ldots,\,n}\left\{\sum_{i=1}^{n}\left(y_i-\left(\sum_{i=1}^{n}\hat{w}_J^T\phi+b\right)\right)^2+\lambda\sum_{i=1}^{n}\hat{w}_i^2\right\},$$

$$=\min_{\hat{w}_1,\, i=1,\,\ldots,\,n}\left\{\sum_{i=1}^{n}\left(y_i-\left(\sum_{i=1}^{n}\hat{w}_J^T K_j(x_i)+b\right)\right)^2+\lambda\sum_{i=1}^{n}\hat{w}_i^2\right\}. \quad (5)$$

The loss function is similar to that of the ridge regression, where only the L2 regulation is added to the RSS term. The regulation strength hyperparameter λ was tuned with proper kernel selection. For other parameters consisting the kernel itself, such as gamma, default value of the *KernelRidge* package in *scikit-learn* of *python* were used.

1-5. Support Vector Regression

Although support vector regression (SVR) is very similar to KRR in that it uses kernel tricks to map nonlinear functions into linear form, there is little difference that KRR is more similar to Ridge regression while SVR takes epsilon-intensive loss function. In epsilon-intensive loss function, errors in ε distance are treated as equal to zero and only the errors beyond the ε boundary are used for learning. When ε becomes sufficiently small, it takes the form similar to the loss function of KRR [11,12].

$$\min_{\hat{w}_1,\, i=1,\,\ldots,\,n}\left\{C\sum_{i=1}^{n}(\zeta_i^-+\zeta_i^+)+\sum_{i=1}^{n}\hat{w}_i^2\right\}, \quad (6a)$$

subject to

$$\zeta_i^-\geq y_i-\left(\sum_{j=1}^{n}\hat{w}_J^T K_j(x_i)+b\right)-\varepsilon, \quad (6b)$$

$$\zeta_i^+\geq\left(\sum_{j=1}^{n}\hat{w}_J^T K_j(x_i)+b\right)-y_i-\varepsilon, \quad (6c)$$

$$\hat{w}_J,\ \zeta_i^-,\ \zeta_i^+\geq 0, \quad (6d)$$

where $\zeta_i^{+/-}$ is slack variables meaning the amount of positive/negative error over the intensive error criterion, ε.

Especially, when we assume $\varepsilon\approx 0$,

$$\zeta_i^-+\zeta_i^+=\left|y_i-\sum_{j=1}^{n}\hat{w}_J^T K_j(x_i)-b-\varepsilon\right|,$$

$$\approx\left|y_i-\sum_{j=1}^{n}\hat{w}_J^T K_j(x_i)-b\right|, \quad (7)$$

resulting in the $\hat{w}$ written as below,

$$\hat{w}=\mathrm{argmin}_{\hat{w}_1,\, i=1,\,\ldots,\,n}\left\{C\sum_{i=1}^{n}\left|y_i-\sum_{j=1}^{n}\hat{w}_J^T K_j(x_i)-b\right|+\sum_{i=1}^{n}\hat{w}_i^2\right\}. \quad (8)$$

Likewise, the proper kernel function was selected and hyperparameter such as regulation parameter C, which indicates inverse protection to regularization strength, and the value of ε are tuned. Kernel dependent hyperparameters such as *gamma*, which determines the extent to which a data has an influence, or order of polynomial kernel were set to the default value of the *SVR* package of *python*, *scikit-learn*.

1-6. Random Forest Regression & Extreme Gradient Boosting

Random forest regression (RFR) and extreme gradient boosting (XGB) are kinds of ensemble methods that use multiple models for learning. Ensemble methods largely include bagging and boosting. In bagging, a number of models are learned from the bootstrapped data subsets respectively. The results obtained by parallel

computing from each independent model are aggregated and their average value is used as the final result. Boosting, on the other hand, is a method in which model learning is sequentially performed. The result of the previous model adjusts weight for the next learning. By putting a high weight on the wrong answer and a low weight on the correct answer, the focus goes on the wrong answer to be improved. Boosting is known to have fewer errors than bagging, but is little bit slower and likely to have overfitting problem [13]. RFR and XGB are ML methods using bagging and boosting respectively. Some of their hyperparameters are also tuned by grid search on some intervals selected empirically, and some other default values from *RandomForestRegressor* and *xgboost* package of *scikit-learn*, *python* were used.

Similar to the NN setting, the categorical inputs were tested trial and error, and the numerical hyperparameters were grid searched at 10-scale intervals in the proper range. Then the hyperparameters are tuned to the value which showed the best performance.

2. Feature Analysis Methods

2-1. Feature Correlation Analysis

Pearson correlation was used to analyze linear correlations. Pearson correlation coefficient r_{XY} is a measurement of the linear relationship between any two probabilistic variables, defined by the following formula,

$$r_{XY}=\frac{\frac{\Sigma_{i=1}^{N}(X_i-X)(Y_i-Y)}{N-1}}{\sqrt{\frac{\Sigma_{i=1}^{N}(X_i-X)^2}{N-1}}\sqrt{\frac{\Sigma_{i=1}^{N}(Y_i-Y)^2}{N-1}}}=\frac{\Sigma_{i=1}^{N}(X_i-X)(Y_i-Y)}{\sqrt{\Sigma_{i=1}^{N}(X_i-X)^2}\sqrt{\Sigma_{i=1}^{N}(Y_i-Y)^2}} \quad (9)$$

Covariance can represent the form and the distribution between two probabilistic variables. However, since the value depends on the units of each individual variable, it can have multiple values for the same dataset. Therefore, it is difficult to clearly represent the degree of association between the two variables with the covariance value alone. By dividing the covariance into the standard deviations corresponding to the two variables, it normalized into the range of [−1, +1], regardless of the units of each variable. This is the pearson correlation coefficient described in (9). The absolute magnitude of this coefficient means the intensity of correlation, and the +/− sign means the direction of linearity. The closer the magnitude is to 1, the stronger the linearity between the two variables. If $r_{XY}=0$, they have no linear correlation.

2-2. Feature Importance Analysis

In order to identify features that have dominant effects through ML models, feature importance analysis was conducted using permutation feature importance.

It is a method of measuring how much the predictive results change when the individual input feature vectors that make up the whole feature set are random-shuffled compared with the original feature set. The problem dealing with in this study is a kind of regression, therefore the R-squared (R^2) was used as the score metric.

$$R^2=1-\frac{\Sigma_{i=1}^{N}(y_i-\hat{y}_i)^2}{\Sigma_{i=1}^{N}(y_i-\mu)^2}, \quad (10)$$

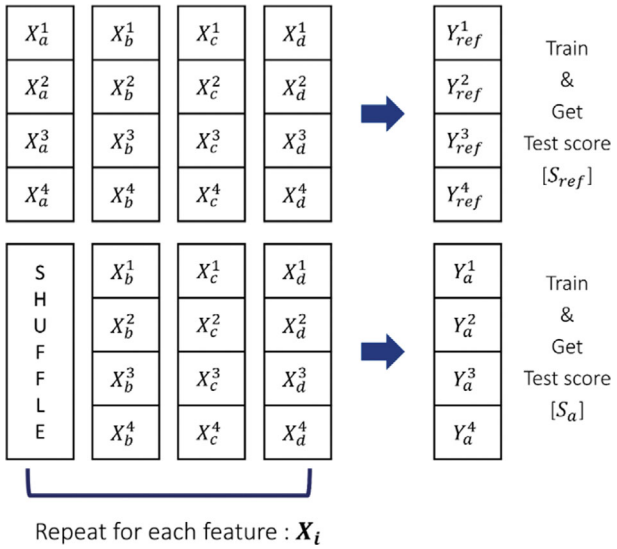

Fig. S3. Description of permutation importance analysis. The subscript alphabet indicates the type of input feature, and the superscript number indicates the index of the data set composed of input features and corresponding output variable.

where μ indicates the mean value of whole y_is.

If the greatest variation in result appears after a certain feature vector is shuffled, it can be interpreted as that feature is substantial for the target system. Since the randomness of shuffling can affect the prediction performance in most of the data science field, five multiple shuffles were performed for each input feature vector and their average score was used for permutation importance analysis.

The R^2 of unshuffled case is represented with subscript *ref* and the R^2 obtained after shuffling a particular feature vector X_i is represented with subscript *i*. The mean value of R^2 obtained from five times random shuffling is notated as $\overline{R_i^2}$ and the permutation importance score (PI score) is defined as below,

$$\text{Score}_{X_i}=R_{ref}^2-\overline{R_i^2}. \quad (11)$$

3. Information about Quantitative Analyses

In order to perform a quantitative comparison, we mainly compared two things that are expected to have the most difference: 1) data storage capacity and 2) the time required to calculate the main features starting from the stored cycle data.

The following factors were taken into consideration.

- Since the data set we used has already been measured, not obtained through direct experiments, it was difficult to precisely compare how much time required during the necessary cycle data for each method was measured in actual battery operation. Therefore, in the analysis of 2), only the time taken for actual pre-processing and feature calculation was measured after all the data required for each method was saved and ready to use.
- In the benchmark study, the Var(ΔQd(V)) feature was considered, and in this study, the Δm feature was considered as the main feature.
- In the benchmark study, after smoothing spline process, it was not clearly mentioned how many points were sampled to cal-

culate variance, so we assumed to sample as many points as the average of the number of data points actually measured in each cycle, at regular intervals.

- In addition, in the case of our proposed method was applied to 100^{th} cycle, it still showed comparable performance to the benchmark and other studies. So we used the case when the same method was applied to 100^{th} cycle into comparison.

REFERENCES

1. K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggedakis, M. Z. Bazant, S. J. Harris, W. C. Chueh and R. D. Braatz, *Nat. Energy*, **4**, 383 (2019).
2. Z. Fei, F. Yang, K.-L. Tsui, L. Li and Z. Zhang, *Energy*, **225**, 120205 (2021).
3. F. Xu, F. Yang, Z. Fei, Z. Huang and K.-L. Tsui, *Reliab. Eng. Syst. Saf.*, **208**, 107396 (2021).
4. C.-I. Chen and Y.-C.Chen, *IEEE Trans. Power Delivery*, **30**, 1577 (2015).
5. X. Lu, D. G. Giovanis, J. Ybonnet, V. Papadopoulos, F. Detrez and J. Bai, *Comput. Mech.*, **64**, 307 (2019).
6. S. Walczak and N. Cerpa, *Inf. Software Technol.*, **41**, 107 (1999).
7. D. Song, C. Sun, Q. Wang and D. Jang, *Energy Power Eng.*, **10**, 10 (2018).
8. X. Li, C. Yuan, X. Li and Z. Wang, *Energy*, **190**, 116467 (2020).
9. D. Yang, X. Zhang, R. Pan, Y. Wang and Z. Chen, *J. Power Sources*, **384**, 387 (2018).
10. Y. Li, H. Sheng, Y. Cheng, D.-I.Stroe and R. Teodorescu, *Appl. Energy*, **277**, 115504 (2020).
11. M. A. Patil, P. Tagade, K. S. Hariharan, S. M. Kolake, T. Song, T. Yeo and S. Doo, *Appl. Energy*, **159**, 285 (2015).
12. L. Melkumova and S. Y. Shatskikh, *Procedia Eng.*, **201**, 746 (2017).
13. B. Lakshminarayanan, A. Pritzel and C. Blundell, *Adv. Neural Inf. Process. Syst.*, 30 (2017).