Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

SPECIAL SERIES ARTICLE

Data-driven battery capacity estimation using support vector regression and model bagging under fast-charging conditions

Yixiu Wang[†1] | Qiyue Luo[†1] | Liang Cao[1] | Arpan Seth[2] | Jianfeng Liu[3] | Bhushan Gopaluni*[1] | Yankai Cao*[1]

[1]Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC, V6T 1Z3, Canada
[2]Process Technology and Engineering, Evonik Corporation, Trexlertown, PA, 18087, USA
[3]Amazon, Seattle, WA, 98109, USA

Correspondence
[†]These authors contribute equally.
*Corresponding author: Bhushan Gopaluni.
Email: bhushan.gopaluni@ubc.ca
*Corresponding author: Yankai Cao.
Email: yankai.cao@ubc.ca

Abstract

Lithium-ion batteries offer significant advantages in terms of their high energy and power density and efficiency, but capacity degradation remains a major issue during their usage. Accurately estimating the remaining capacity is crucial for ensuring safe operations, leading to the development of precise capacity estimation models. Data-driven models have emerged as a promising approach for capacity estimation. However, existing models predominantly focus on constant current charging conditions, limiting their applicability in real-world scenarios where fast-charging conditions are commonly employed. The primary objective of this work is to develop a more versatile machine learning model (i.e., support vector regression [SVR]) capable of estimating battery capacity under fast-charging conditions, with broader applicability across various work conditions. Genetic algorithm and cross-validation techniques are employed to simultaneously optimize feature extraction hyperparameters and SVR hyperparameters. A model bagging method is further implemented to address prediction challenges under unknown fast-charging conditions. The effectiveness of the developed model is validated on a cycling dataset of lithium-ion batteries under different two-stage fast-charging conditions.

KEYWORDS:
battery capacity estimation, support vector regression, model bagging, fast-charging conditions

1 | INTRODUCTION

Lithium-ion batteries (LiBs) stand out for their high energy density, offering a smaller size and lighter weight than other rechargeable battery technologies.[1] Whittingham pioneered the development of the first rechargeable LiB in the late 1970s,[2] which was further advanced in 1980 by Goodenough and colleagues' introduction of lithium cobalt oxide as a cathode material.[3] Sony's

commercialization of LiBs in 1991 [4] has led to widespread adoption across diverse industries, including portable electronics, electric vehicles, and energy storage systems. [5]

Nonetheless, during the repetitive charging and discharging cycles, LiBs undergo degradation possibly due to lithium plating [6] and the growth of solid electrolyte interphase (SEI) film. [7,8] Such degradation results in an irreversible loss in capacity known as capacity fade, which directly affects battery performance and lifespan. [9] Accurate quantification of battery capacity is crucial to prevent premature failure and maximize the utilization of LiBs, thereby enhancing their reliability, efficiency, and safety in practical applications. However, directly measuring battery capacity presents significant challenges, requiring full charging and discharging processes while measuring energy flow. This measurement is time-consuming and often impractical to perform online during usage. To address this, continuous advancements in modelling approaches have contributed to capacity estimation with reasonable accuracy.

Two major capacity estimation models exist—model-based and data-driven approaches. [10] The model-based approach, also known as the state of charge (SOC)-based approach, [11] attempts to estimate the battery capacity based on the SOC estimation results in a dual time scale system. Within this framework, equivalent circuit models [12] and electrochemical models [13] are frequently employed to delineate the battery's dynamic behaviour. Following this, some recursive adaptive filters such as the extended Kalman filter (EKF) [14,15] and particle filter (PF) [16] are utilized to identify model parameters for SOC estimation and consequent capacity update. For instance, Plett [14] proposed a dual EKF method to jointly estimate SOC and capacity of the battery, and the experimental results confirmed that capacity estimation can converge to the correct value. However, despite the high accuracy of model-based capacity estimation methods, they involve extensive computational processes, limiting their applicability for online usage.

In recent years, data-driven methods have garnered significant interest in the research field of batteries. These methods focus on discerning the intrinsic relationship between the battery capacity and features extracted from sensor measurements of voltage, current, and temperature. Extracting valuable features is the initial step in developing a data-driven model, and much attention has been focused upon leveraging physical and electrochemical principles to extract battery characteristics related to capacity. Noteworthy methodologies in this domain include the use of incremental curves (IC), [17] differential voltage (DV) curves, [18] differential thermal (DT) curves, [19] and electrochemical impedance spectroscopy (EIS). [20] IC methods estimate battery capacity through analysis of the incremental change in capacity during charging, a process aided by Coulomb counting. Zhang et al. [21] utilized the second peak from high-rate IC curves, while Guo et al. [22] identified the IC peak from constant-current charging data for battery capacity estimation. DV curves estimate capacity based on voltage differentiation over capacity, and Han et al. [23] demonstrated good consistency between DV and IC curves. Additionally, considering reversible heat generation due to entropy changes during battery operation, [24], Maher and Yazami [19] introduced the DT curve method, utilizing peak and valley points on the curve to estimate capacity. For methods based on EIS, Zhang et al. [20] employed Gaussian process regression (GPR) to

predict LiB capacity and achieved good performance. However, these methods require specialized knowledge of the underlying electrochemical reactions, and implementing sensors such as EIS measurements can pose challenges in practical applications.

Recent explorations have also considered deriving estimations directly from the charging curve.[25–27] More specifically, Yang et al.[25] identified four critical features from the constant current–constant voltage (CC-CV) charging curve, including the time of CC model duration, the time of CV mode duration, the slope of the curve at the end of CC charge mode, and employed GPR for estimating battery capacity. Guo et al.[26] extracted 14 features related to capacity, charge time, temperature, and current/voltage drop features from the charging process, achieving accurate capacity estimation through relevance vector machine (RVM). Zhu et al.[27] introduced a novel approach by extracting three statistical features from the relaxation voltage profile after the charging process, employing techniques like XGBoost and support vector regression (SVR) to achieve high accuracy in capacity estimation. These methods demonstrate a keen ability to navigate the nonlinear dynamics often present in battery data, thereby enhancing their predictive accuracy and applicability in complex scenarios. Nevertheless, a notable challenge with data-driven methods is their reliance on a substantial volume of training data, often specific to a particular operating condition. This requirement can hinder their adaptability and effectiveness in different or previously unobserved scenarios, underscoring the importance of developing methodologies for effective model transfer to broaden their applicability.

Fast-charging, also known as rapid-charging, allows for the charging of batteries at significantly higher rates than conventional methods, reducing charging time from hours to minutes.[28,29] This technology has seen considerable application in the electric vehicle sector, where one of its primary goals is to mitigate range anxiety.[30] By significantly reducing the time needed for charging, it lessens the downtime associated with recharging on long trips, thereby making electric vehicles more appealing to consumers. Despite the benefits, a critical drawback of fast-charging is that it can accelerate capacity fade in batteries, making accurate capacity estimation essential for reliable battery performance under fast-charging conditions.[31] While recognizing the importance of precise capacity assessments under these conditions, the challenges posed by the reduced charging time, requiring feature extraction from shorter time frames and less data, have limited research on capacity estimation under fast-charging conditions, hindering the understanding of battery behaviour and degradation during rapid charging processes. Bridging this research gap is crucial to fully realize the potential of fast-charging technologies and ensure efficient and dependable battery management in applications that rely on rapid energy replenishment.

The primary objective of this manuscript is to develop a highly adaptable machine learning model capable of accurately estimating battery capacity during fast-charging conditions across diverse operating scenarios. SVR is selected for its high efficacy in handling small- to medium-sized datasets, ability to model nonlinear relationships effectively, and robustness against overfitting given the dynamic nature of the battery data. This work employs the genetic algorithm and cross-on techniques to simultaneously optimize feature extraction parameters and SVR hyperparameters. Additionally, a model bagging method is implemented to address potential prediction challenges associated with unknown fast-charging conditions. The effectiveness and

reliability of the developed model are validated through experimentation using a cycling dataset of LiBs subjected to various two-stage fast-charging conditions.

The remainder of the paper is organized as follows. The cycling dataset utilized in this study is presented in Section 2, and Section 3 introduces the proposed methodology. The results of capacity estimation are thoroughly discussed in Section 4 and, finally, conclusions are drawn in Section 5.

2 | CYCLING DATASET

To evaluate the performance of the proposed method on capacity estimation, this work utilizes the open-source cycling dataset from Severson et al.[32] The batteries tested in the dataset are commercial high-power LFP/graphite A123 APR18650M1A cells, each with a nominal capacity of 1.1 Ah and a nominal voltage of 3.3 V. The original paper primarily focuses on the impact of fast-charging protocols, employing a two-step fast-charging strategy denoted as C1(Q1%)–C2. More specifically, the battery cell is first charged at a C1 current rate until SOC reaches Q1%. Then, the charging rate switches to C2 until the SOC hits 80%. Following this, the battery is charged at a constant current of 1C until the voltage reaches 3.6 V, followed by constant voltage charging. As for the discing process, all batteries are uniformly discharged at a 4C rate in a constant current mode until the voltage drops to 2.0 V, as illustrated in Figure 1A. To improve the applicability of the model, in this paper, we propose using only the voltage curve from the first stage of the fast-charging process for feature extraction in capacity estimation, as depicted in Figure 1B.

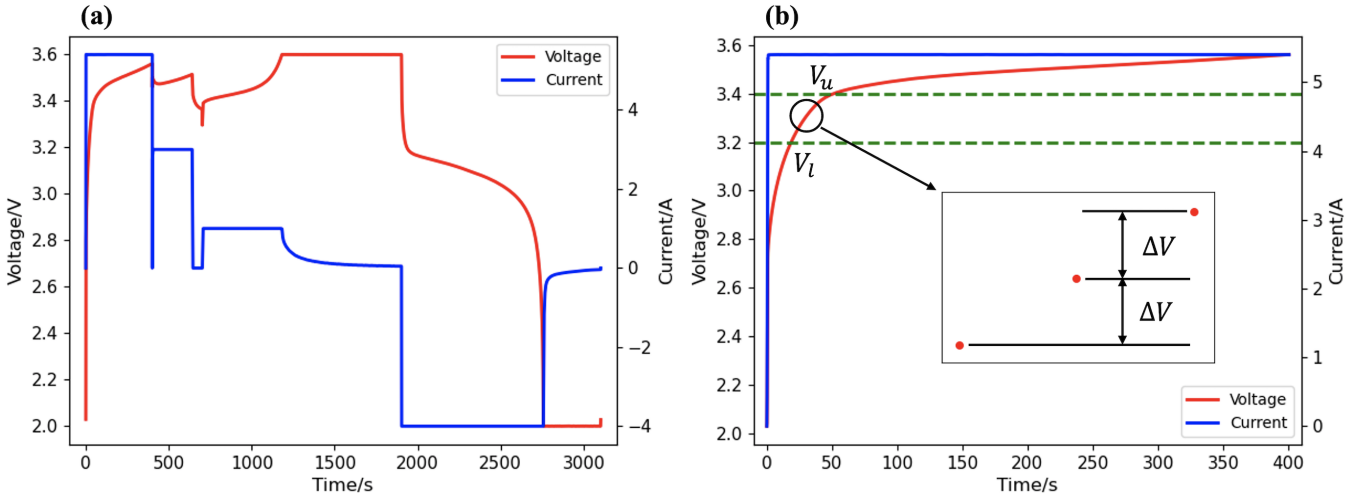

Figure 1 Voltage and current profile: (A) a complete charging and discharging process, (B) the first stage of fast-charging.

In this work, we utilized 12 battery cells cycled under different operating conditions, that is, with different combinations of C1, Q1%, and C2, as indicated in Table 1. It is important to note that we have renumbered the cells to facilitate discussion. These

cells were categorized into different sets based on the different purposes of our study. From Group 1 to Group 5, each consisted of two cells with identical fast-charging policy. One battery data in each group was used to train an SVR model, while the other battery's data was employed to test the performance of the proposed SVR model. The last group, comprising two cells, operated under conditions different from the previous five sets and was used to evaluate the effectiveness of model bagging in adapting to new and unknown scenarios. We assumed that for these new scenarios, only a limited number of initial cycle data points are available for training the weights in the model bagging.

Table 1 Description of two-step charging policy for cells.

Battery group	Charging policy			Cell index
	C1	Q1%	C2	
Group 1	4.8	80%	4.8	#1, #2
Group 2	5.4	60%	3	#3, #4
Group 3	6	50%	3.6	#5, #6
Group 4	7	40%	3	#7, #8
Group 5	8	25%	3.6	#9, #10
Group 6	5.2	50%	4.25	#11
	5.3	54%	4	#12

3 | METHODOLOGY

3.1 | Feature extraction from the charging process

Initially, the process involves extracting input features from the voltage and current data gathered during the charging and discharging cycles of the cells. This forms the foundation for constructing a data-driven model. Given the practical constraints in real-world applications, where the battery's discharging process can become unpredictable due to varying operating conditions and loads[27], this study strategically selects portions of the charging process curves for feature extraction. In particular, we focus on a specific range of the first fast-charging process to enhance the practical applicability of the proposed method. To define this range, we establish a lower limit (V_l) and an upper limit (V_u), with V_u set to be 0.2 V higher than V_l ($V_u = V_l + 0.2$).

For the data-driven model, as suggested in the work of Li et al.[33], the input features consist of a vector formed by the relative capacity $\hat{Q}_k$ at each point. To guarantee the same number of sampling points across various cycles and ensure that the length of the input features remains consistent, we discretize the charging curve at uniform voltage intervals ΔV, as shown in Figure 1B, followed by linear interpolation to determine the charging capacity at each discrete point $\hat{V}_k$. Subsequently, by setting the charging capacity at a lower limit V_l as the reference value of 0, these charging capacities at discrete points serve as the relative capacities

input into the model, which provides a precise measure of the capacity at various stages of the charging process. The output of the model is the recorded discharging capacity for the corresponding cycle, derived from the integral of the current over time throughout the complete discharging process, offering an evaluation of the current health state of the battery cell.

Crucially, selecting the specific capacity range, particularly the lower limit V_l, and the discretization interval ΔV, is not arbitrary. These parameters are meticulously optimized using a genetic algorithm in conjunction with cross-validation. This optimization process is crucial to determine the most accurate and predictive range for capacity calculation, thereby enhancing the model's precision and reliability.

3.2 | Support vector regression

SVR is a machine learning regression technique extending from support vector machines (SVM), which was originally developed in 1992.[34] SVR has several demonstrated advantages—it is highly effective in high-dimensional spaces, is very suitable for time-series prediction, and has excellent versatility in changing different kernel functions to adapt to specific decision functions.[35] Unlike traditional regression approaches that aim to minimize the error between predicted and actual values, SVR focuses on finding a hyperplane that maximizes the margin around the data points within a certain tolerance level. In addition, the kernel trick enables it to efficiently handle nonlinear relationships.

Consider a training dataset $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ consisting of n samples, where $x_i \subset R^d$ donates the d-dimensional input feature of the sample and y_i is the corresponding output value (target). The generic SVR estimation function takes the following form:

$$\hat{y}_i = \omega^T \varphi(x_i) + b \tag{1}$$

where $\varphi(x_i)$ is a nonlinear kernel transformation from R^d to R^m, and m is the dimension of the new feature space.

The goal of SVR is to find the values of ω and b that minimize the total loss:

$$\min_{\omega,b} \frac{1}{2}||\omega||_2^2 + C\sum_{i=1}^{n} \ell_\varepsilon\left(\hat{y}_i - y_i\right) \tag{2}$$

where C is a constant penalty and ℓ_ε is the loss function. In this study, we use the ε-insensitive loss function as follows:

$$\ell_\varepsilon\left(\hat{y}_i - y_i\right) = \begin{cases} \left|\hat{y}_i - y_i\right| - \varepsilon, & \left|\hat{y}_i - y_i\right| \geq \varepsilon \\ 0, & \left|\hat{y}_i - y_i\right| < \varepsilon \end{cases} \tag{3}$$

3.3 | Genetic algorithm for hyperparameters tuning

The genetic algorithm is an optimization method inspired by the principles of natural selection and genetics, rooted in Darwin's theory of evolution.[36] John Holland introduced the genetic algorithm in 1975 and expanded it in 1992.[37] The genetic algorithm

is currently widely applied in solving highly complex and multivariate optimization problems, especially when the mathematical model of the problem is unclear or when gradient information is challenging to compute. In this work, we utilize the genetic algorithm as a pivotal tool for simultaneously optimizing the hyperparameters involved in feature extraction (V_l and ΔV) and the hyperparameters of the SVR model (kernel, gamma, epsilon, and C).

The implementation of the genetic algorithm begins with the generation of an initial population. This population comprises various combinations of the hyperparameters, each gene representing a potential solution. The evaluation of these solutions is conducted using a fitness function. In this work, the assessment of the performance of each candidate hyperparameter set in predicting battery capacity is achieved through cross-validation.

The algorithm then iterates through a cycle of selection, crossover, and mutation to evolve the population towards optimal solutions, as shown in Figure 2. In the selection phase, the best-performing individuals, according to the fitness function results, are chosen to pass their 'genes', that is, their hyperparameter combinations, to the next generation. This mimics the natural selection process where fitter individuals are more likely to reproduce. The crossover and mutation processes then introduce new genetic combinations and variations, respectively. Crossover combines aspects of two parent solutions to create offspring, while mutation introduces random changes to individual parameters, ensuring diversity within the population and preventing premature convergence on sub-optimal solutions. This cycle continues until a predefined termination condition is met, such as finding a solution that meets the minimum criteria or reaching the maximum number of generations.

The fitness function evaluates the performance of each individual, determining whether their genetic information should be retained and if the iteration cycle should terminate. In this study, we utilize the root mean square error (RMSE) from 5-fold cross-validation to assess the performance of each hyperparameter combination, thereby avoiding the potential overfitting of the data and yielding more reliable results. More specifically, we first randomly divide the complete dataset of each battery into 5 equal folds, as illustrated in Figure 3. Subsequently, we conduct five iterations to calculate the model error. In each iteration, an SVR model is trained using four folds of the data and tested for its RMSE on the remaining fold. The fitness is determined by the average RMSE across these five iterations. Furthermore, to ensure that the proposed SVR model performs well across different groups of batteries, the final fitness is calculated as the average fitness of the five battery groups.

3.4 | Model bagging

Training a machine learning model typically requires extensive data. For instance, capacity estimation necessitates cycling data covering an entire lifespan to ensure the model learns the degradation patterns at different stages, which is expensive and time consuming to collect. Therefore, constructing a machine learning model specific to each working condition is unrealistic. Instead, building a transferable model is essential and economically favourable. In this manuscript, model bagging is used to develop models for cell #11 and cell #12 (Group 6), which are considered as the unknown condition cells, providing the knowledge of

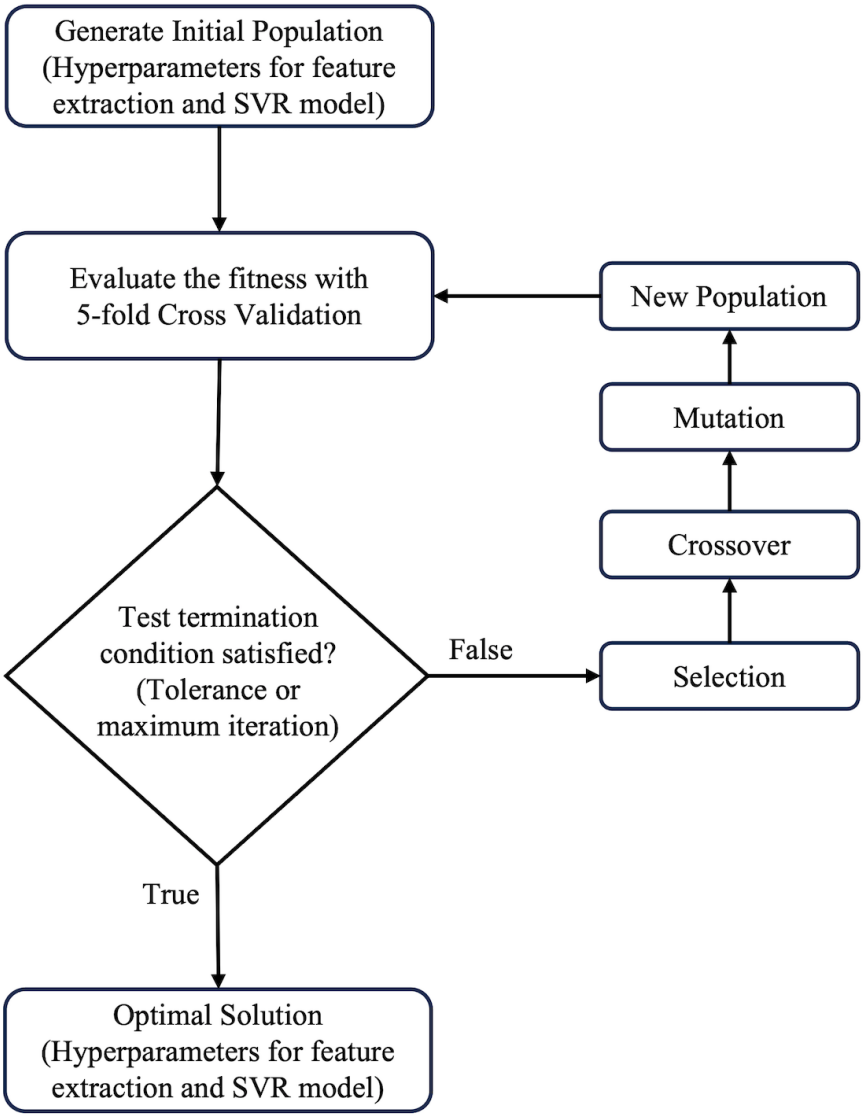

Figure 2 The procedure of the genetic algorithm.

the data for the cells working under five different work conditions. The rationale behind using model bagging was to enhance the robustness and accuracy of predictions by leveraging the strengths of multiple models and mitigating the weaknesses inherent in a single-model approach.[38] This method allows for a more comprehensive understanding and adaptation to the complex and varying conditions faced by the cell, ultimately leading to a more accurate and reliable predictive model.

Initially, this study involves training five distinct SVR models with the hyperparameters that have been tuned earlier using the genetic algorithm under each specific work condition in Table 1. These five models are referred to as source models. Then, for cells cycled under new conditions, these five source models can predict varying capacity values. Our final target model is a combination of predictions from these five models.

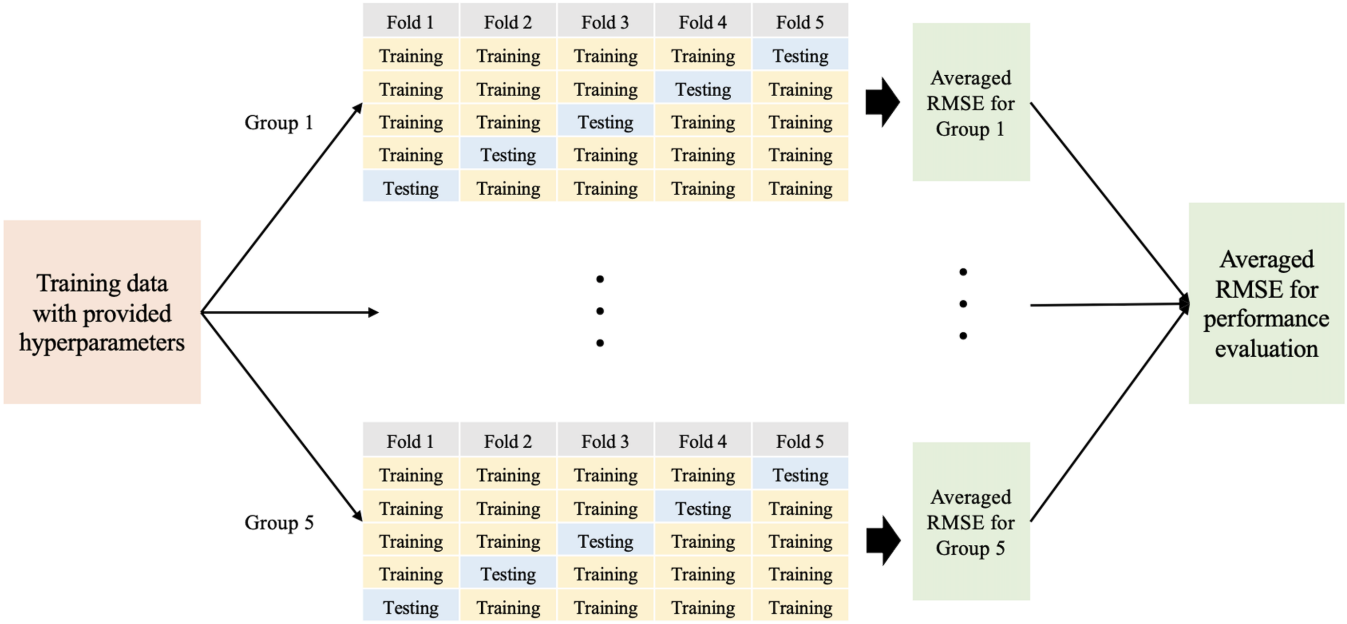

Figure 3 The procedure of 5-fold cross-validation.

Unlike traditional model bagging, such as random forest regression (RFR), which uses the average of predictions from each sub-model as the output, here we assign different weights to the different models. More specifically, we use the first 100 cycles of cycling data from the new cell as a retraining dataset. The weights for each model are then determined by solving the following optimization problem aimed at minimizing the possible prediction error:

$$\begin{aligned} \min_{w} \quad & \sum_{i=1}^{n'} (\hat{y}_i - y_i)^2 \\ \text{s.t.} \quad & \hat{y}_i = \sum_{j=1}^{5} \alpha_j \hat{y}_i^{\text{SVR}_j} \\ & \sum_{j=1}^{5} \alpha_j = 1 \end{aligned} \tag{4}$$

where $\hat{y}_i^{\text{SVR}_j}$ is the prediction from the j-th SVR model for sample i, n' is the number of samples for retraining, and α_j is the weight for the j-th SVR model. After computing the optimal weights of the five source models, the target model is used to test the remaining cycling data of the new cell.

4 | RESULTS AND DISCUSSIONS

4.1 | Optimal hyperparameters

In this work, partial charging curves within a specific voltage range are utilized to extract features for battery capacity estimation. The selected voltage range and discretization interval significantly influence model accuracy. Additionally, the hyperparameters

of the SVR model affect its performance. To enhance the effectiveness of the proposed SVR model, the genetic algorithm is applied to identify the optimal hyperparameters. In particular, for feature extraction, the candidate lower limit of the voltage range V_l extends from 2.5 to 3.3 V, increasing in 0.05 V intervals, resulting in 17 possible selections. The candidate discrete interval ΔV ranges from 0.0005 to 0.005 V, increasing in 0.0005 V intervals, offering 10 choices. Regarding the hyperparameters of the SVR mbodel, the choices of kernel include 'poly', 'rbf', and 'sigmoid', and the choices of gamma include include 'scale' and 'auto'. Both epsilon and C vary continuously, with candidate ranges of $[10^{-5}, 10]$ and $[10^{-3}, 10]$, respectively.

Table 2 lists the optimal parameters for feature extraction and the SVR model as obtained through the genetic algorithm. In the context of feature extraction hyperparameters, the voltage range is narrowed down to 3.2–3.4 V with a ΔV of 0.004 V. This specific range and discrete interval are likely determined to be the most relevant for capturing the essential characteristics of the data, potentially improving the model's accuracy and performance. As for the SVR model, the chosen kernel is the radial basis function ('rbf'), renowned for its flexibility and efficiency in managing non-linear data.[39] The gamma parameter, setting to 'scale', plays a crucial role in determining the influence radius of the support vectors[40]. In this setting, it adapts to the dataset's scale, providing a balance between influence reach and model complexity. The epsilon value, at 0.0020, defines the margin of tolerance within which no penalty is given to errors. This small epsilon value indicates a model that closely fits the training data, allowing minimal deviation. The C parameter, set at 0.5835, represents the trade-off between maximizing the margin and minimizing the prediction error.

Table 2 Optimal hyperparameters from the genetic algorithm.

Feature extraction		SVR model	
V_l	3.2 V	kernel	'rbf'
ΔV	0.004 V	gamma	'scale'
		epsilon	0.0020
		C	0.5835

Figure 4 further demonstrates the RMSE changes under different combinations of feature extraction hyperparameters. It is evident from Figure 4A that as the lower limit of voltage range V_l increases, the RMSE initially decreases significantly before gradually ascending, suggesting that data from the early stages of charging contain insufficient information for capacity estimation. It seems that only the charging curve data after reaching 3.0 V can yield satisfactory results. In contrast, the discrete interval ΔV has a minor impact on RMSE. The highest RMSE recorded is at an interval of 0.003 V with a value of 0.00633 Ah, while the lowest RMSE is achieved at an interval of 0.004 with a value of 0.00623 Ah, showing no significant difference.

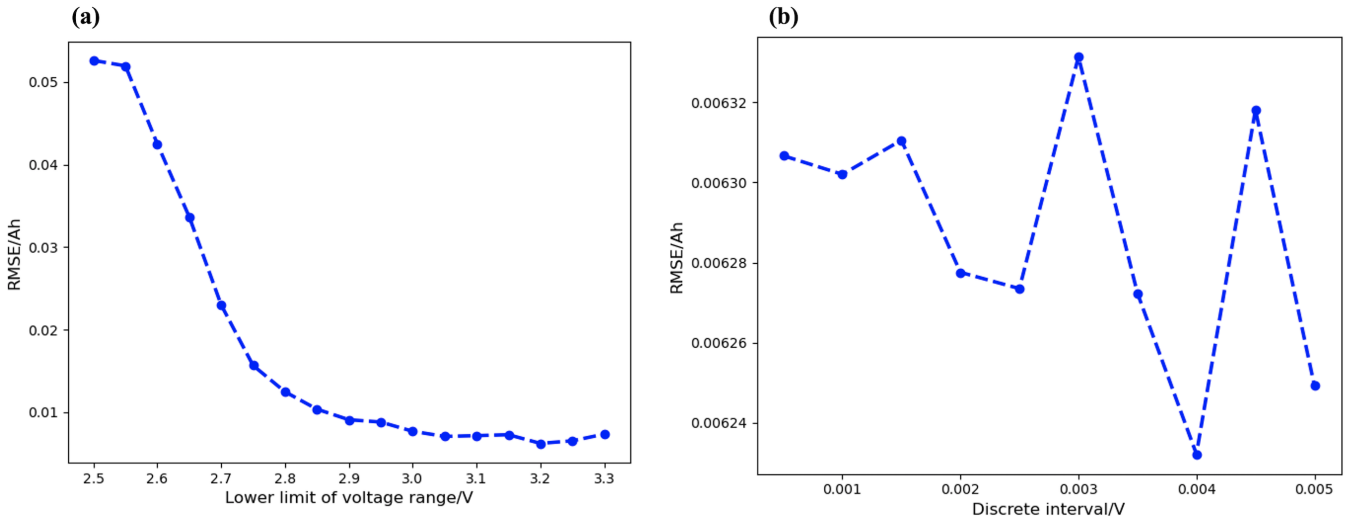

Figure 4 Root mean square error (RMSE) (Ah) change with different feature extraction hyperparameters: (A) lower limit of voltage range V_l, (B) discrete interval ΔV.

4.2 | Results of the SVR model on capacity estimation

In the evaluation of the proposed SVR model, its efficacy is benchmarked against other commonly used machine learning models, including ridge regression,[41] RFR,[33] GPR,[42] and long short-term memory (LSTM) network,[43] covering a range from simple linear models to complex neural networks. All models are trained on one cell from groups 1 to 5 listed in Table 1, which includes cells #1, #3, #5, #7, and #9, and then tested on another cell from each group, which includes cells #2, #4, #6, #8, and #10. To ensure that the results are comparable, all models use the same feature extraction hyperparameters, resulting in identical input features. Table 3 depicts the performance comparison of different models, with RMSE serving as the evaluation metric. The results revealed that the proposed SVR model consistently achieved superior performance, followed by the RFR. It is also noteworthy that ridge regression shows the worst performance, with acceptable results only on cell #4, achieving an RMSE of 0.0136 Ah, suggesting its inability to capture the nonlinear dynamic of the battery degradation. GPR performs slightly better than ridge regression but only achieved acceptable accuracy on cell #6 with an RMSE of 0.0140 Ah, implying a tendency to overfit when handling this battery dataset. The LSTM network works well on cells #4, #6, and #8, but fails to accurately estimate the capacities of cells #2 and #8, with errors exceeding 0.02 Ah, indicating a high risk of overfitting for complex neural networks with our medium-sized dataset (fewer than 1000 samples per group). More specifically, the SVR model records the lowest RMSE for each of the five cells, all being below 0.0140 Ah. This denotes not only its precision in estimating capacity but also highlights its reliability in various conditions, making it a robust choice in the realm of battery capacity estimation.

Figure 5A–E depicts the comparison between measured and estimated battery capacity across different testing cells. These figures reveal that, aside from a single estimation outlier in cell #8 and cell #10, the estimated values are highly consistent with the measured ones for most of the points, demonstrating the model's generalizability and precision in estimating battery

Table 3 Root mean square error (RMSE) (Ah) comparison of different models on capacity estimation.

Model	Cell #2	Cell #4	Cell #6	Cell #8	Cell #10
Ridge regression	0.0215	0.0136	0.0358	0.4327	0.0458
RFR	0.0157	0.0120	0.0129	0.0104	0.0085
GPR	0.2904	0.0218	0.0140	0.0430	0.0427
LSTM network	0.0239	0.0148	0.0132	0.0430	0.0138
Proposed SVR	**0.0137**	**0.0106**	**0.0082**	**0.0080**	**0.0084**

capacity. Particularly, the capacity estimation curves in the figures tightly follow the measured capacity trajectories, capturing the trends accurately even in areas where capacity sharply decreases, highlighting the model's ability to accurately capture the nonlinear degradation characteristics of battery cells. Figure 5F further summarizes the distribution of estimation errors, where it is observed that the errors for the majority of the samples are contained within 0.01 Ah, confirming the effectiveness of the proposed SVR model.

4.3 | Verification of model bagging

After training 5 SVR models to accurately estimate the capacity under five known conditions, model bagging is employed to train a target model for new conditions using only the data from the initial 100 cycles. To demonstrate the efficacy of the proposed method, we compared it with two other models: one is an SVR model trained on the combined data from all five conditions, referred to as the average model, and another SVR model trained from scratch using only the data from the initial 100 cycles, referred to as the TL0 model. To ensure comparability of the results, all SVR models use the same hyperparameters, which are optimized through the genetic algorithm.

Table 4 displays a comparative performance of different models on testing samples under new conditions. The results indicate that the TL0 model achieves the poorest performance, suggesting that cells exhibit different degradation characteristics at different stages, and relying solely on early-cycle data does not accurately estimate the capacity changes over the entire lifespan of the battery. In contrast to the average model, which has RMSE all above 0.02 Ah, the proposed target model delivered the most superior results with an RMSE of 0.0101 Ah for cell #11 and an RMSE of 0.0055 Ah for cell #12, validating the effectiveness of the proposed model bagging strategy.

Figure 6 displays the performance of the proposed target model on two cells cycled at new working conditions. As depicted in Figure 6A, with the exception of three points, the majority of the capacity for cell #11 is accurately estimated, especially during the mid-cycle stage, where a rapid decline in battery capacity is observed. Figure 6B further exhibits the capacity estimation

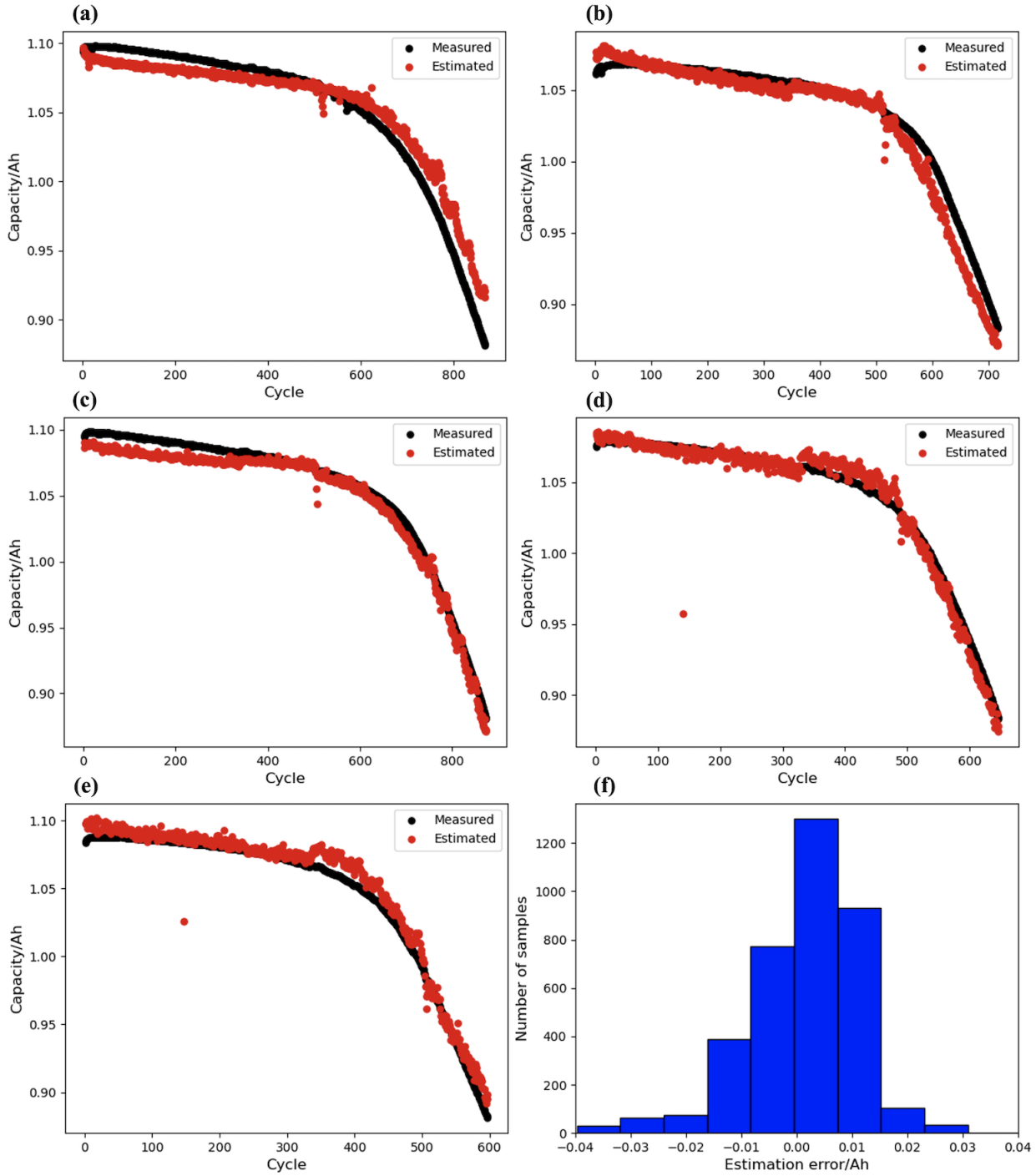

Figure 5 Performance of the proposed SVR model: Comparison of measured and estimated capacity for (A) cell #2, (B) cell #4, (C) cell #6, (D) cell #8, and (E) cell #10. (F) Distribution of prediction error for all testing samples.

errors across various cycles, showing that the errors between the 100th and 450th cycles remain within 0.01 Ah. In the later-cycle stages, as the battery degrades, the errors increase but are predominantly concentrated around zero, signifying the model's overall high precision. Figure 6C,D showcases the results of capacity estimation for cell #12, demonstrating a similar trend

Table 4 Root mean square error (RMSE) (Ah) comparison of different models on new work conditions.

Model	Cell #11	Cell #12
Average model	0.0266	0.0274
TL0 model	0.0837	0.0526
Target model	**0.0101**	**0.0055**

to cell #11 but with significantly smaller estimation errors, aligning with the RMSE value presented in Table 4. This further confirms the validity of the proposed model bagging method when applied to new operating conditions.

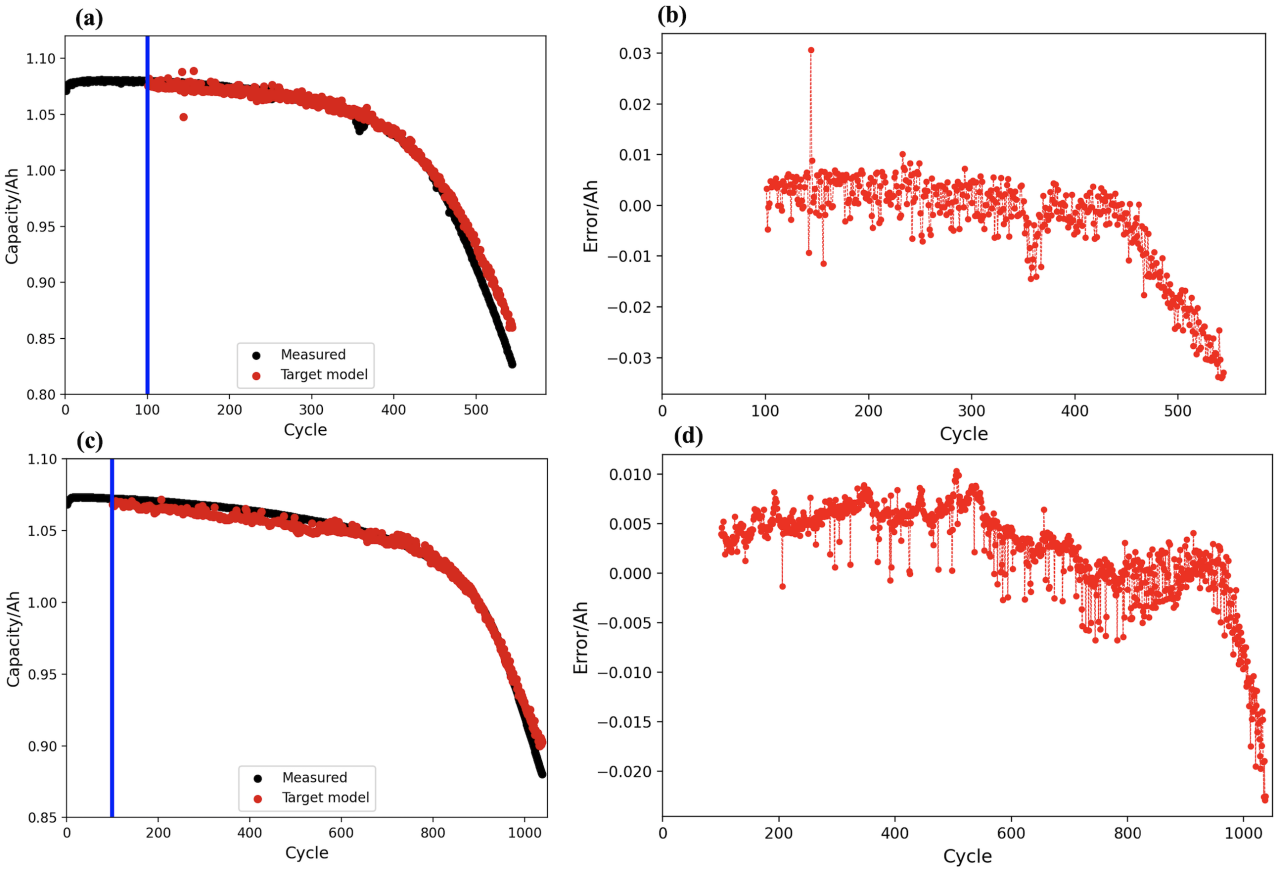

Figure 6 Performance of the target model: Comparison of measured and (A) estimated capacity and (B) estimation error for cell #11; comparison of (C) measured and estimated capacity and (D) estimation error for cell #12.

5 | CONCLUSIONS

This study successfully developed a highly adaptable SVR model for estimating battery capacity under fast-charging conditions. Utilizing the genetic algorithm and cross-validation, the model demonstrated remarkable accuracy in estimating battery capacity

across diverse work conditions. The implementation of a model bagging method effectively addressed the challenge associated with unknown fast-charging conditions, as validated through an LiB cycling dataset. The results, particularly the low RMSE in testing, underscore the model's robustness and reliability in capturing the complex dynamics of battery performance.

Building upon the successful development of the adaptable SVR model for battery capacity estimation under fast-changing conditions, it is recommended that this research be extended in several key directions. First, integrating the model with real-time monitoring systems, particularly in electric vehicles, could significantly improve the efficiency of LiBs. Additionally, future studies might also benefit from exploring the model's performance with larger and more diverse datasets to further understand its scalability.

ACKNOWLEDGMENTS

Yankai Cao acknowledges funding from New Frontiers in Research Fund under grant NFRFE-2022-00663 and the NSERC Discovery Program under grant RGPIN-2019-05499. Bhushan Gopaluni would like to acknowledge the financial support from NSERC Discovery grant. We gratefully acknowledge the computing resources provided by SciNet (www.scinethpc.ca) and Digital Research Alliance of Canada (alliancecan.ca).

References

[1] Brian J Landi, Matthew J Ganter, Cory D Cress, Roberta A DiLeo, Ryne P Raffaelle, *Energy & Environmental Science* **2009**, *2* (6), 638–654.

[2] M Stanley Whittingham, *Science* **1976**, *192* (4244), 1126–1127.

[3] KJPC Mizushima, PC Jones, PJ Wiseman, John B Goodenough, *Materials Research Bulletin* **1980**, *15* (6), 783–789.

[4] Mogalahalli V Reddy, Alain Mauger, Christian M Julien, Andrea Paolella, Karim Zaghib, *Materials* **2020**, *13* (8), 1884.

[5] John B Goodenough, Youngsik Kim, *Chemistry of Materials* **2010**, *22* (3), 587–603.

[6] Upender Rao Koleti, Ashwin Rajan, Chaou Tan, Sanghamitra Moharana, Truong Quang Dinh, James Marco, *Energies* **2020**, *13* (13), 3458.

[7] Toshihiro Yoshida, Michio Takahashi, Satoshi Morikawa, Chikashi Ihara, Hiroyuki Katsukawa, Tomoyuki Shiratsuchi, Jun-ichi Yamaki, *Journal of The Electrochemical Society* **2006**, *153* (3), A576.

[8] Satu Kristiina Heiskanen, Jongjung Kim, Brett L Lucht, *Joule* **2019**, *3* (10), 2322–2333.

[9] Andrew Carnovale, Xianguo Li, *Energy and AI* **2020**, *2*, 100032.

[10] Yixiu Wang, Jiangong Zhu, Liang Cao, Jianfeng Liu, Pufan You, Bhushan Gopaluni, Yankai Cao, *Industrial & Engineering Chemistry Research* **2023**, *63* (1), 345–357.

[11] Jichang Peng, Jinhao Meng, Dan Chen, Haitao Liu, Sipeng Hao, Xin Sui, Xinghao Du, *Batteries* **2022**, *8* (11), 229.

[12] Bor Yann Liaw, Ganesan Nagasubramanian, Rudolph G Jungst, Daniel H Doughty, *Solid State Ionics* **2004**, *175* (1-4), 835–839.

[13] Joel C Forman, Scott J Moura, Jeffrey L Stein, Hosam K Fathy, *Journal of Power Sources* **2012**, *210*, 263–275.

[14] Gregory L Plett, *Journal of Power Sources* **2004**, *134* (2), 277–292.

[15] Changfu Zou, Chris Manzie, Dragan Nešić, Abhijit G Kallapur, *Journal of Power Sources* **2016**, *335*, 121–130.

[16] Simon Schwunk, Nils Armbruster, Sebastian Straub, Johannes Kehl, Matthias Vetter, *Journal of Power Sources* **2013**, *239*, 705–710.

[17] Theodoros Kalogiannis, Daniel Ioan Stroe, Jonas Nyborg, Kjeld Nørregaard, Andreas Elkjær Christensen, Erik Schaltz, *ECS Transactions* **2017**, *77* (11), 403.

[18] Ira Bloom, Andrew N Jansen, Daniel P Abraham, Jamie Knuth, Scott A Jones, Vincent S Battaglia, Gary L Henriksen, *Journal of Power Sources* **2005**, *139* (1-2), 295–303.

[19] Kenza Maher, Rachid Yazami, *Journal of Power Sources* **2014**, *247*, 527–533.

[20] Yunwei Zhang, Qiaochu Tang, Yao Zhang, Jiabin Wang, Ulrich Stimming, Alpha A Lee, *Nature Communications* **2020**, *11* (1), 1706.

[21] Caiping Zhang, Yubin Wang, Yang Gao, Fang Wang, Biqiang Mu, Weige Zhang, *Applied Energy* **2019**, *256*, 113841.

[22] YongFang Guo, Kai Huang, XiaoYa Hu, *Journal of Energy Storage* **2021**, *36*, 102372.

[23] Xuebing Han, Minggao Ouyang, Languang Lu, Jianqiu Li, Yuejiu Zheng, Zhe Li, *Journal of Power Sources* **2014**, *251*, 38–54.

[24] Qiaoping Zhang, Fanglin Wei, Peng Zhang, Ruize Dong, Jiaxin Li, Pengzhao Li, Qiurong Jia, Yanxia Liu, Jing Mao, Guosheng Shao, *Fire Technology* **2023**, *59* (3), 1029–1049.

[25] Duo Yang, Xu Zhang, Rui Pan, Yujie Wang, Zonghai Chen, *Journal of Power Sources* **2018**, *384*, 387–395.

[26] Peiyao Guo, Ze Cheng, Lei Yang, *Journal of Power Sources* **2019**, *412*, 442–450.

[27] Jiangong Zhu, Yixiu Wang, Yuan Huang, R Bhushan Gopaluni, Yankai Cao, Michael Heere, Martin J Mühlbauer, Liuda Mereacre, Haifeng Dai, Xinhua Liu, Anatoliy Senyshyn, Xuezhe Wei, Michael Knapp, Helmut Ehrenberg, *Nature Communications* **2022**, *13* (1), 2261.

[28] Manuel Weiss, Raffael Ruess, Johannes Kasnatscheew, Yehonatan Levartovsky, Natasha Ronith Levy, Philip Minnmann, Lukas Stolz, Thomas Waldmann, Margret Wohlfahrt-Mehrens, Doron Aurbach, Martin Winter, Yair Ein-Eli, Jürgen Janek, *Advanced Energy Materials* **2021**, *11* (33), 2101126.

[29] Peter M Attia, Aditya Grover, Norman Jin, Kristen A Severson, Todor M Markov, Yang-Hung Liao, Michael H Chen, Bryan Cheong, Nicholas Perkins, Zi Yang, Patrick K Herring, Muratahan Aykol, Stephen J Harris, Richard D Braatz, Stefano Ermon, William C Chueh, *Nature* **2020**, *578* (7795), 397–402.

[30] Anna Tomaszewska, Zhengyu Chu, Xuning Feng, Simon O'kane, Xinhua Liu, Jingyi Chen, Chenzhen Ji, Elizabeth Endler, Ruihe Li, Lishuo Liu, Yalun Li, Siqi Zheng, Sebastian Vetterlein, Ming Gao, Jiuyu Du, Michael Parkes, Minggao Ouyang, Monica Marinescu, Gregory Offer, Billy Wu, *ETransportation* **2019**, *1*, 100011.

[31] Sheng Shui Zhang, *Journal of Power Sources* **2006**, *161* (2), 1385–1391.

[32] Kristen A Severson, Peter M Attia, Norman Jin, Nicholas Perkins, Benben Jiang, Zi Yang, Michael H Chen, Muratahan Aykol, Patrick K Herring, Dimitrios Fraggedakis, Martin Z Bazant, Stephen J Harris, William C Chueh, Richard D Braatz, *Nature Energy* **2019**, *4* (5), 383–391.

[33] Yi Li, Changfu Zou, Maitane Berecibar, Elise Nanini-Maury, Jonathan C-W Chan, Peter Van den Bossche, Joeri Van Mierlo, Noshin Omar, *Applied Energy* **2018**, *232*, 197–210.

[34] Bernhard E Boser, Isabelle M Guyon, Vladimir N Vapnik, in *Proceedings of the fifth annual workshop on Computational Learning Theory*, **1992**, pp. 144–152.

[35] Pablo Rivas-Perea, Juan Cota-Ruiz, David Garcia Chaparro, Jorge Arturo Perez Venzor, Abel Quezada Carreón, Jose Gerardo Rosiles, *International Journal of Intelligence Science* **2013**, *3*, 5–14.

[36] Sourabh Katoch, Sumit Singh Chauhan, Vijay Kumar, *Multimedia Tools and Applications* **2021**, *80*, 8091–8126.

[37] John H Holland, *Scientific American* **1992**, *267* (1), 66–73.

[38] Peter Bühlmann, *Handbook of Computational Statistics: Concepts and Methods* **2012**, 985–1022.

[39] Zhiliang Liu, Hongbing Xu, *Journal of Algorithms & Computational Technology* **2014**, *8* (2), 163–177.

[40] Intisar Shadeed Al-Mejibli, Jwan K Alwan, Hamed Abd Dhafar, *International Journal of Electrical and Computer Engineering* **2020**, *10* (5), 5497.

[41] Gary C McDonald, *Wiley Interdisciplinary Reviews: Computational Statistics* **2009**, *1* (1), 93–100.

[42] Robert R Richardson, Christoph R Birkl, Michael A Osborne, David A Howey, *IEEE Transactions on Industrial Informatics* **2018**, *15* (1), 127–138.

[43] Yan Ma, Ce Shan, Jinwu Gao, Hong Chen, *Energy* **2022**, *251*, 123973.