Electronic Supplementary Material (ESI) for Nanoscale.

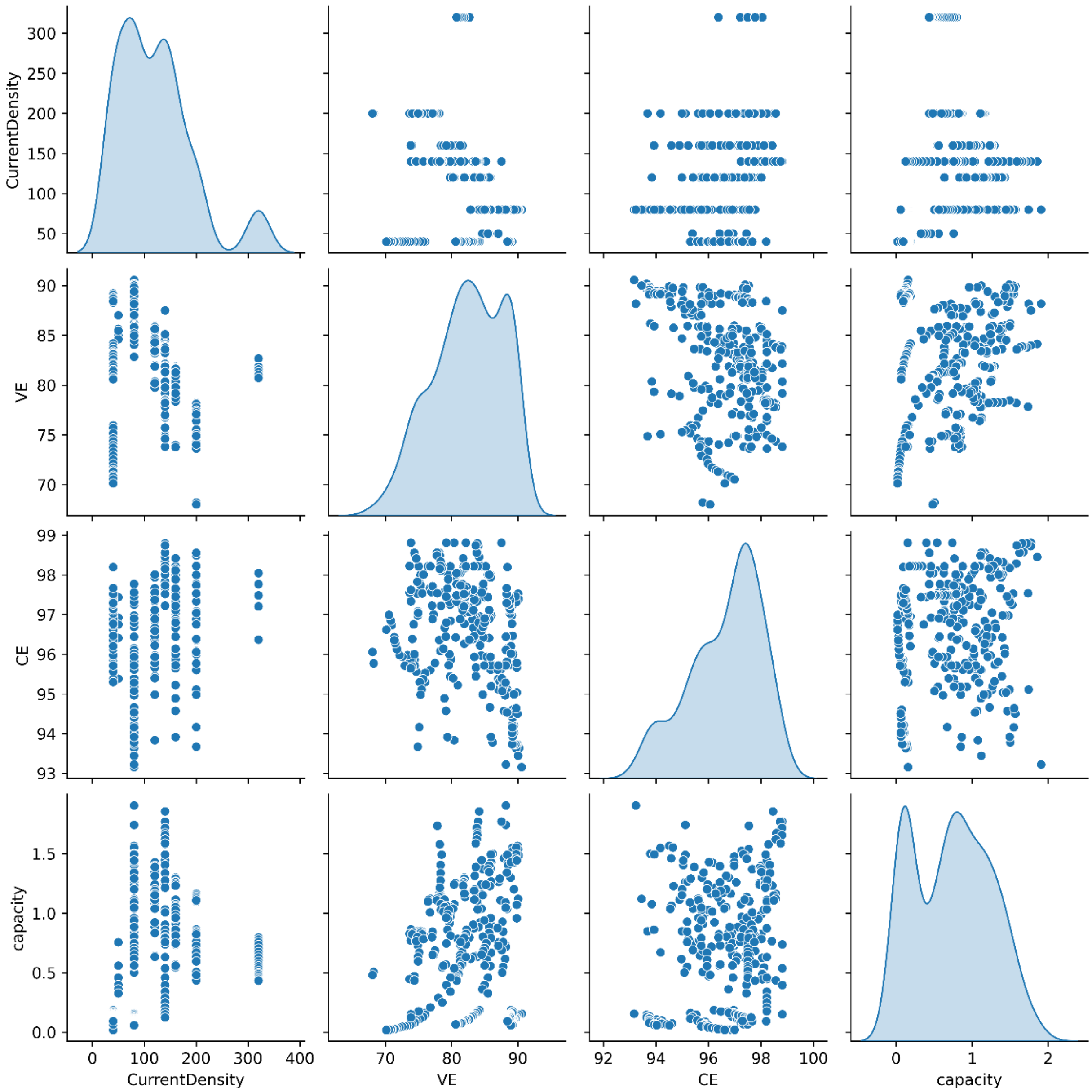
Machine Learning-Enabled Performance Prediction and Optimization for Iron-Chromium Redox Flow Batteries

Yingchun Niu[a=], Ali Heydari[a=], Wei Qiu [a], Chao Guo [a], Yinping Liu [a], Chunming Xu [a], Tianhang Zhou [a*], Quan Xu [a*]

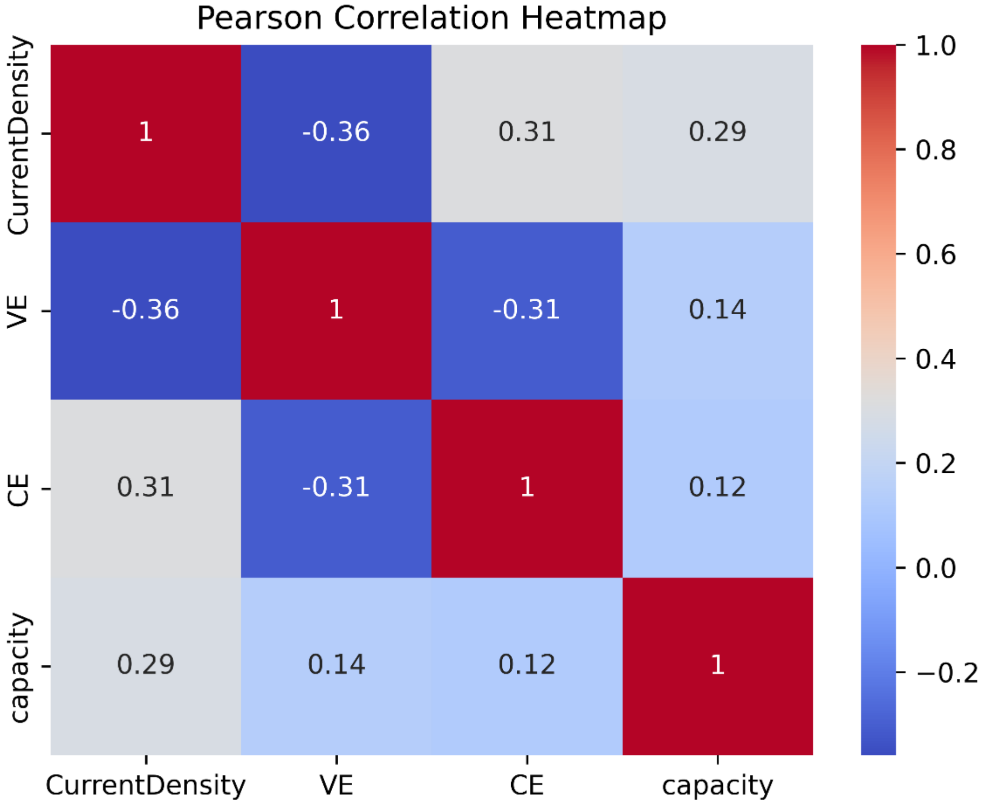
[a] State Key Laboratory of Heavy Oil Processing; China University of Petroleum (Beijing), Beijing 102249, China

E-mail: zhouth@cup.edu.cn, xuquan@cup.edu.cn

Fig. S1a illustrates the distribution of datapoints of voltage efficiency, coulomb efficiency and capacity vs current density in pairplots to evaluate the relationship between these variables. Fig S1b coefficients of Pearson correlation heat map for the investigated variables. Accordingly, while current density is inversely related to voltage efficiency with the highest absolute value equal to -0.36 , it is directly proportional to both coulomb efficiency and capacity with coefficients of 0.31 and 0.29 respectively. In other words, with the increase in current density the voltage efficiency will decay whereas coulomb efficiency and capacity are enhanced. Therefore, current density is considered to have higher linear relationship with the efficiency parameters and hence is chosen as the main feature affecting the battery performance.



(a)



(b)

Fig. S1. Pairplots and coefficient of Pearson correlation for variables of current density, EE, VE, CE and capacity (a) pairplots of 4 variables including current density, VE, CE and capacity (b) coefficients of Pearson correlation heat map for the mentioned variables. Because of relatively higher linearity of current density with efficiency variables, current density was selected as the main feature.

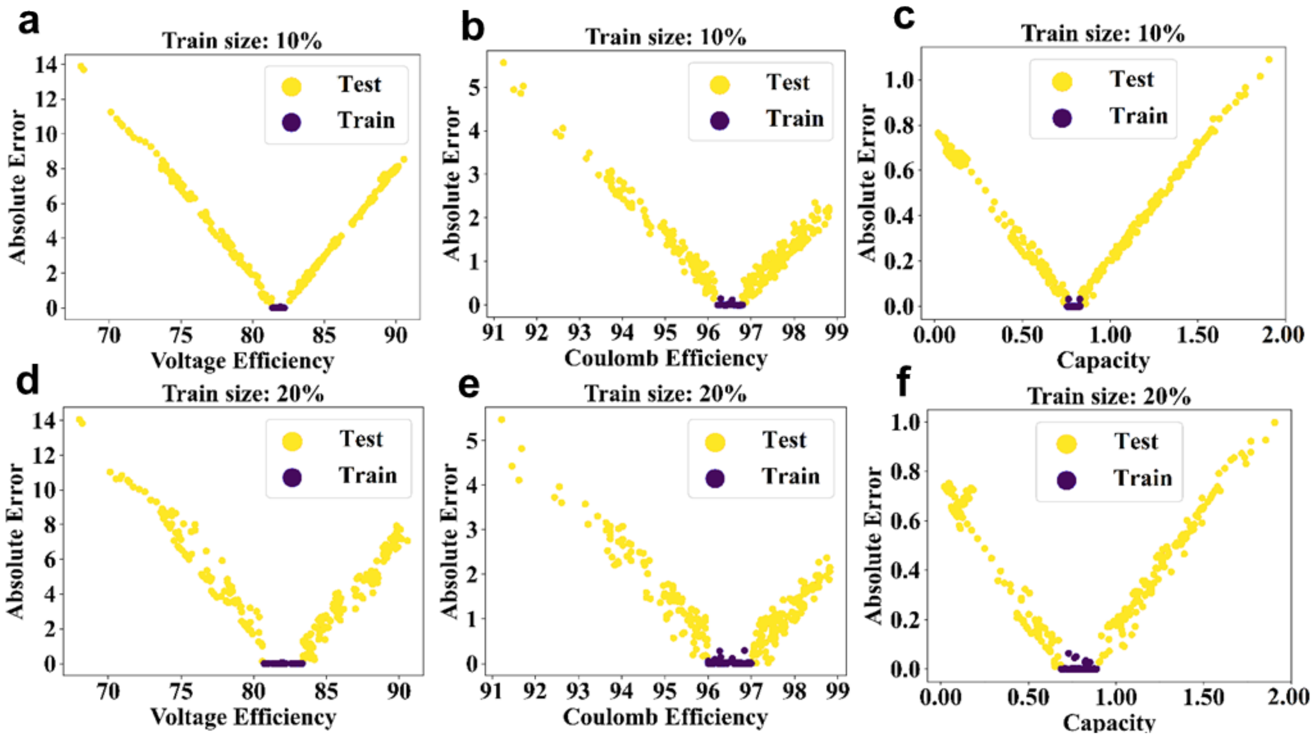


Fig. S2. Transferability of ML models for (a,d) voltage efficiency (b,e) coulomb efficiency and (c,f) capacity. The absolute error in ML model's prediction is plotted as a function of the actual efficiency value of the battery. As the train size increases, the values of absolute error decrease for each efficiency parameters which proves the transferability of our model

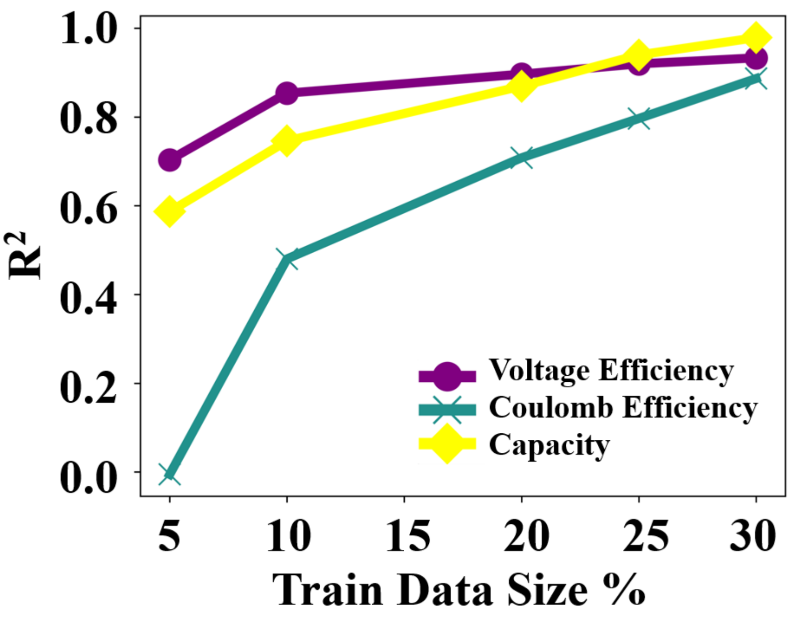
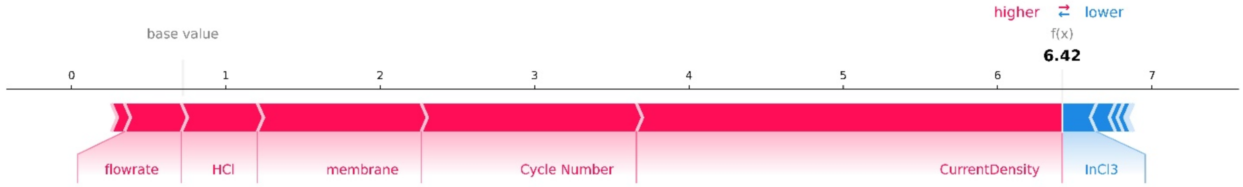
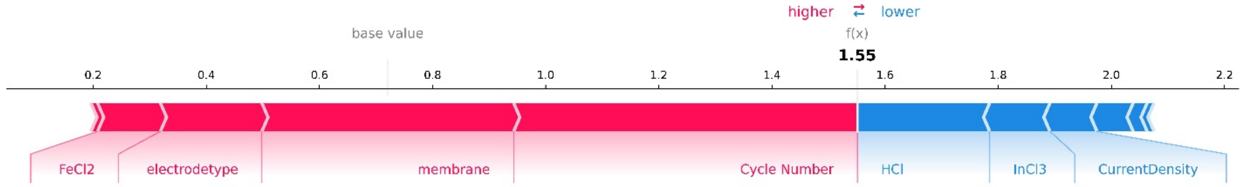


Fig. S3. Transferability of an ML model. The coefficient of determination R^2 for test data set is plotted against the number of training data for three models. The accuracy of model increases with increase in size of train set that implies on reliability of our model.

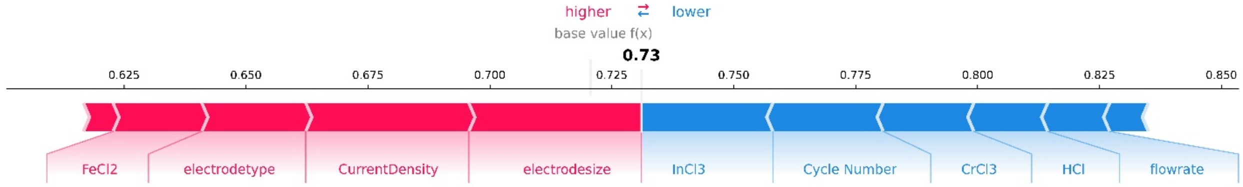
The SHAP force plots for performance parameters are plotted within fig. S4a-c with their corresponding average values provided in fig. S5. Based on the values, it is obvious that generally increase in electrolyte and catalyst concentration will result in performance enhancement on batteries. Furthermore, as we go further in cycles the voltage efficiency decreases while capacity and coulomb efficiency slightly increase. It is also notable that while current density is conversely proportional to all the performance parameters, it has the highest impact on voltage efficiency compared to other performance parameters. Other features such as electrode size and type either have minor contribution on the performance parameters or are qualitative parameters.



(a)



(b)



(c)

Fig. S4. SHAP Force Plots for (a) voltage efficiency (b) coulomb efficiency and (c) capacity

Feature name	Voltage Efficiency	Coulomb Efficiency	Capacity
Flow rate	-0.006056	0.012550	-0.012712
Electrode type	0.017893	0.000182	0.004519
Electrode size	0.002184	0.002788	-0.006652
Membrane	-0.028035	0.040592	-0.004009
Cr^{3+}	-0.018175	0.027569	0.011856
Fe^{2+}	0.001861	0.014300	0.002775
H^{1+}	0.195122	0.114809	0.013372
Bi^{3+}	0.066031	0.031968	0.005162
In^{3+}	0.018304	0.013299	0.003505
Current Density	-0.393186	-0.061147	-0.007983
Cycle Number	-0.035180	0.016836	0.004965

Fig. S5. SHAP values of features for each model, red color indicates increase the efficiency with increase in feature value while blue color indicates decrease in performance with increase in feature value. increase in electrolyte and catalyst concentration will mainly result in efficiency performance.

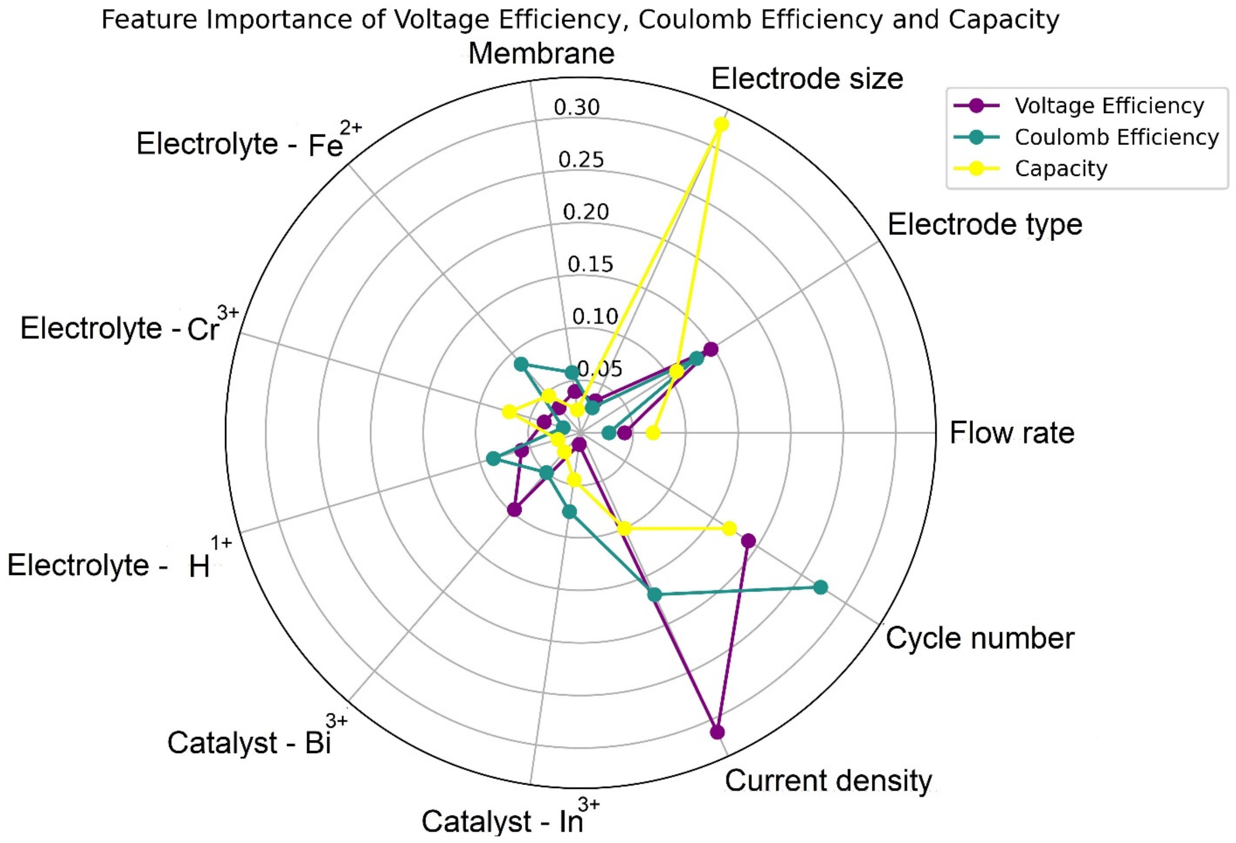


Fig. S6. depicts feature importance rank for all the utilized features. Firstly, cycle number primarily affects all the performance parameters particularly the coulomb efficiency. In addition, current density is the main feature influencing the voltage efficiency. For capacity, electrode size has the highest rank among the other variables.

The influence of each electrode type is demonstrated in fig. S7. While TCC can be considered as an electrode that simultanously enhances all the performance variables, TiN-3D GF has negative influence on all the performance parameters, leading to lesser efficiency. The rest of the electrode types do not possess universal influence on battery performance and their function must be separately investigated for each performance parameter.

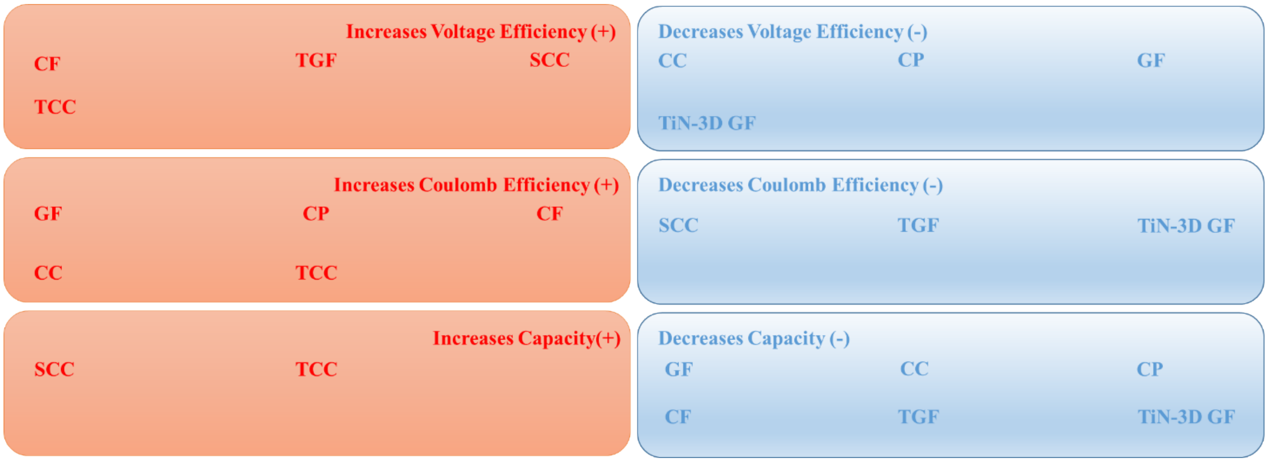

Increases Voltage Efficiency (+)
CF
TGF
SCC
TCC
Decreases Voltage Efficiency (-)
CC
CP
GF
TiN-3D GF
Increases Coulomb Efficiency (+)
GF
CP
CF
CC
TCC
Decreases Coulomb Efficiency (-)
SCC
TGF
TiN-3D GF
Increases Capacity(+)
SCC
TCC
Decreases Capacity (-)
GF
CC
CP
CF
TGF
TiN-3D GF

Fig. S7. Feature influence on the performance parameters. The model recommends utilization of TCC for the battery while discourages TiN-3D GF electrode to be utilized in the system.

Machine Learning

The equation of the (i) linear regression (LR) model to a data set (n features and m samples) is

$$Y = \beta_0 + X\beta \qquad (1)$$

In the context of a linear regression model, we have a rectangular input matrix X, comprising m rows and n columns, representing the feature data for each sample. The features X_{i1}, X_{i2}, X_{i3}, $X_{i4,}$ …, X_{i11} correspond to electrode size, electrode type, flow rate, etc., for the i'th sample. The corresponding output vector Y contains the performance efficiency data, with Y_i representing the performance efficiency for the i'th sample.

To perform linear regression, scikit-learn employs the concept of "ordinary least squares" (OLS). The LR model aims to minimize the sum of squared residuals (SSR) between the observed and predicted values in the given dataset. This is achieved by fitting the data with intercept vector β_0, comprising identical elements, and coefficient vector β, where β_j refers to the coefficient for the j'th feature.

$$SSR = \overset{min}{\beta} ||A||_2{}^2 = \overset{min}{\beta} ||(\beta_0 + X\beta) - y||_2{}^2 \qquad (2)$$

In the given equation, y represents the actual value matrix obtained from the dataset, while (β_0 + Xβ) corresponds to the predicted value matrix. The symbol || ||2 represents the modulus of the square root of the sum of squared elements in the column matrix. Apart from the linear regression model, three other regression models based on ensemble methods are utilized for data prediction. Ensemble methods combine base machine learning (ML) models to create an optimized predictive model. Two such models are the random forest (RF) and extra trees (ET) models, which belong to the averaging method. These models involve multiple decision trees (estimators) that independently make predictions, which are then averaged to provide an overall prediction for the dataset. Each decision tree in the RF and ET models recursively splits the MD dataset into nodes or subgroups. The splitting process starts with a parent node, which is divided into two children nodes. This process continues until the nodes reach a leaf node, which does not have any children. The prediction for a specific sample is determined by traversing through all the trained decision trees until the corresponding leaf node is reached. The main differences between the RF and ET models are:

a) The RF model uses bootstrap samples, which are sub-samples of the input data with replacement, whereas the ET model uses the entire input dataset.

b) When splitting a node, the RF model selects the best feature, while the ET model uses a random feature from a random subset of features of size $\sqrt{n}$.

To optimize the RF and ET models, several hyperparameters are tuned, including the number of estimators, maximum number of features, and maximum tree depth. Determining the appropriate values for these hyperparameters can save computational time, as further increasing these parameters may not significantly improve the model's accuracy. Additionally, setting a maximum tree depth can help prevent overfitting of the training dataset.

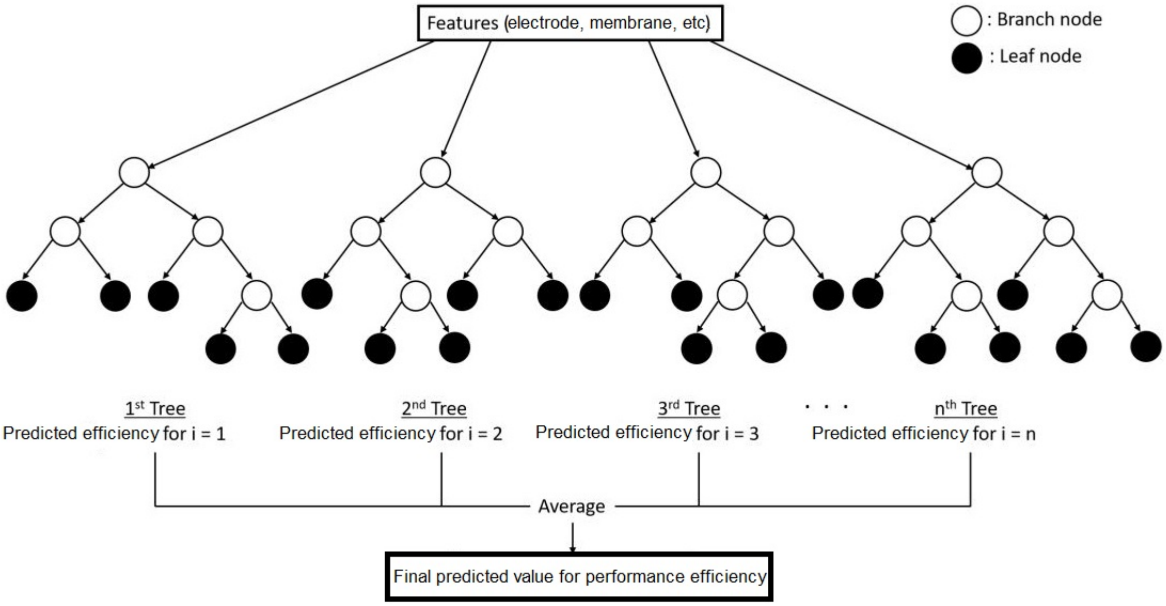

Features (electrode, membrane, etc)
○ : Branch node
● : Leaf node
1st Tree
Predicted efficiency for i = 1
2nd Tree
Predicted efficiency for i = 2
3rd Tree
Predicted efficiency for i = 3
. . .
nth Tree
Predicted efficiency for i = n
Average
Final predicted value for performance efficiency

Fig. S8. RF and ET models function schematics (for illustration purposes only four branches are shown) predicting the battery efficiency. The data set contains 11 features.

The last model, gradient boosting (GB) model is a boosting approach where estimators are built sequentially, taking into account the results of the previous estimator. In the case of the GB model, the estimators are decision trees. The training process of the GB model starts with calculating the average efficiency (y_{avg}) from the training dataset. Pseudo-residuals are then computed by taking the difference between the efficiency in the training data and the average value. The first decision tree is trained using these pseudo-residuals. The pseudo-residual obtained from the leaf node of the decision tree (PR_{i1}, where i represents the i'th sample and 1 denotes the first decision tree) is added to the average performance efficiency multiplied by a learning rate (α), resulting in the predicted efficiency from the first decision tree ($y_{pi1} = y_{avg} + \alpha\ .\ PR_{i1}$). After calculating the efficiency using the first decision tree, new pseudo-residuals are computed by taking the difference between the efficiency in the training data and the efficiency predicted by the first decision tree. These new pseudo-residuals are generally smaller than the initial pseudo-residuals. The second decision tree is then trained using these new pseudo-residuals. The pseudo-residual obtained from the leaf node of this tree (PR_{i2}) is added to the prediction from the first decision tree to obtain the prediction from the second decision tree ($y_{pi2} = y_{pi1} + \alpha\ .\ PR_{i2}$). This process continues to add decision trees sequentially, with each tree reducing the errors (pseudo-residuals) from the previous trees by incorporating a learning rate. The trained sequence of decision trees is then used to predict the efficiency for any new sample. The hyperparameters that are tuned in this process include the learning rate, number of estimators, maximum tree depth, and maximum number of features. It is important to mention that in naive gradient boosting, the learning rate is set to 1. The learning rate and the number of estimators are correlated. If the learning rate is less than 1, fewer corrections are made for each estimator, requiring more estimators. On the other hand, if the learning rate is greater than 1, more corrections are made per estimator, reducing the need for a large number of estimators.

To avoid the overfitting and assess the division of the dataset into training and testing samples, we examine the learning curve, which plots the prediction accuracy against the number of training samples. Figure S9 displays the learning curve and indicates that the mean cross-validated R^2 score, calculated through 10 cross-validation splits, reaches a plateau after 225 training samples. This finding suggests that the train-test split can be set at 75%.

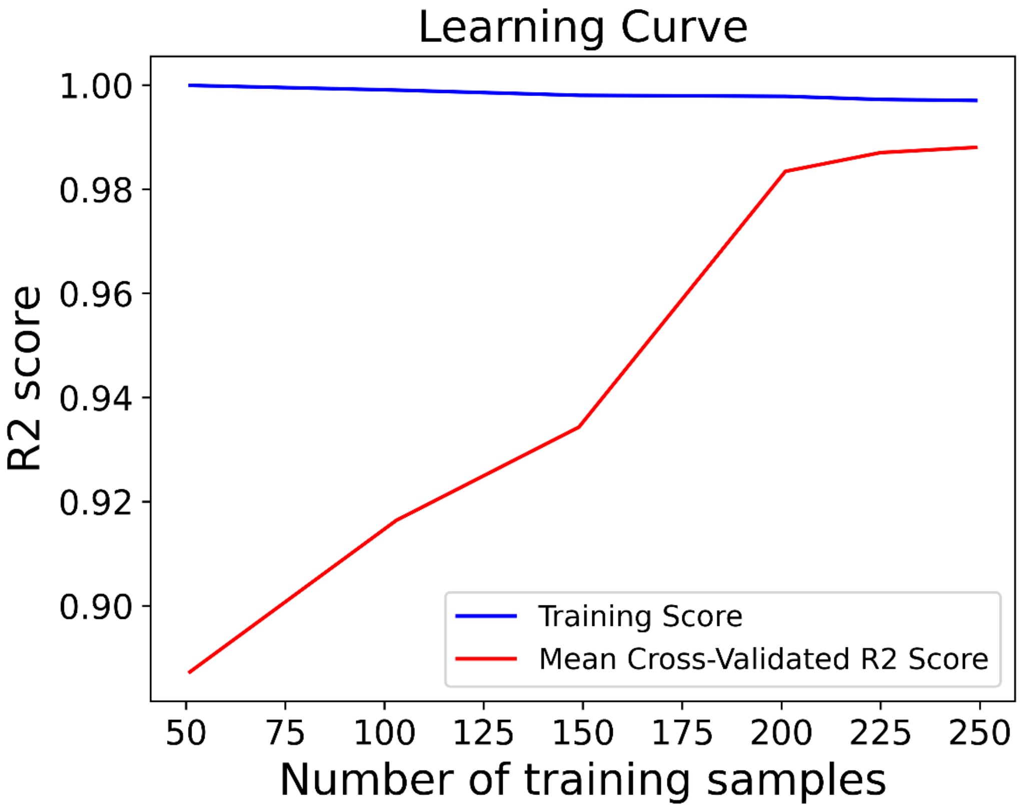


Figure S9. The learning curve for the model, displaying the relationship between the mean cross-validated R^2 score (MCRS) and the training score. The MCRS is determined by calculating the R^2 score across 10 cross-validation splits of the dataset, while the training score is the R^2 score obtained using the training samples. Based on the observations, the number of train size was selected to be 75%.

The hyperparameter optimization for each model is evaluated based on the Mean Cross-validated Root Squared Error (MCRS) across ten cross-validation splits. For the Random Forest (RF) and Extra Trees (ET) models, we observed that the accuracy saturates when the maximum depth of the trees reaches 8, as depicted in Figure S10. Typically, setting no restrictions on tree growth can lead to overfitting. However, in our case, efforts to limit tree growth resulted in a decrease in MCRS. Based on our findings, we determined that a maximum tree depth of 13 for RF and 10 for ET achieves the highest accuracy. Moreover, we identified the optimal number of estimators and maximum number of features for both RF and ET models. For RF, the values determined were 800 estimators and 5 maximum features. Similarly, for the ET model, the optimal values were found to be 100 estimators and 5 maximum features. We observed that reducing the maximum number of features had a substantial negative impact on model accuracy. Next, we assessed the

Gradient Boosting (GB) model. We found that to achieve good accuracy, a low learning rate is required as the number of estimators increases, as shown in Figure S11. Based on our analysis, the optimal learning rate and number of estimators were determined to be 0.1 and 600, respectively, for achieving the best accuracy in our work. Additionally, we found that a maximum tree depth of 2 yields the best results. It is worth noting that reducing the maximum number of features resulted in a significant decrease in the model's accuracy, highlighting the importance of maintaining a value of 5 for optimal performance.

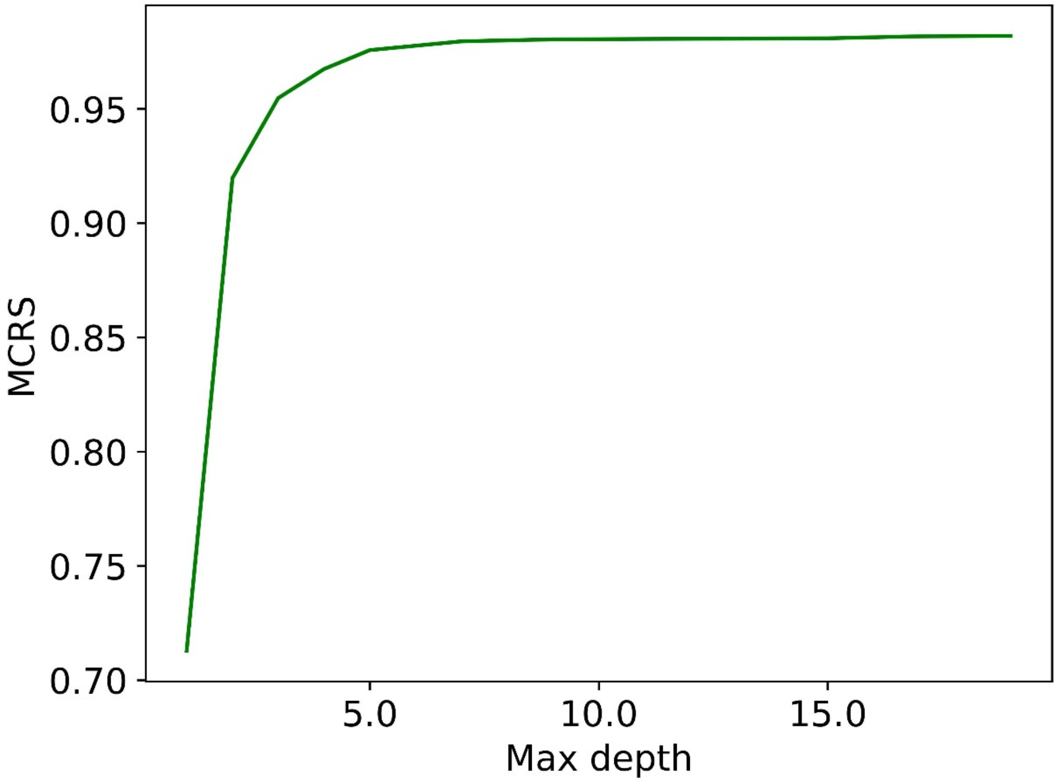


Fig. S10. The values of mean cross-validated R^2 score with regard to trees maximum depth of random forest model. Maximum depth was selected to be 13 for RF and 10 for ET based on the accuracy metrics.

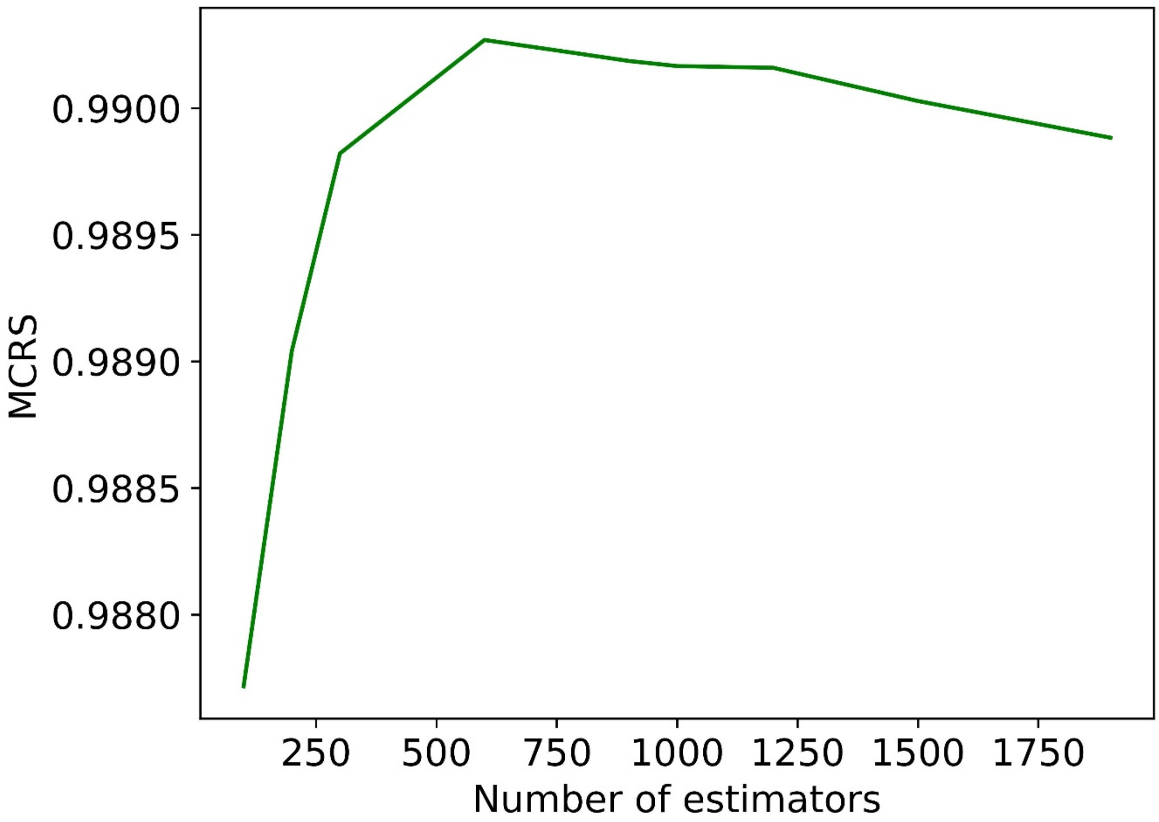


Fig. S11. The change in mean cross-validated R^2 score with regard to the number of estimators at learning rate of 0.1, maximum tree depth of 2 and maximum number of features of 5 utilizing the gradient boosting model. Number of estimators was chosen to be 600 for this model.

Active Learning

The exploration and exploitation strategy is employed to dynamically guide the discovery of favorable combinations, based on predictions and uncertainties. Exploration prioritizes combinations with higher uncertainty, driven by curiosity, while exploitation favors combinations with higher predicted performance efficiency, driven by perceived usefulness. Ultimately, the top five selected candidate materials' energy efficiency values are determined using the adopted methodology in every iteration step.

To generate a comprehensive benchmark dataset, we compiled over 300 data points on battery performance efficiency from existing literature. Subsequently, we conducted 10 iterations and generated 300 new datasets, encompassing 30 battery combinations with optimized efficiency, using the aforementioned framework.

Uncertainty proves to be a valuable metric when searching for battery compositions. It gauges the "distance" between an unexplored composition and the known dataset. A larger "distance" indicates higher uncertainty assigned to that specific composition. The exploration and exploitation strategy is a fundamental concept in active learning [1, 2], representing a trade-off mechanism that facilitates incremental progress towards a defined objective within an unfamiliar setting. This strategy operates based on an objective function that guides decision-making. In this work, we use the rank of performance multiplied 0.9 and the rank of uncertainty multiplied with 0.1 as the determination strategy. Specifically, when striving for optimal efficiency, the objective is to identify compositions with the highest achievable efficiency, which can be mathematically expressed as an optimization problem:

$$\begin{aligned}&find\ input\ x \in H\\ &to\ the\ function\ f{:}H{\to}R\\ &such\ that\ x = \arg min_{x\,\in\,H}\{f(x)\}\end{aligned}$$

where H represent the space of compositions, f is the function that connects a composition with its corresponding properties and efficiency. R encompasses the range of all possible efficiency values. The selection of a composition, denoted as x, is influenced by both the average and uncertainty associated with efficiency values. Exploration prioritizes compositions with higher uncertainty, while exploitation favors compositions with higher predicted efficiency.

References

[1] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research,* vol. 4, pp. 237-285, 1996.

[2] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning,* vol. 3, pp. 9-44, 1988.