ACS OMEGA

Open Access

http://pubs.acs.org/journal/acsodf

Article

Machine Learning-Based Prediction of Cyclic Voltammetry Behavior of Substitution of Zinc and Cobalt in $BiFeO_3/Bi_{25}FeO_{40}$ for Supercapacitor Applications

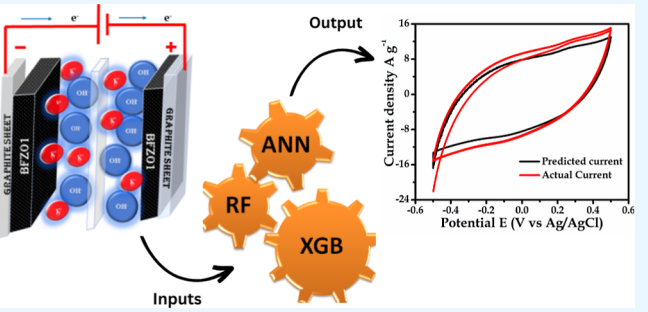
Abhilash Ravichandran, Valliappan Raman, Yogapriya Selvaraj, Prabhavathy Mohanraj,* and Hemalatha Kuzhandaivel

Cite This: *ACS Omega* 2024, 9, 33459−33470

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Artificial intelligence and machine learning have become indispensable tools across various disciplines in the present century. In that way, the role of artificial intelligence and machine learning in energy storage devices was investigated. As a preliminary study, the data derived from electrochemical studies were used for the prediction. The prediction of current from cyclic voltammetry (CV) studies was undertaken for bismuth ferrite (BFO), substitution of zinc in BFO (BFZO), and substitution of cobalt in BFO composite (BFCO). CV is a vital electrochemical technique used for studying the electrochemical behavior of any material. The electrochemical study provides insights into the energy storage behavior of the material through the specific capacitance. The machine learning models, such as Artificial Neural Network (ANN), Random Forest (RF), and XGBoost (XGB), are trained and implemented to predict current at different scan rates. These models are trained and validated using the data collected from a CHI 600E electrochemical workstation. Multiple trials of experiments were performed to build the most optimum model for the material. The predicted values provide promising results and align well with the experimental data. The XGBoost, ANN and RF models perform well for the CV data set with an average testing accuracy >97%. Also, a meta-model was created using stacking of the above three machine learning models which further improved the predictive performance, achieving a slightly higher average testing accuracy of over 97.73%. The outcomes from the models can promote the development of machine learning applications in the field of electrochemistry and provide insights into optimizing supercapacitor performance and design through data-driven approaches.



1. INTRODUCTION

Energy is one of the most important needs for the twenty-first century and it is a part of human existence. The change in climatic conditions and the depletion of fossil fuels drives us to look for new energy harvesting and storage technology. Renewable energy technology, such as solar, wind, tidal, and geothermal, is an effective alternative due to its properties, such as cost-effective, high performance, and ecological benign. The harvested energy through renewable technology is stored in the energy storage devices, such as batteries and supercapacitors (SCs). Batteries have the properties of high energy density, whereas SCs have the properties of high power density. So, the present focus is on improving the energy density of SC devices. SCs are used in commercial machinery, digital cameras, hybrid electric cars, etc.[1,2] Based on the mechanism, SCs are classified into three types (i) electrical double layer capacitor (EDLC)—the carbon-based material utilizing nonfaradaic physical adsorption at the electrode−electrolyte interface, (ii) pseudocapacitor—the metal oxides and conducting polymer-based material utilizing faradaic reaction or redox reaction on the electrode and electrolyte interfaces, and (iii) hybrid capacitor—the combination of EDLC and pseudocapacitor electrodes.[3,4] The transition-metal oxides, such as NiO, MnO_2, Co_3O_4, V_2O_5, ZnO, Fe_3O_4, etc, having high specific capacitance, good redox reversibility, good cyclic stability, low cost, and nontoxic material are used as electrode materials for SCs.[4] In the present study, bismuth ferrite (BFO) and substitution of zinc/cobalt BFO are used as the electrode material, which were subjected to electrochemical studies. The data derived from the electrochemical studies were subjected to the prediction of current through modeling. Artificial intelligence (AI) has emerged as a transformative tool in the

Received: December 29, 2023
Revised: June 21, 2024
Accepted: June 26, 2024
Published: July 26, 2024

realm of SC technology. By utilizing AI methodologies, such as machine learning (ML) and deep learning, researchers were able to enhance the performance, efficiency, and reliability of SCs. The SC performance and its cyclic voltammetry (CV) behavior were modeled using an artificial neural network (ANN) and random forest algorithm for the Co−CeO_2/rGO nanocomposite,[5] which motivated us to expand the approach to our own material composite. There have been numerous studies focusing on the assessment of SC performance using intelligent models like ANN.[6] These investigations employ ANN to analyze and predict the electrochemical behavior and characteristics of SCs. Also, ML algorithms are utilized to establish a correlation between intrinsic features and the cyclic stability of SCs.[7] The advancements and progress made in applying ML techniques to predict the capacitance and estimate the remaining useful life of SCs.[8,9] Moreover, the ML approach was implemented on various physicochemical features to predict the specific capacitance of a material.[10] Traditionally, experimental techniques, such as CV tests, are employed to characterize the electrochemical properties of materials. It provides valuable information about the current−voltage relationship and energy storage capability of the SC electrode.[2] However, conducting extensive experimental studies can be a time-consuming and resource-intensive work.

To overcome these challenges, we propose the use of AI and ML techniques to predict the current in CV curves for the substitution of the zinc and cobalt BFO composite. ML models, including RF, XGBoost, and ANN, have been trained and implemented to learn the complex relationships between the input parameters (scan rates) and predict the corresponding output variable (current). The main objective of this study is to assess the performance of these ML models in predicting the electrochemical behavior of the zinc and cobalt in the BFO composite. The accuracy of the predictions is assessed by comparing the model's results with experimental values. Furthermore, the study aims to identify the most effective ML model for accurately predicting the electrochemical properties of the composite. The successful application of ML in this field can have significant implications for the development of SCs. By leveraging AI and ML techniques, we can gain valuable insights into the relationships between the material composition, electrode design, and electrochemical performance. This data-driven approach can guide the optimization of SC performance and facilitate the design of more efficient energy storage devices.

2. METHODOLOGY

2.1. Electrode Preparation and Electrochemical Measurements. The electrochemical measurements of BFO, BFZO, and BFCO composites were performed in the three electrode system using 3 M KOH as the electrolyte. The active electrode materials were prepared as a paste by mixing the active material (BFO, BFZO, and BFCO) in an 80:10:10 ratio with 10% PVDF as a binder and 10% acetylene black. This mixture was thoroughly combined using NMP solvent until it formed a fine paste. The mixture was coated on graphite sheet substrate of 1 × 1 cm dimension. The substrate underwent precleaning with acetone and deionized water, then dried at 80 °C. The prepared paste was coated on the substrate and dried at 60 °C for 12 h to remove the solvent. The prepared active material acts as a working electrode, a platinum wire acts as a counter electrode, and Ag/AgCl acts as the reference and counter electrodes, respectively. Electrochemical measurements, such as CV, galvanostatic charge−discharge (GCD), and electrochemical impedance spectroscopy were analyzed using a CH analyzer (Model: CHI 600E) (Figure 1).

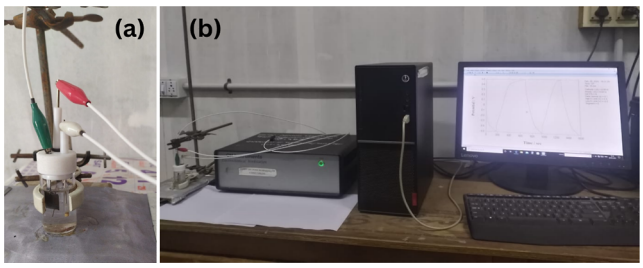


Figure 1. (a) Three-electrode system and (b) experimental setup used for collecting CV data from an electrochemical workstation.

The CV data of the SC material are collected using a three-electrode system and an electrochemical workstation. Subsequently, the collected data are sent to the computer for data set creation. Eqs 1 and 2 were used to calculate the specific capacitance (C_{sp}) from the CV curve and the GCD curve, respectively.

$$C_{sp} = \left(\int IdV \right)/2\text{mv}\Delta V \quad (1)$$

$$C_{sp} = (I \times \Delta t)/(m \times \Delta V)) \quad (2)$$

In the equation, C_{sp} stands for specific capacitance (F g^{-1}), $\int IdV$ for integrated area of total charge corresponding to potential window, ν for scan rate (mV s^{-1}), m for mass of active material (g), V for potential window (V), I for discharge current density (A g^{-1}), and t' for discharging time (s).[2]

The potential window of BFO and BFZO composites is −0.5 to 0.5 and −0.7 to 0.5 V for BFCO composites in 3 M KOH as a electrolyte solution. The CV curves of all the composites were scanned at a scan rate of 10−50 mv s^{-1} and the GCD curves were measured at 1−5 A g^{-1} of current density. Due to the highly time-dependent nature of GCD data, we exclusively used CV data for modeling.

2.2. Data Preprocessing. The open source software we used to create ML models was Jupyter Notebook 6.5.2 with the Python 3 programming language. Two different data sets are created for the study. One data set includes current and potential values for different scan rates at different concentrations (Data set 1). The other data set includes current and potential values for one scan rate at a single concentration (Data set 2). The data sets are converted to a structured format having rows and columns. Further, based on the CV graph, the redox curve is classified into oxidation and reduction. The dummy variables are used to represent the column "Oxidation" in the Data sets. One or 0 indicates whether the curve is at oxidation or reduction, respectively.[5] The one-hot encoding method was used to differentiate the substitution of zinc and cobalt BFO data sets (Figure 2).

2.2.1. Dataset. The datasets were created for BFO, substitution of zinc in BFO, and substitution of cobalt in BFO at three different concentrations of $Zn(NO_3)_2 \cdot 6H_2O$ at 1.5 mmol (BFZO1), 2.5 mmol (BFZO2), 3.5 mmol (BFZO3), 1.5 mmol (BFCO1), 2.5 mmol (BFCO2), and 3.5 mmol (BFCO3).

2.2.2. Dataset 1. CV data for BFO, BFZO1, BFZO2, BFZO3, BFCO1, BFCO2, and BFCO3 are taken at different

https://doi.org/10.1021/acsomega.3c10485
ACS Omega 2024, 9, 33459−33470

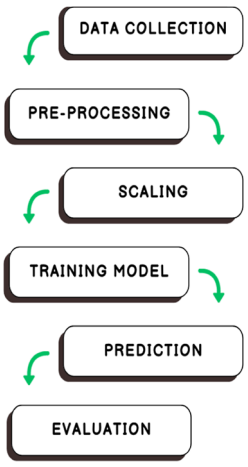


Figure 2. Schematic representation of ML pipeline/steps involved in model construction.

scan rates (10, 20, 30, 40, and 50 mV s^{-1}). The size of the data set is 216200 rows and 7 columns (216,200 × 7).[2]

2.2.3. Dataset 2. CV data for BFZO3 is taken at 60 mV s^{-1} scan rate to demonstrate a comparison of specific capacitance (predicted vs experimental). The size of the data set is 2000 rows and 7 columns (2000 × 7)[2]

After preprocessing, the whole data set was divided into training (80%) and validation part (20%). Here, the validation part is addressed as the testing part and it is used to assess the different model's performance and tune hyperparameters. This helps to ensure that the models generalize well to unseen data. Data set 2 is the primary data set for testing the models on real-world data. Based on the nature of the above data sets, Deep learning and ensemble ML models were trained and used to predict output current from Data set 1. Tensorflow and Scikit-learn libraries from Python 3 were used to build these models and evaluate them under different evaluating metrics.

3. ML MODELS

The CV profiles of BFO and substitution of zinc and cobalt BFO composite are trained under different ML algorithms. A deep learning model, such as ANN, and ensemble models, such as RF and XGBoost, are trained and optimized for accurate prediction. The choice of these algorithms over others was based on their respective strengths and characteristics. ANNs were employed to capture intricate patterns within the data, RF was chosen for its robustness and its capacity to yield insights into feature importance, while XGBoost (XGB) was utilized due to its high performance and effectiveness in managing structured data. Differences in the prediction mechanisms arise among these models primarily because of their distinct algorithms and methodologies. ANNs employ interconnected layers of nodes to transform input features, capturing intricate patterns through iterative training. RF utilizes independently constructed decision trees, combining their predictions through majority voting or averaging. In contrast, XGBoost (XGB) employs a sequential ensemble approach, building decision trees sequentially to correct errors, optimizing a predefined loss function. The used evaluation metrics for model validation are root mean square error, MSE, and coefficient of determination/*R*-squared value.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (yi - \overline{y}_i)^2}{n}} \tag{3}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n} (yi - \overline{y}_i)2 \tag{4}$$

Also, $\mathrm{MSE} = \mathrm{RMSE}^2$

$$R\ \text{squared} = 1 - \left(\frac{\mathrm{SSR}}{\mathrm{SST}}\right) \tag{5}$$

where, n is the number of instances in the dataset, y_i is the true target value for instance i, and $\overline{y}_i$ is the predicted value for instance i, SSR (sum of squared residuals) quantifies the total variability that remains unaccounted for by the regression model, while SST (total sum of squares) measures the overall variability present in the dependent variable.[11,12]

The inputs to the ML models are potential, oxidation/reduction, Zn/Co concentration, scan rate, zinc (yes/no), cobalt (yes/no), and the output is current. Figure 3 shows a schematic diagram that shows the collection of input from the electrodes, processes through models, and produces a predicted output.

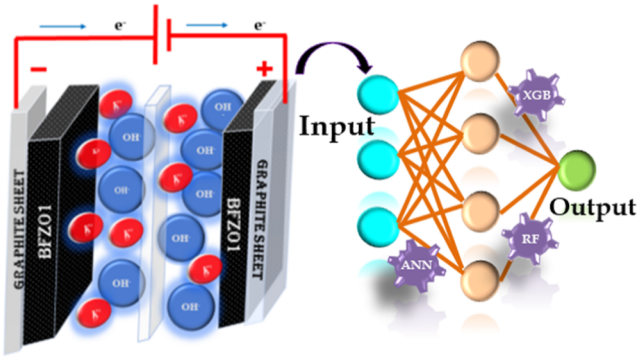


Figure 3. Input, process, and output diagram for current prediction.

3.1. Artificial Neural Network. ANNs consist of a network of interconnected artificial neurons organized into layers, commonly referred to as nodes. The information passes through the network in the forward direction, with each neuron receiving inputs. After processing, the output of each layer is transmitted to the next layer, where connections between neurons carry information weighted according to their respective strengths. This process allows ANNs to capture intricate patterns and nonlinear relationships within the data, enabling them to learn and adapt to complex information structures. ANNs are typically organized into layers, which are composed of multiple neurons. An ANN typically comprises three primary layer types: the input layer, one or more hidden layers, and the output layer.[13,14] The ANN architecture determines the model's capacity to learn complex patterns from the data. Parameters like the number of layers, the number of neurons per layer, the type of activation functions, and the connections between neurons profoundly impact the network's ability to generalize and perform well on unseen data. A deeper architecture with more layers can capture intricate relationships in the data but may also increase the risk of overfitting. On the other hand, a shallow architecture might not capture enough complexity to effectively learn the underlying patterns. Thus, it is crucial to select the appropriate

https://doi.org/10.1021/acsomega.3c10485
ACS Omega 2024, 9, 33459−33470

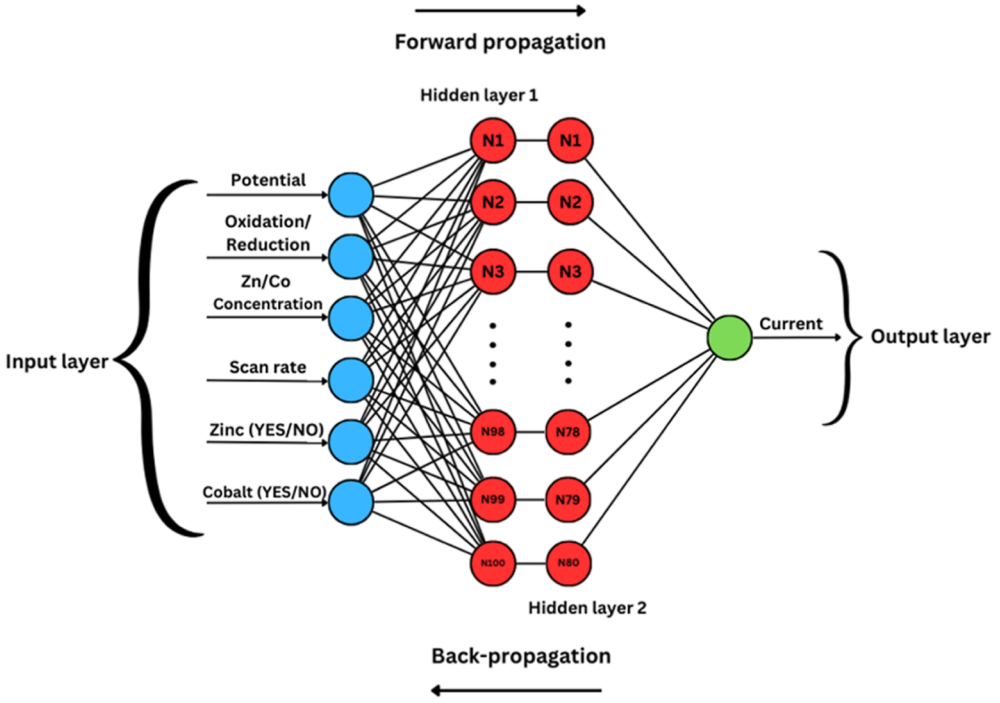


Figure 4. Neural network architecture for CV behavior of the composite.

ANN architecture based on a balance between model complexity and the complexity of the underlying data patterns.[25,26] The neural network architecture used for current prediction is given in Figure 4.[27]

By applying the `StandardScalèr from scikit-learn Python library, the input data is scaled in such a way that each feature has zero mean and unit variance.[15]

$$\text{Standardized feature} = \frac{(\text{feature value} - \text{mean})}{\text{standard deviation}} \quad (6)$$

We used TensorFlow (a ML framework for developing neural networks) for building ANN. In TensorFlow, a linear transformation is commonly implemented using a dense layer or fully connected layer. When provided with an input vector "x", a bias vector "B", and a weight matrix "W", the output "y" can be computed as follows

$$y = Wx + B \quad (7)$$

The activation function applied to the hidden layers in this work is called the rectified linear unit (ReLU) activation function, which is given as

$$\text{ReLU}(x) = (0, x) \quad (8)$$

where "x" is the input value to the activation function.[16,17]

The ReLU function operates by returning the input value if it is greater than or equal to zero and returning zero for any negative input value

$$\text{ReLU}(x) = x, \quad \text{if } x >= 0$$
$$0, \quad \text{if } x < 0 \quad (9)$$

The output layer uses a linear activation function for regression.[16] The computational intensity of ANNs is a drawback, particularly evident during the training of deep architectures.

3.2. RF Algorithm. RF is a predictive modeling technique that utilizes an ensemble of decision trees to generate accurate predictions. The concept underlying ensemble learning is that by combining the predictions of multiple models, it is possible to enhance the overall performance and generalization ability of the ensemble model compared to using a single model. In RF, an ensemble of decision trees is trained on different subsets of the data, and their individual predictions are aggregated to make the final prediction. RF builds an ensemble of decision trees using a technique called bagging (bootstrap aggregating), where each decision tree in the ensemble is trained on a randomly sampled subset of the training data, with replacement (bootstrapping) allowing them to learn from diverse data subsets. Bagging used all predictor variables to create random splits in the bootstrapping process which created highly correlated trees. So, we used RF, which creates subsets of predictor variables to split different trees for each model. In this way, we reduced the high variance after aggregation. During tree construction, each split in a decision tree is chosen from a random subset of features.[14] M out of P total predictor variables, which helps to introduce randomness and reduce correlation among trees. The number of predictor variables 'M' utilized to construct various trees from a total of 'P' variables is

$$M = P/3 \quad (10)$$

The ultimate prediction of the RF is derived by taking the average of predictions made by each individual tree within the ensemble. That is

$$\text{aggregated prediction} = \frac{(\text{prediction 1} + \text{prediction 2} + \ldots + \text{prediction } n)}{n} \quad (11)$$

where prediction 1, prediction 2, ..., prediction n represents the predictions done by each individual decision tree in the RF and

https://doi.org/10.1021/acsomega.3c10485
ACS Omega 2024, 9, 33459−33470

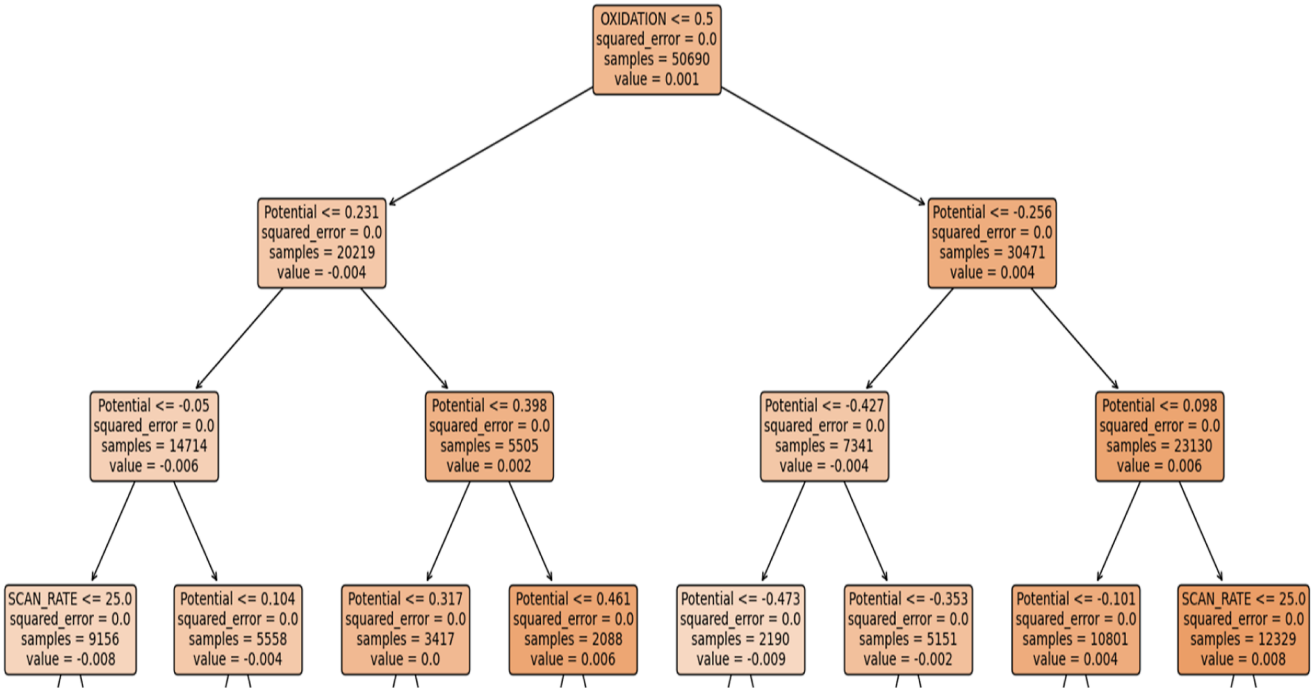


Figure 5. Decision trees in the RF model for CV behavior of the composite.

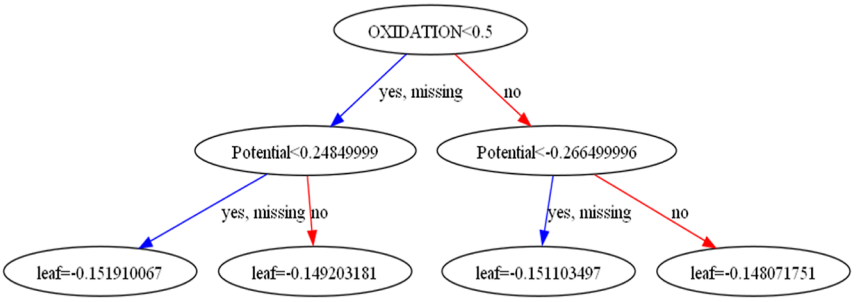


Figure 6. First tree in the XGBoost model for CV behavior of BFO composite.

the variable "n" denotes the total count of decision trees within the ensemble.[18] An advantage of RF is that it reduces overfitting by combining the predictions of multiple trees trained on different subsets of the data, thereby improving the model's generalization performance, a drawback of RF is that interpreting individual trees within the ensemble can be challenging, limiting its transparency and interpretability (Figure 5).

3.3. XGBOOST. Extreme gradient boost regression is commonly abbreviated as "XGBoost" regression.[19] XGBoost belongs to the family of gradient-boosting algorithms. Gradient boosting sequentially combines multiple weak learners into an ensemble method. Each new model is trained to correct the errors of the previous models, leading to a robust and accurate predictive model.[20] XGBoost uses the gradient and Hessian of the loss function to optimize the model during training. The gradient represents the derivative of the loss function with respect to the predicted values, while the Hessian represents the second derivative. These values are used to update the model parameters in each boosting iteration.[20,21]

$$\text{Objective function} = \text{loss function} + \text{regularization term} \tag{12}$$

XGBoost incorporates regularization components into the objective function to manage model complexity effectively and prevent overfitting. The regularization terms can be either L1 regularization (Lasso) or L2 regularization (Ridge).

$$L1 \text{ regularization term} = \lambda \times \sum_{i=1}^{n} |w| \tag{13}$$

$$L2 \text{ regularization term} = 0.5 \times \lambda \times \sum_{i=1}^{n} w^2 \tag{14}$$

where λ serves as the penalty factor that controls the strength of the regularization and w represents the model parameters.[22,23]

XGBoost employs the Taylor expansion to approximate the loss function and compute the gradient and Hessian efficiently. The Taylor expansion approximates the loss function as a second-order polynomial around a specific point. This approximation simplifies the computation of the gradient and Hessian.[24] An advantage of XGBoost (XGB) over other models is its built-in mechanisms for handling missing values and efficiently managing large data sets. However, like RF, interpreting XGB models can be challenging, especially when

https://doi.org/10.1021/acsomega.3c10485
ACS Omega 2024, 9, 33459−33470

dealing with a large number of trees in the ensemble (Figure 6).

3.4. Meta-Model. A meta-model is a higher-level model that learns to integrate predictions from multiple base models (ANN, RF, and XGB) to make final predictions. It is a second-level model trained on the outputs (predictions) of other models (Figure 7).

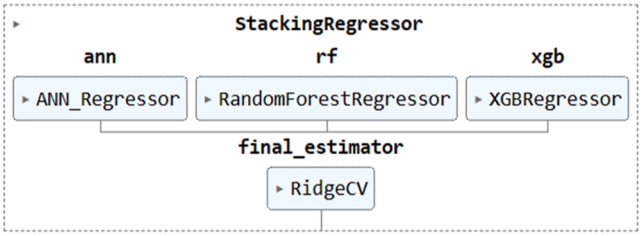


Figure 7. Meta-model (StackingRegressor) structure.

A meta-model was created using stacking of 3 machine-learning models. In the stacking ensemble method, base models, such as ANN, RF and XGB (level-0 models), are trained independently on the data set, with each model generating predictions. These predictions serve as input features for the meta-model (level-1 model), which learns to combine them to produce the final prediction. Stacking offers considerable flexibility in selecting both the base models and the meta-model, allowing for the exploration of different configurations to optimize performance.[28] Here, we used RidgeCV as the final estimator. RidgeCV (ridge cross-validation) is a regression algorithm that combines the ridge regression technique with cross-validation to automate the selection of the optimal regularization parameter. Additionally, it effectively mitigates overfitting by shrinking coefficient values toward zero, promoting simpler and more interpretable models.[29]

4. RESULTS AND DISCUSSION

4.1. Current Prediction Model. The output current is predicted for CV behavior of BFO and BFO substitution with zinc using features, such as potential, redox nature, substitution concentration, and scan rate, as the input.

4.1.1. ANN Model. The CV profile data set for different scan rates were divided into training and testing sets, with a distribution of 80% for training and 20% for testing (validation). The selection of training and testing data sets was random and unbiased. Our ANN model has 1 input layer, 2 hidden layers, and 1 output layer. The number of nodes in each hidden layer is 100 and 80. The ANN architecture was chosen after multiple experimentation and iterative refinements.

Table 1 shows the variation of *R*-squared and RMSE values for different ANN layers at a constant learning rate of 0.001. Here, we have chosen 10,080 as the first and second hidden layer neurons as the *R*-squared accuracy and RMSE values of test and train data sets are very close, which indicates that the architecture strikes a balance between model complexity and generalization ability for unseen data. The nodes of the hidden layers were also selected based on multiple trails. The parameters of the ANN model is shown in Table 2.

Table 2. Hyperparameter Table for ANN Model

ANN parameters	number/type
input layer nodes	6
output layer nodes	1
no of hidden layers	2
hidden layer 1 nodes	100
hidden layer 2 nodes	80
hidden layer activation function	ReLU
output layer activation function	linear activation function
learning rate	0.001
batch size	32
no of epochs	100

Hyperparameter tuning is the process of systematically searching for the optimal hyperparameters for a machine-learning model, and it involves selecting the best combination of hyperparameters to maximize the model's performance on the data set. The goal of hyperparameter tuning is to strike a balance between model complexity and its ability to generalize.

Each chosen hyperparameter such as no of hidden layers, no of nodes/neurons, type of activation function, learning rate, batch size, and no of epochs plays a critical role in determining the performance and behavior of an ANN model. These hyperparameters are tuned using testing data (20% of Data set 1) until we get a similar test and train accuracy and error values. A lower learning rate of 0.001 and smaller batch sizes of 32 in our deep learning model were chosen to facilitate smoother convergence, better generalization, and efficient exploration of the solution space while reducing the risk of overfitting and memory requirements. The reduction in the loss function curve was observed for both training and testing data. The model training was stopped immediately after observing an increase in test loss. Model training was also stopped if no improvement was found in reducing error or increment in accuracy.[6]

Figure 8 shows that the decrease in the error of the test and train set occurs with an increase in the number of epochs. The model training was stopped after 100 epochs. After various hyperparameter tuning, the best parameters for the model are noted (Table 2). The predicted test samples are validated

Table 1. Model Performance Summary Table for Various ANN Architectures

no of hidden layers	RMSE test	*R* square test	RMSE train	*R* square train	learning rate
120,60,30	0.0010773947784315	0.9786377990772107	0.001063727760395827	0.9790838149660833	0.001
60,30,10	0.0011061341722161722	0.9774829298207669	0.0010870158009392782	0.9781579597694783	0.001
120,60	0.0011574612210172692	0.9753447641225788	0.0011422561867760876	0.9758815981701627	0.001
100,80	0.001129789056085967	0.9765095687547274	0.0011254247366706952	0.9765871435774782	0.001
60,20	0.0011373289763688778	0.976194984379553	0.0011155516759022465	0.9769961315947602	0.001
120	0.0012768575041076568	0.969995863742859	0.0012714935816441153	0.9701152358286212	0.001
50	0.0013625685636169773	0.9658325165642594	0.0013509776980902821	0.9662621154254316	0.001

https://doi.org/10.1021/acsomega.3c10485
ACS Omega 2024, 9, 33459−33470

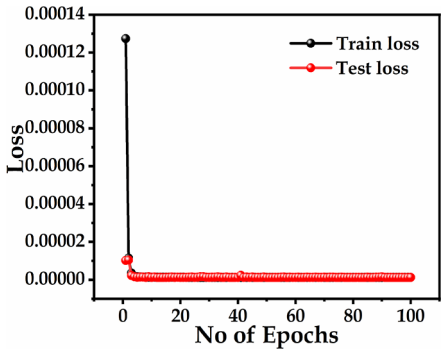


Figure 8. Variation of training and testing loss curves for different number of epochs for the neural network model.

using a 4-fold cross-validation technique for credibility. Cross-validation *R*-squared scores are 0.977508, 0.977669, 0.976918, and 0.977271.

The experimental and predicted target values for current were compared for analyzing the model performance, Figure 9 shows the actual vs predicted current graph for train and test data. The closer the points toward the fit line in the ANN model is indicative of accurate prediction.

4.1.2. RF Model. The RF algorithm was trained and tested using Data set 1. To gain insights into the impact of different features on the model's performance, we noted the feature importance for ensemble learning models. Figure 10 shows the percentage of feature importance in the final RF model. The potential (53.1%) and oxidation (32.5%) features contribute majorly to model prediction in the RF model. After train and test split, the model was trained under different number of trees. Then, the predicted values are examined and studied under different hyperparameters. The optimal model was achieved through fine-tuning various hyperparameters associated with it.

Table 3 shows the hyperparameters associated with the RF model, such as n_estimators, max_depth, min_samples_split, and min_samples_leaf, which are tuned to optimize the performance of RF model. From the graphs (Figures 11 and 12), it was observed, that the model's predictive accuracy improved a lot after hyperparameter tuning from defaults. Here, the obtained optimum values may not necessarily correspond to the global minimum of the optimization space but rather to a local minimum where the model achieves satisfactory performance. Reaching the global minima of the optimization space is challenging and often not guaranteed, especially in complex and high-dimensional optimization problems. So, our ML models strive to achieve a satisfactory

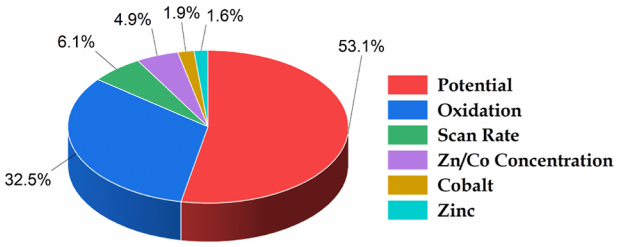


Figure 10. Feature importance (%) of independent variables in prediction for the RF Model.

Table 3. Hyperparameter Table for the RF Model

RF parameters	number/type
N_estimators (no of trees)	40
Max_depth	11
Min_samples_split	2
Min_samples_leaf	1

performance rather than reaching the absolute global minimum. Here, the lower values of min_samples_split and min_samples_leaf are used so that the trees get more flexible and can potentially capture more complex patterns in the data.

In Figure 11, as the number of trees increases, the accuracy improves consistently for both the test and train data and becomes constant after a certain number of trees. Additionally, the root-mean-square (RMSE) shows a decreasing trend as the number of trees increases, eventually stabilizing after reaching 40 trees. Figure 12 shows how the model accuracy varies with test and train data for different max_depth values. As we observe this graph, the testing accuracy starts to drop after a certain max_depth value, whereas the training accuracy keeps increasing. This indicates that after certain max_depth values, the trees are prone to overfit on the data set. So, the optimum chosen max_depth value for the RF model is 11. The minimum RMSE was 0.001155. The maximum testing *R*-squared obtained was 0.9754. The actual vs predicted plots for the RF model are given in Figure 13. The proximity of data points to the fitted line is indicative of the RF model's accuracy in making predictions. Also, the model was validated using 10-fold validation techniques to avoid overfitting of the model. Cross-validation *R*-squared scores are 0.97551536 0.97461391, 0.97474077, 0.97540098, 0.97756644, 0.97750095, 0.97519813, 0.97645132, 0.97316205, and 0.97561293.

4.1.3. XGBoost Model. **The boosting process iterates sequentially, with each subsequent model aimed at rectifying the errors of its predecessors.** Similar to ANN and RF, XGBoost model also trained with 80% training and

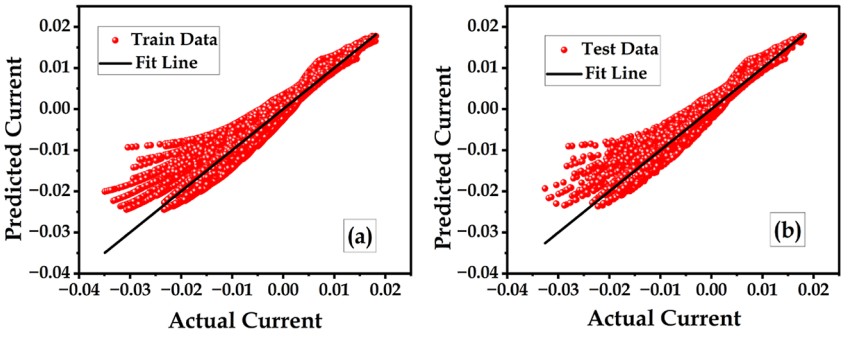


Figure 9. Comparison of actual vs predicted current values (a) train data and (b) test data for the ANN model.

https://doi.org/10.1021/acsomega.3c10485
ACS Omega 2024, 9, 33459−33470

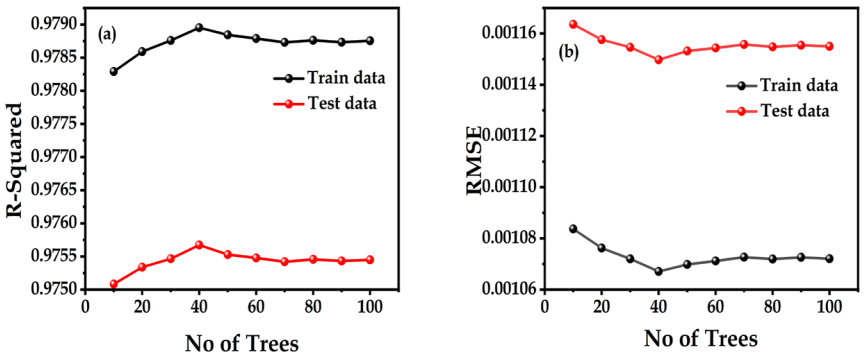


Figure 11. (a) *R*-Squared and (b) RMSE values for different number of trees for the RF model.

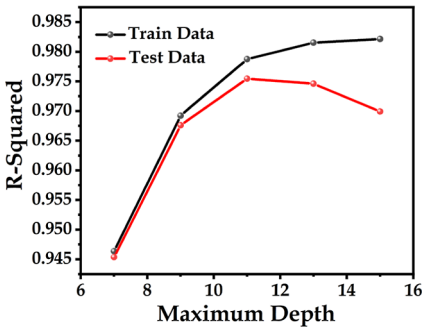


Figure 12. Variation in *R*-squared values for testing and training of CV profile for different maximum depths of RF trees.

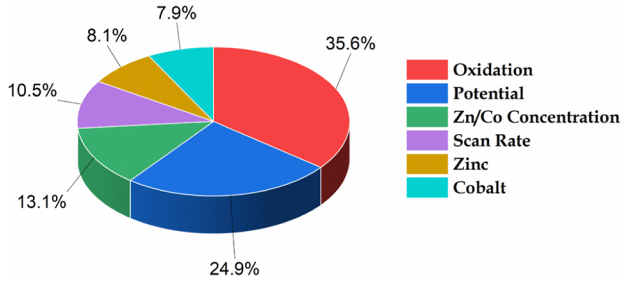


Figure 14. Feature importance (%) of independent variables in prediction for the XGB Model.

Table 4. Hyperparameters Table for the XGB Model

XGB parameters	number/type
N_estimators (no of boosting rounds)	100
Max_depth	6
learning rate (eta)	0.1

tested with 20% over CV profile data. The predicted current values are validated using RMSE, MSE, and *R*-squared metrics. Figure 14 shows the percentage of feature importance in the final XGB model. Unlike RF, in XGB model the oxidation (35.6%) feature contributes more than other features followed by the potential (24.9%), zinc (or) cobalt concentration (13.1%), scan rate (10.5%), etc. The various adjusted hyperparameters for the XGB model are.

Table 4, shows the three major hyperparameters (n_estimators, max_depth and eta) which are adjusted to obtain optimum model performance. A lower eta of 0.1 makes the algorithm more robust to variations and noise in the training data and allows the model to make smaller adjustments to the model weights during each boosting iteration.

Figure 15 shows a gradual decrease in error and increase in accuracy for an increase in number of boosting rounds. The test and train curves are almost similar indicating an absence of overfit. Also, the maximum depth hyperparameter has contributed a significant improvement to the model performance. The variation of *R*-squared with increasing maximum depth is shown in Figure 16. To find the optimum depth of trees, the curve between accuracy and maximum depth was studied. The maximum depth of 6 and the number of boosting rounds as 100 yields us the best performance model for the given data set. The testing accuracy of the model was 97.71, with a root-mean-square error of 0.001114. Figure 17 shows the actual vs predicted current graph for XGB. The closeness of data points to the fitted line serves as an indicator of the XGBoost model's good accuracy in making predictions.

The XGBoost model was validated using a 10-fold cross-validation technique, the cross-validation scores are 0.97709, 0.976916, 0.977619, 0.976612, 0.977568, 0.976897, 0.977224

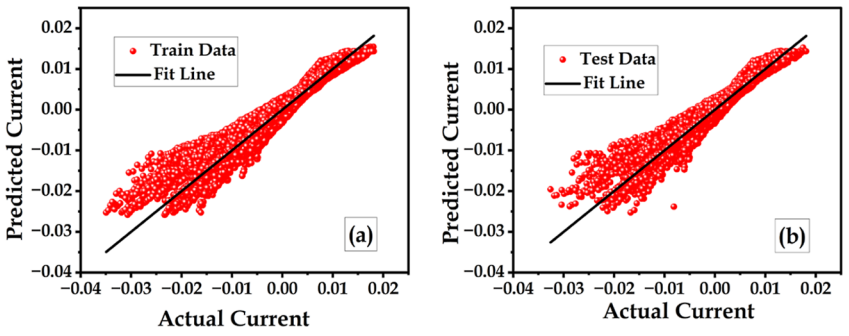


Figure 13. Comparison of actual vs predicted current values (a) train data and (b) test data for the RF model.

https://doi.org/10.1021/acsomega.3c10485
ACS Omega 2024, 9, 33459−33470

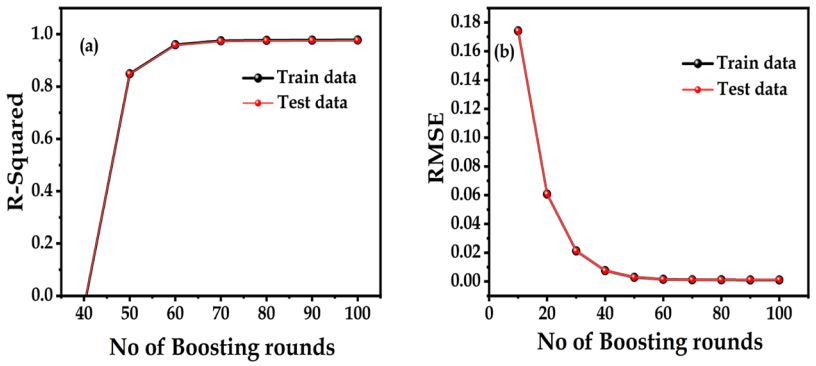


Figure 15. (a) *R*-Squared values and (b) RMSE values for testing and training CV data sets for different number of boosting rounds.

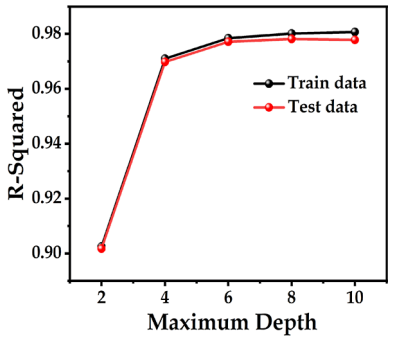


Figure 16. Variation in *R*-squared values for testing and training of CV profile for different maximum depths of XGB trees.

0.978407, 0.978286, and 0.977863, which is approximately equal to the obtained test and train r-scores.

4.1.4. Stacked Model. Figure 18 shows actual vs predicted current values for the meta-model. The meta-model's test accuracy was 97.73% and train accuracy was 97.95%. This similarity suggests that the model is performing consistently well on both the training and validation data sets. Here, the final estimator (RidgeCV) internally performs cross-validation to select the optimal value of alpha and helps in finding the right balance between bias and variance, ensuring that the meta-model does not overfit.

4.2. Specific Capacitance Comparison. The area under the CV curves was found and used for specific capacitance calculations, the actual capacitance with a CV curve area (0.00727468) for BFZO3 (m = 3.5 g) at 60 mV s^{-1} scan rate and potential window of 1 V was 0.01732 F g^{-1}. The XGBoost model was taken to predict the current for BFZO3 at 60 mV s^{-1}, and the CV curve was plotted. From the predicted CV curve, area under the curve was calculated (0.00755503) to get the predicted specific capacitance. From eq 1, the predicted capacitance was found to be 0.01798 F g^{-1}. The predicted specific capacitance was approximately similar to the experimental value, indicating the good performance of these machine-learning models for out-of-sample data sets (Figure 19).

4.3. Evaluated Metric Values for the ML Models. From Table 5, the comparison graphs were plotted to compare the *R*-squared and RMSE values for the above three predictive models. Also train and test errors and model accuracies for each individual model are approximately the same, which indicates good fit with absence of overfitting.

The plot (Figure 20) compares the performance of different models using validation metrics such as *R*-squared and RMSE. In Figure 21, the test accuracy and RMSE values are compared between the average of three models (ANN, RF, and XGB) and the meta-model. These models exhibit a high level of prediction accuracy for the given data sets, benefiting from their ability to correct errors made in previous steps. This results in robust and accurate predictive models.

5. CONCLUSIONS

A comprehensive study on the application of ML models to predict current from CV curves for substitution of zinc BFO composite at different scan rates were presented. The generalizability and interpretability performances of three different ML models are studied under different parameters. All three ML models performed well in predicting current from other CV curve parameters with an average accuracy of 97.65%. A stacked meta-model was developed, demonstrating a slight improvement over the aggregate performance of three individual ML models. The specific capacitance was

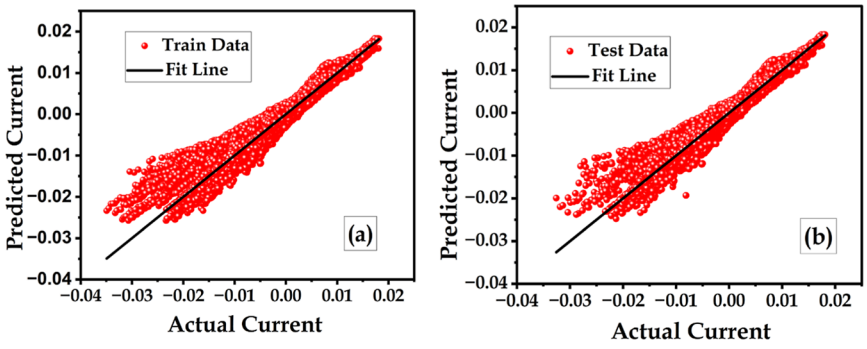


Figure 17. Comparison of actual vs predicted current values (a) train data and (b) test data for the XGB model.

https://doi.org/10.1021/acsomega.3c10485
ACS Omega 2024, 9, 33459−33470

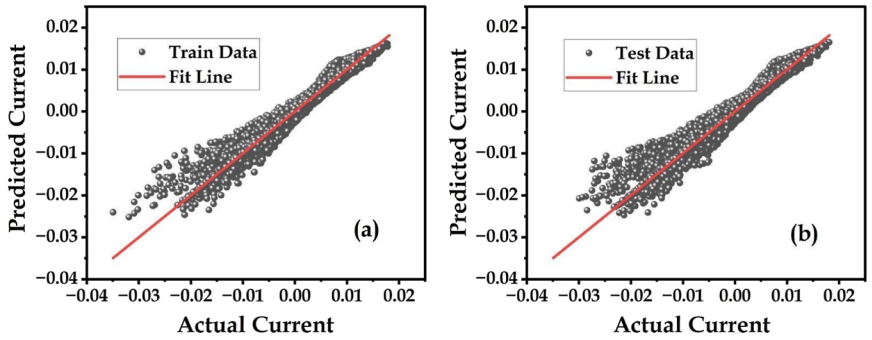


Figure 18. Comparison of actual vs predicted current values (a) train data and (b) test data for meta-model.

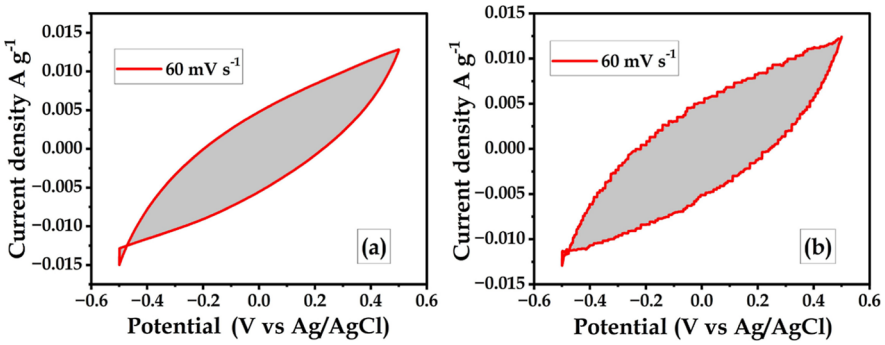


Figure 19. Comparison of CV plots for data set 2 (a) actual CV plot and (b) predicted CV plot by the XGB model for BFZO3 taken at 60 mV s^{-1} scan rate.

Table 5. Comparison Table Representing RMSE and *R*-Squared Values for Different ML Models

MODEL	TEST/TRAIN	RMSE	R2
ANN	TEST	0.001111	0.97725
ANN	TRAIN	0.001098	0.97768
RF	TEST	0.001155	0.97541
RF	TRAIN	0.001073	0.97871
XGB	TEST	0.001114	0.97714
XGB	TRAIN	0.001079	0.97847
meta/stacked	TEST	0.001108	0.97737
meta/stacked	TRAIN	0.001052	0.97953

determined for the out-of-sample data set (Data set 2), yielding a predicted value of 0.01798 F g^{-1}. This closely aligns with the experimental specific capacitance of 0.01732 F g^{-1}, providing compelling evidence for the model's accuracy and reliability. By utilizing these ML models to predict specific capacitance from CV data, researchers can rapidly screen a wide range of electrode materials without the need for extensive experimental testing which can accelerate the material discovery process and save both time and resources associated with the experiment. Once these ML models are trained, they can be deployed to make real-time predictions on new CV data. They can predict current and specific capacitance at different concentration doping and scan rates for a material with different dopants. They can also predict current for some more CV cycles at a fixed scan rate for a material, provided we must consider the temporal dynamics in the input data before being fed to the model. Combining the predictions from three algorithms leads to more accurate and generalizable predictions compared to using a single algorithm which assists in the design and fabrication of SC electrodes with specific capacitance targets in mind. Further integrating

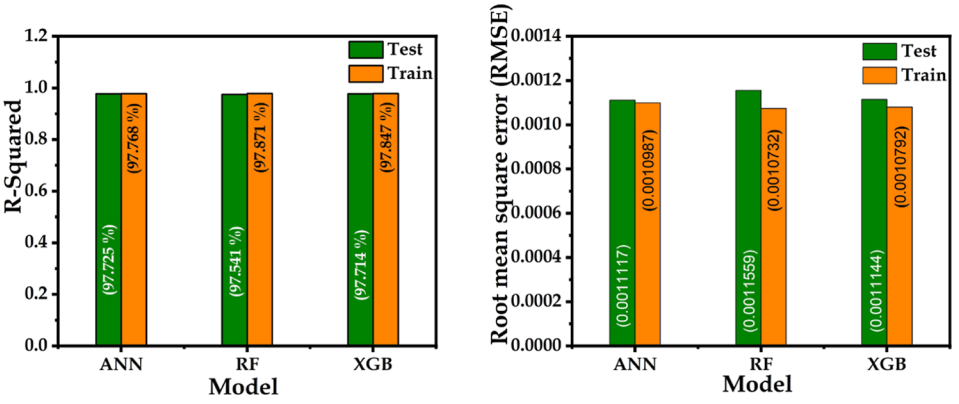


Figure 20. Comparison plots for test and train data against coefficient of determination/*R*-squared values for three different ML models.

https://doi.org/10.1021/acsomega.3c10485
ACS Omega 2024, 9, 33459−33470

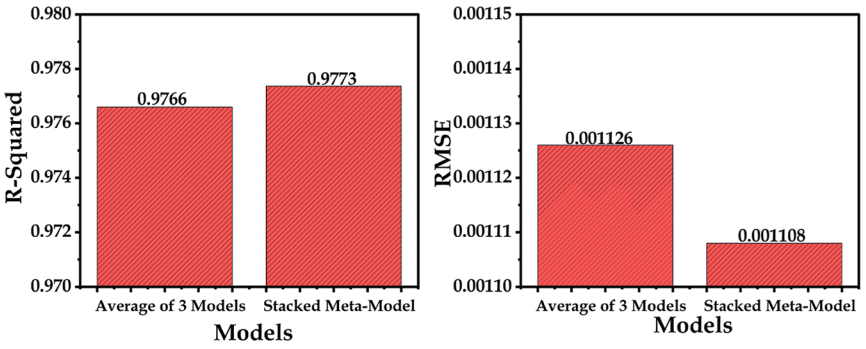


Figure 21. Comparison plots for *R*-squared and RMSE values for test data.

these ML models into the experimental setup or data acquisition system can enable seamless prediction of CV data in real time. We can also implement a feedback loop in the future to continuously update and refine these ML models based on new data and feedback from real-time predictions. Regular model evaluation and retraining ensure that these models remain accurate and reliable as the underlying system evolves. By harnessing the power of these models, we can advance the understanding and optimization of SC electrodes, ultimately enabling the advancement of more effective and eco-friendly energy storage solutions.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acsomega.3c10485.

Data set 1 (CV data for BFO, BFZO1, BFZO2, BFZO3, BFCO1, BFCO2, and BFCO3 are taken at different scan rates (10, 20, 30, 40, 50 mV s^{-1}), size of the data set 1 was 216,200 rows and 7 columns, data set 2 has CV data for BFZO3 taken at 60 mV s^{-1}, and size of the data set is 2000 rows and 7 columns (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Prabhavathy Mohanraj − *Department of Artificial Intelligence and Data Science, Coimbatore Institute of Technology, Coimbatore, Tamilnadu 641014, India;* orcid.org/0000-0003-2575-9024; Email: prabhavathy.phd@gmail.com

Authors

Abhilash Ravichandran − *Department of Chemical Engineering, National Institute of Technology Andhra Pradesh, West Godavari, Andhra Pradesh 534101, India*

Valliappan Raman − *Department of Artificial Intelligence and Data Science, Coimbatore Institute of Technology, Coimbatore, Tamilnadu 641014, India; Faculty of Engineering Computing and Science, Swinburne University of Technology Sarawak, 93350 Kuching, Sarawak, Malaysia*

Yogapriya Selvaraj − *Materials Research and Product Laboratory, Department of Chemistry, Coimbatore Institute of Technology, Coimbatore, Tamilnadu 641014, India; Department of Chemistry, Vinayaka Mission's Kirupananda Variyar Arts & Science, Vinayaka Mission's Research Foundation Deemed to be University, Salem, Tamil Nadu 636308, India*

Hemalatha Kuzhandaivel − *Materials Research and Product Laboratory, Department of Chemistry, Coimbatore Institute of Technology, Coimbatore, Tamilnadu 641014, India;* orcid.org/0000-0003-4082-1156

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.3c10485

Author Contributions

The authors Abhilash Ravichandran, Valliappan Raman, and Hemalatha Kuzhandaivel, planned the work. The author Abhilash Ravichandran created and tested models along with writing part. The author Yogapriya Selvaraj helped with writing works and formatting and exporting graphs. The author M Prabhavathy helped with model verification and paper correction.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The author Abhilash Ravichandran[a] would like to thank management and principal for giving the opportunity to work at Coimbatore Institute of Technology (CIT). The author also thanks the Department of AI and Data Science and Materials Research and Product laboratory, Department of Chemistry, Coimbatore Institute of Technology (CIT) for data collection and execution of work.

■ REFERENCES

(1) Al-Maswari, B. M.; Al-Zaqri, N.; Alkanad, K.; AlOstoot, F. H.; Boshaala, A.; Radhika, R. T.; Venkatesha, B. M. Magnesium Bismuth Ferrite Nitrogen-Doped Carbon Nanomagnetic Perovskite: Synthesis and Characterization as a High-Performance Electrode in a Supercapacitor for Energy Storage. *ACS Omega* **2023**, *8* (18), 16145−16157.

(2) Selvaraj, Y.; Kuzhandaivel, H.; Nallathambi, K. S.; Elayappan, V. Enhanced Cyclic Stability of Cobalt Doped $Bi_{25}FeO_{40}/BiFeO_3$ as an Electrode Material for a Super Long Life Symmetric Supercapacitor Device. *Energy Fuels* **2023**, *37* (12), 8624−8636.

(3) Adhikari, A. D.; Oraon, R.; Tiwari, S. K.; Saren, P.; Lee, J. H.; Kim, N. H.; Nayak, G. C. CdS-$CoFe_2O_4$@Reduced Graphene Oxide Nanohybrid: An Excellent Electrode Material for Supercapacitor Applications. *Ind. Eng. Chem. Res.* **2018**, *57* (5), 1350−1360.

(4) Shinde, P. V.; Shinde, N. M.; Mane, R. S.; Kim, K. H. Ferrites for Electrochemical Supercapacitors. In *Spinel Ferrite Nanostructures for Energy Storage Devices*; Mane, R. S., Jadhav, V. V., Eds.; Micro and Nano Technologies; Elsevier, 2020; Chapter 5, pp 83−122.

(5) Parwaiz, S.; Malik, O. A.; Pradhan, D.; Khan, M. M. Machine-Learning-Based Cyclic Voltammetry Behavior Model for Supercapacitance of Co-Doped Ceria/RGO Nanocomposite. *J. Chem. Inf. Model.* **2018**, *58* (12), 2517−2527.

(6) Zhu, S.; Li, J.; Ma, L.; He, C.; Liu, E.; He, F.; Shi, C.; Zhao, N. Artificial Neural Network Enabled Capacitance Prediction for Carbon-Based Supercapacitors. *Mater. Lett.* **2018**, *233*, 294−297.
(7) Nanda, S.; Ghosh, S.; Thomas, T. Machine Learning Aided Cyclic Stability Prediction for Supercapacitors. *J. Power Sources* **2022**, *546*, 231975.
(8) Sawant, V.; Deshmukh, R.; Awati, C. Machine Learning Techniques for Prediction of Capacitance and Remaining Useful Life of Supercapacitors: A Comprehensive Review. *J. Energy Chem.* **2023**, *77*, 438−451.
(9) Zhou, Y.; Huang, Y.; Pang, J.; Wang, K. Remaining Useful Life Prediction for Supercapacitor Based on Long Short-Term Memory Neural Network. *J. Power Sources* **2019**, *440*, 227149.
(10) Saad, A. G.; Emad-Eldeen, A.; Tawfik, W. Z.; El-Deen, A. G. Data-Driven Machine Learning Approach for Predicting the Capacitance of Graphene - Based Supercapacitor Electrodes. *J. Energy Storage* **2022**, *55*, 105411.
(11) Chicco, D.; Warrens, M. J.; Jurman, G. The Coefficient of Determination R-Squared Is More Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. *PeerJ. Comput. Sci.* **2021**, *7*, No. e623.
(12) Ghosh, S.; Rao, G. R.; Thomas, T. Machine Learning-Based Prediction of Supercapacitor Performance for a Novel Electrode Material: Cerium Oxynitride. *Energy Storage Mater.* **2021**, *40*, 426−438.
(13) Livingstone, D. J. Artificial Neural Networks Overview of Artificial Neural Networks. *Methods Mol. Biol.* **2009**, *458*, 14−22.
(14) Ahmad, M. W.; Mourshed, M.; Rezgui, Y. Trees vs Neurons: Comparison between Random Forest and ANN for High-Resolution Prediction of Building Energy Consumption. *Energy Build.* **2017**, *147*, 77−89.
(15) Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; Layton, R.; Vanderplas, J.; Joly, A.; Holt, B.; Varoquaux, G. API Design for Machine Learning Software: Experiences from the Scikit-Learn Project. *arXiv* **2013**, arXiv:1309.0238.
(16) Agarap, A. F. Deep Learning Using Rectified Linear Units (ReLU). *arXiv* **2019**, arXiv:1309.0238.
(17) Bell, A. J.; Sejnowski, T. J. An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Comput.* **1995**, *7* (6), 1129−1159.
(18) Efron, B.; Tibshirani, R. J. *An Introduction to the Bootstrap*; CRC Press, 1994.
(19) Avanijaa, J.; Al, E. Prediction of House Price Using XGBoost Regression Algorithm. *Turk. J. Comput. Math. Educ.* **2021**, *12* (2), 2151−2155.
(20) Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; KDD '16; Association for Computing Machinery: New York, NY, USA, 2016, pp 785−794.
(21) Shetty, S. S.; Hoang, D. C.; Gupta, M.; Panda, S. K. Learning Desk Fan Usage Preferences for Personalised Thermal Comfort in Shared Offices Using Tree-Based Methods. *Build. Environ.* **2019**, *149*, 546−560.
(22) Li, H.; Cao, Y.; Li, S.; Zhao, J.; Sun, Y. XGBoost Model and Its Application to Personal Credit Evaluation. *IEEE Intell. Syst.* **2020**, *35* (3), 52−61.
(23) Demir-Kavuk, O.; Kamada, M.; Akutsu, T.; Knapp, E.-W. Prediction Using Step-Wise L1, L2 Regularization and Feature Selection for Small Data Sets with Large Number of Features. *BMC Bioinf.* **2011**, *12* (1), 412.
(24) Wang, C.; Deng, C.; Wang, S. Imbalance-XGBoost: Leveraging Weighted and Focal Losses for Binary Label-Imbalanced Classification with XGBoost. *Pattern Recogn. Lett.* **2020**, *136*, 190−197.
(25) Yang, Q.; Du, S. "Prediction of concrete cubic compressive strength using ann based size effect model,". *Comput. Mater. Continua* **2015**, *47* (3), 217−236.
(26) Anitescu, C.; Atroshchenko, E.; Alajlan, N.; Rabczuk, T. "Artificial neural network methods for the solution of second order boundary value problems,". *Comput. Mater. Continua* **2019**, *59* (1), 345−359.
(27) Samaniego, E.; Anitescu, C.; Goswami, S.; Nguyen-Thanh, V.; Guo, H.; Hamdia, K.; Zhuang, X.; Rabczuk, T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Comput. Methods Appl. Mech. Eng.* **2020**, *362*, 112790.
(28) Barton, M.; Lennox, B. Model stacking to improve prediction and variable importance robustness for soft sensor development. *Digit. Chem. Eng.* **2022**, *3*, 100034.
(29) Khooran, M.; Haghighi, M. R. G.; Malekzadeh, P. Remaining Useful Life Prediction by Stacking Multiple Windows Networks with a Ridge Regression. *Iran J. Sci. Technol. Trans Mech Eng.* **2023**, *47*, 583−594.

https://doi.org/10.1021/acsomega.3c10485
ACS Omega 2024, 9, 33459−33470