PAPER

Efficient hyperparameter-tuned machine learning approach for estimation of supercapacitor performance attributes

Syed Ishtiyaq Ahmed, Sreevatsan Radhakrishnan, Binoy B Nair and Rajagopalan Thiruvengadathan

SIERS Research Laboratory, Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India

E-mail: t.rajagopalan@cb.amrita.edu

Keywords: supercapacitor, machine learning, modelling, simulation, artificial neural network, random forest, carbon

Supplementary material for this article is available online

Abstract

Recent years have witnessed the rise of supercapacitor as effective energy storage device. Specifically, carbon-based electrodes have been experimentally well studied and used in the fabrication of supercapacitors due to their excellent electrochemical properties. Recent publications have reported the use of Machine Learning (ML) techniques to study the correlation between the structural features of electrodes and supercapacitor performance metrics. However, the poor R-squared values (i.e., large deviations from the ideal value of unity) and large RMSE values reported in these works reflect the lack of accurate models’ development. This work reports the development and utilization of highly tuned and efficient ML models using hyperparameter tuning, that give insights into correlation between the structural features of electrodes and supercapacitor performance metrics namely specific capacitance, power density and energy density. Artificial Neural Networks (ANN) and Random Forest (RF) models have been employed to predict the various in-operando performance metrics of carbon-based supercapacitors based on three input features such as mesopore surface area, micropore surface area and scan rate. Experimentally measured values of these parameters used for training and testing these two models have been extracted from a set of research papers reported in literature. The optimization techniques and various tuning methodologies adopted for identifying model hyperparameters are discussed in this paper. The $R^2$ values obtained for prediction of specific capacitance, power density and energy density using RF model are in the range from 0.8612 to 0.9353 respectively, while the RMSE values of the above parameters are 18.651, 0.2732 and 0.5764 for respective input parameters. Similarly, the $R^2$ values obtained for prediction of specific capacitance, power density and energy density using ANN model are in the range from 0.9211 to 0.9644 respectively, while the RMSE values of the above parameters are 18.132, 0.1601 and 0.5764 for respective input parameters. Thus, the highly tuned ANN and RF models depict higher R-squared and lower RMSE values in comparison to those previously reported in literature, thereby demonstrating the importance of hyperparameter tuning and optimization in building accurate and reliable computational models.

1. Introduction

The rising human population and massive industrialization has imposed the need to generate, store and utilize energy efficiently. In particular, scientists continue to pursue research and development (R & D) in the domain of energy storage, after having demonstrated the capability to generate electrical energy to the desired level by technological advancements. Among the different energy storage mechanisms available, supercapacitors occupy centre stage due to their high-power density, long shelf life, high efficiency and higher flexibility in operating temperature [1–3].

Electric Double Layer Capacitor (EDLC) is a type of supercapacitor that stores and generates energy via reversible adsorption/desorption of ions at electrode surfaces. EDLCs are gaining prominence due to their...
ability to overcome the energy-storage gap between batteries and conventional capacitors. Carbon-based electrodes are being extensively studied and used for the fabrication of EDLC-type supercapacitors due to the impressive physical and chemical properties offered by carbon and its analogues [4]. To enhance the performance of supercapacitors, R & D activities are being focussed towards improving the specific capacitance and energy density offered by carbon electrodes, while still retaining their high-power density. The various techniques that have been employed by researchers to enhance the specific capacitance include introducing various functional moieties, increasing the surface area or altering the pore network of electrodes [5, 6].

It is generally expected that increasing the amount of micropores in the electrodes (which offer higher surface areas than mesopores and macropores), should lead to an increase in the capacitance and thus overall performance attributes of supercapacitor. However, a few experimental studies also report that increasing the micropore surface area leads to a decrease in the capacitance and power density [7–10]. One of the problems could be due to the inaccessibility of the micropores. This demonstrates the challenges encountered while studying the effect of pore size on the overall performance of the EDLC. An alternative approach to improving the specific capacitance, entails optimizing the structural features of carbon electrodes to improve EDLC performance. This need for optimization necessitates the development of efficient methodologies that help in understanding correlation between supercapacitor performance and structural features of electrodes in addition to studying the measurement parameter(s) like scan rate. Conventional methods include the usage of equivalent circuit- and molecular-based approaches, to understand the intricate charging/discharging kinetics of supercapacitors and analysing ionic behaviour. However, these physics-based approaches miss the microscopic details that explain the various processes behind energy storage and are constrained to operate under equilibrium conditions.

In the past one decade, metaheuristic-based and machine learning-based methods have become very popular to understand the physical processes in all branches of science and engineering domains [11–21]. More recently, a few works report the usefulness of machine learning approach in understanding the charge storage characteristics of supercapacitors [22–24]. These three publications discuss the usefulness of powerful ML algorithms like Generalized Linear Regression (GLR), Random Forest (RF), Support Vector Machine (SVM) and Artificial Neural Network (ANN) in studying the correlations between EDLC performance and its various features. Among the aforementioned algorithms, ANN prediction accuracy was found to be the highest, outperforming its ML counterparts. An excellent advantage offered by ANNs is their ability to model highly nonlinear feature and input-output relationships without having a complete understanding of the complex physical processes involved, thereby making the study of carbon electrodes, a suitable problem to be solved by ANN [22–24]. More specifically, the performance of ANN in predicting the optimum structural and operational features to improve EDLC performance has been studied and reported. The coefficient of determination (R²) and Root Mean Squared Error (RMSE) values were employed to evaluate the performance of the models in predicting the specific capacitance, power density and energy density in these works [22–24]. However, the process of analysing the dataset and the methodology adopted to evaluate the performance of the ML models described in these publications are not comprehensive and therefore these observations motivated us to expand the scope of analysis and improve accuracy.

To evaluate the performance of a ML model with an unbiased approach, it is undeniably appropriate to bifurcate the dataset into training and testing sets. When the model is trained using the training set, it learns the complex relationships and fundamental characteristics of input features and output labels. Once the model is sufficiently trained, to evaluate the strength of the model in predicting output labels for unseen input features, the model performance on testing data is assessed. This splitting of the dataset into training and testing data and evaluating the performance of the model on unseen data, provides an unbiased insight into the strength of the model. This process of bifurcation of the dataset into training and testing data was found to be missing in the publication [22], wherein all 70 data vectors were used for training as well as evaluation, which is a weak measure of model performance. Furthermore, the scarcity of data available to train and test the model built for predicting EDLC performance has to be tackled with refining and tuning the model to make it efficient. One approach to building highly tuned and efficient ML models is by tuning the hyperparameters used to construct the model.

Model hyperparameters have a major impact on performance as their values determine how a model learns the complex relationships between input features and output labels. A major distinguishing factor of model hyperparameters from other model parameters is that hyperparameters cannot be learnt during the training phase but must be set manually by the data scientist. Efficient methods facilitating hyperparameter tuning should be employed to build accurate predictive models. However, the previous works reported in literature have not employed hyperparameter tuning, as also reflected in the nominal R² and RMSE values [22–24]. Additionally, the publications discussing the building of ML models for predicting EDLC performance, have not disclosed vital details about the steps and the processes, parameters used, and evaluation criteria adopted for constructing and analysing various ML methods for EDLC performance prediction. Both the absence of above-mentioned information and development of less-efficient ML-based models with lower accuracy in predicting...
the performance attributes of supercapacitor as a result of not exploiting hyperparameter tuning in previous publications motivated us to revisit and develop highly efficient ML-based models through a systematic approach based on hyperparameter tuning.

Briefly, the experimental data used here have been extracted from previous publications reported in literature [25–31]. This data is bifurcated into training and testing sets to evaluate the model performance on unseen data and obtain an unbiased measure of model efficiency. A major focus of this publication is on the importance of hyperparameter tuning and how the process is imperative to building highly accurate and efficient ML models, especially for study of devices with scarcity of datasets or for devices whose performance-measuring techniques are costly and time-consuming, leading to lesser datasets. The various hyperparameters of ANN and RF, their relevance and techniques adopted to tune their values are discussed in detail. After building optimized models using hyperparameter-tuning techniques, the performance evaluation of these models depicts a higher $R^2$ and lower RMSE value for test data, compared to those reported in previous works [22–24]. The model-building techniques, hyperparameter tuning methodologies, details of models used, evaluation techniques and their results are discussed in the following sections, thereby addressing the afore-mentioned lacuna in knowledge.

2. Computational methodology

Data selection and method of processing are critical steps in building effective ML model as the range and distribution of data have a major impact on the information that the model learns. For example, numerical values and their ranges have significant effects on the relationships that regression models learn between input and output features [23, 24]. Exploratory data analysis (EDA) is an effective step-by-step approach that helps investigating data and identifying important and relevant patterns and relationships. EDA is utilized to study the range and distribution of data by generating relevant summary statistics of various input features and output labels identified in datasets.

In our work, EDLC performance depends on many factors including structural properties of electrode material, type of electrolyte used and other operating conditions like voltage window and scan rate. Physics-informed reasoning and analysis facilitates in determining the features to be retained and utilised for predicting the output labels. Since the carbon materials utilised in fabricating EDLC electrodes are pure, the need for including features talking about chemical properties like doping elements and their ratios can be eliminated. The charge and energy storage mechanism are rendered at the surface of electrodes, which allows us to conclude that the surface area of the electrodes holds greater significance as opposed to their pore volumes in predicting EDLC performance. Therefore, the surface area of electrodes is an important feature and are further classified into mesopore (>2 nm and <50 nm) and micropore (<2 nm). One might expect that for higher capacitance, it requires to have larger surface area and thus more micropores. On the contrary, it is documented that more micropores result in poorer capacitance as well as power density. This can be explained that poorer power density is due to increased resistance of ion diffusion and poorer capacitance is attributed to inaccessibility of micropore due to sieving and/or transport effect. However, it is also established that 3D pore network leads to improved capacitance and power density by optimising the surface areas of mesopores and micropores. Therefore, the BET areas of both mesopores and micropores are included to be one of the input features of the model.

Since charging/discharging duration of EDLC is intertwined with the scan rate, as this parameter determines the area under CV curve, and thus is directly proportional to integral capacitance measured. In general, the capacitance achieved in cyclic voltammetry at lower scan rate is higher than the corresponding measurements at higher scan rates, and the equivalent capacitance is achieved with infinitesimally slow charging. For better comparison of different features of electrodes, scan rate is therefore an important input feature for predicting supercapacitor performance. The output labels that denote EDLC performance are specific capacitance, power density and energy density. The above output parameters are selected as these correspond to key performance attributes of a given supercapacitor. The specific capacitance explains the magnitude of charge stored, while power and energy densities denote the delivered power and energy with respect to their weights and therefore are key output parameters. The dataset corresponding to the input features and output labels are extracted from publications that report experimental measurements using 6 M KOH as the electrolyte and a three-electrode cell arrangement with a potential window of 1 V [22–31]. The two ML models employed to make predictions are: (i) RF and (ii) ANN. RF is an ensemble method which combines outputs of multiple predictors (called ‘Decision Trees’), employing an if-then logic sequence. RF possesses the advantage of displaying low variance in the predicted output by using numerous and diverse group of predictors [32]. ANN is a machine learning algorithm that replicates the arrangement of neurons in the human brain and mimics its thought process. ANN is composed of multiple layers, each of which is composed of multiple neurons [11].
In ML, a model is expected to learn certain parameters from the given data, thereby identifying relationships between input and output variables during training. These are called ‘model parameters’. Hyperparameters are parameters that are not learned during the training phase but are fixed by the data scientist prior to training the model. Hyperparameters determine how a model learns the model parameters and the complexity involved. Recent publications discuss the importance of hyperparameter tuning in the development of ML-based models and the debate continues [33, 34]. Some publications show that hyperparameter tuning is fundamental to building highly accurate models as they determine the methodology and rate at which a model learns relationships from data [35, 36]. Taking the lessons from published works, we clearly demonstrate the importance and application of hyperparameter tuning, i.e., optimizing the values of hyperparameters to build effective ML models by improving the R² and RMSE values in comparison to values reported in literature using the same data sets [22–24].

To showcase the generalized nature of the developed models, the R² and RMSE values are evaluated on test data i.e., unseen data, to obtain a less-biased insight into the performance of the tuned models. ANN calculates the weighted sum of input features before passing through the various hidden layers consisting of neurons (each of which is attached to an activation function, which activate a neuron based on input relevance) to calculate the output of the neural network. To optimize the output, ANN employs back-propagation to update the weights used. Back propagation techniques cannot be employed to update model hyperparameters and therefore necessitate manual intervention [33].

ANN depends on multiple tunable hyperparameters, some of which are: (i) the number of hidden layers used, (ii) the activation function, (iii) number of neurons in hidden layers, (iv) number of epochs and batch size and (v) Kernel Initializer. The number of neurons and hidden layers determine the depth and width of the model in addition to its complexity. The number of epochs determine the number of cycles for which the model learns the entire dataset in a single iteration, since it is uncommon that a neural network learns all trends in a single training cycle. The activation function used determines the output calculated for weighted input and hence has an immense effect on model performance. Batch size decides the number of training samples the model works upon before updating the model parameters in a single training iteration [35, 36]. The Kernel Initializer denotes the statistical distribution from which, the initial weights allotted to different neurons are selected. RF also depends on multiple hyperparameters such as the number of estimators (decision trees used), the number of leaf neurons for each tree, maximum depth of estimators etc. The hyperparameter fixing the number of decision trees employed in the random forest is found to be of prime importance in determining model performance since its value determines output variance in addition to time complexity of the algorithm [32, 35, 36].

The route to building a highly accurate and efficient model is selecting the best set of values for different hyperparameters. The methodology used for finding the best possible set of hyperparameters could be on a trial-and-error basis i.e., randomly implementing different values for hyperparameters and evaluating the model performance. However, randomly choosing hyperparameter values and evaluating model performance is not an efficient process. Specifically, choosing hyperparameter values randomly and evaluating model performance is a faulty process for comparing the performance of multiple algorithms as different models might perform variably for different values of hyperparameters. Due to these reasons mentioned above, an automatic and accurate approach to find the best hyperparameter values is a better approach to hyperparameter tuning. One such approach available to researchers is Grid Search in combination with Cross Validation [37–39]. This process involves evaluating sets of parameter values and selecting the best model out of a family of models based on their test-set performance, which is calculated by cross-validation evaluation technique.

The cross-validation technique adopted for evaluating different possible sets of hyperparameters located on the grid is the ‘K-fold cross validation’. K-fold cross validation is a popular evaluation method for analysing model performance on unseen data. In this method, the dataset is split into k-sets where the first set is treated as the test set and the remaining k-1 sets are used as training data. The process is repeated for all k sets of data. This method gives an accurate measure of the model performance on unseen data as it depicts a less biased measure of model performance. The value of ‘k’ denoting the number of sets the data is split into for training and testing is fixed as 5, in the scope of this discussion. Grid Search is used to evaluate multiple models with different combinations of hyperparameter values. The different hyperparameter combinations represent different models and each model lies on a particular point in the grid created. The models are then evaluated for their prediction accuracy employing the K-fold cross validation technique discussed above and the model with the best performance is returned [37–39]. The performance evaluation metrics used to benchmark the models are R² and RMSE. Although computationally expensive, it is highly successful in finding the best set of hyperparameter values for a given predictive problem. To assist readers in visualization and implementation of the discussed techniques, the steps and processes involved in analysing data and building effective models for EDLC prediction is depicted in the flowchart shown in figure 1. The results and analysis of hyperparameter tuning for ANN and RF are discussed in subsequent sections.
3. Results and discussion

The input features identified for the predictive model are mesopore surface area, micropore surface area, and scan rate, based on the physics-based reasoning and analysis procedure discussed in sections 1 and 2. The output labels denoting the efficiency and performance of the EDLC device are its specific capacitance, power density and energy density. The dataset employed for the training the models is a 70×6 datasets extracted from several works reported in literature as mentioned earlier [25–31]. The summary statistics of the dataset used here is given in table 1.

As previously discussed, Grid Search in combination with Cross-Validation proves to be effective in testing performance of multiple combinations of model hyperparameter values and concluding the best performing set. The tunable hyperparameters identified to have a significant effect on ANN performance are: (i) the number of

---

**Table 1.** Summary Statistics of the Dataset used in this Work.

| Summary statistics | Specific capacitance | Energy density | Power density | Micro surface area | Meso surface area | Scan rate |
|--------------------|----------------------|----------------|---------------|--------------------|-------------------|-----------|
| Minimum            | 0.0000               | 0.0000         | 0.0000        | 0.0000             | 0.0000            | 0.0000    |
| Maximum            | 280.1000             | 9.7260         | 11.7040       | 1990.0000          | 1200.0000         | 500.0000  |
| Median             | 185.1300             | 6.4285         | 0.2175        | 314.5000           | 279.5000          | 10.0000   |
| Mean               | 173.1450             | 5.9612         | 0.8052        | 436.9143           | 452.7429          | 47.6143   |
| Skew               | −0.9163              | −0.9323        | 5.0344        | 1.4488             | 0.7233            | 3.8401    |

---
hidden layers, (ii) number of neurons in each hidden layer, (iii) optimization algorithm, (iv) the activation function employed, (v) number of epochs and batch size, and (vi) the Kernel Initializer.

The number of hidden layers employed by the model is fixed to 2 to reduce model complexity. A node (or neuron) is connected to multiple weighted input values, which are passed as inputs to the activation function employed by the neurons, and an output connection line which forwards the output to subsequent layers. Consequently, the number of neurons in each hidden layer and the number of hidden layers employed by the model play a fundamental role in determining model performance and constitute model architecture [11]. The number of neurons in each hidden layer is optimized by evaluating model performance with values ranging from 10 to 150 in steps of 10. One of the core functionalities of ANNs is their weight updating process, since it helps the neural network learn vital relationships between input features and output labels. When data is forward passed through various layers of the network, the weights allotted to different neurons are updated based on the difference between actual and predicted values. Every neural network is initialized with certain weights and biases after which they iteratively update these values and optimize the accuracy.

The optimization algorithm employed has a significant effect on model performance as the algorithm is responsible for updating the weights used for input features, based on the error calculated between predicted and actual values. Several optimization algorithms including SGD optimizer [40], RMSprop [41], Adagrad [42], Adadelta [43], and Adam [44] have been employed to check suitable algorithm based on best $R^2$ (close to unity) and RMSE values (close to zero). The activation function constituted in different neurons establish the non-linearity of the neurons and calculated the result generated at the output connection of these neurons, which are passed onto subsequent layers. Activation functions considered in this study include softmax, softplus, softsign, relu (Rectified Linear Unit), tanh, sigmoid, and linear. While the activation function on the input front is optimized for the hidden layers, the activation function employed in the output layer is ‘linear’ function.

The values used for optimizing batch size ranged from 5 to 25 in steps of 5, while the values tested for number of epochs ranged from 100 to 2500 in steps of 200. The ‘Kernel Initializer’ denotes the function or statistical distribution from which the initial weights to the neurons are allotted, i.e., they help initialize the weights. For e.g., if the kernel initializer is set to a normal distribution, the initial weights associated with different neurons are chosen from a normal distribution range and are employed as starting weights. Therefore, kernel initializer is one of the prominent and effective hyperparameter whose value needs to be tuned to achieve optimum accuracy. The different types of initializers offered by TensorFlow library include Uniform, Normal, HeNormal, HeUniform, GlorotNormal, GlorotUniform and so on. To find optimum hyperparameter values, a grid of possible values is constructed [35]. The regularization function adopted for backpropagation is Bayesian regularization function [45]. As mentioned earlier, it is found that the hyperparameter of significance that played a fundamental role in determining the performance of the RF Model was number of decision trees employed in training the model. Therefore, determining the optimum number of decision trees to be employed in a RF model to maximize the performance is of fundamental importance [35, 36, 45–47]. Grid Search in combination with cross validation is used to optimize this feature. Values ranging between 100 and 500 in steps of 50 are analysed for selecting the optimum number of decision trees.

The dataset employed for evaluating the performance of ANN and RF models is bifurcated into training and testing data in the ratio of 80:20 i.e., 56 vectors are employed for training and 14 vectors are employed for testing. The training dataset is employed for extracting the best-performing hyperparameter sets using Grid Search followed by cross validation for ANN and RF models. Grid search is performed by storing the hyperparameter ranges in dictionaries which are fed as input to the grid, following which the different grid points are tested for prediction accuracies for the three output labels by employing the K-fold cross validation method with a K-value of 5. Therefore, by employing the training dataset, we obtain a highly tuned model with the best performing set of hyperparameters for ANN and RF. The highly tuned models are then evaluated over unseen data, i.e., the test dataset comprising of 14 vectors. $R^2$ and RMSE values are then computed for the prediction made by the tuned models for the test data. This gives us a good insight into how the highly tuned model fares over unseen data. However, it can be argued that a single training-testing split can give a biased picture of the tuned model performance. In order to obtain an unbiased evaluation of the tuned model performance, the model is trained and tested for 20 iterations with completely random train-test splits, which results in a generalized unbiased evaluation of model performance. The $R^2$ and RMSE values evaluated for predictions made from the test data at the end of 20 iterations are reported and discussed via graphs for the tuned ANN and RF models. The $R^2$ and RMSE values reported in previous publications for untuned model predictions over the entire dataset i.e., 70 vectors are compared with the $R^2$ and RMSE values reported by the tuned model for the test dataset i.e., 14 vectors, at the end of 20 iterations. This comparison ratifies the importance of hyperparameter tuning in building efficient models. The workflow adopted in optimizing the hyperparameter values and testing their performance is described in scheme 1. The workflow in scheme 1 can be implemented using MATLAB or any other programming language with libraries supporting machine learning applications. The Python code for implemented for aforementioned steps is available for reader reference in the Supporting Information.
The analysis and comparison of tuned and un-tuned model performance is discussed below. Zhou et al. reported a $R^2$ value of 0.7167 and a RMSE value of 36.4013 for the prediction of specific capacitance using the developed ANN model for all 70 vectors without bifurcation into exclusive training and testing sets [22]. In contrast, in our work, when the tuned ANN model was evaluated over the test dataset, it displayed an impressive $R^2$ value of 0.9561 and a RMSE value of 18.1322 at the end of 20 iterations for test data containing specific capacitance values as shown in figure 2(a).

The power density values predicted by the tuned ANN model when compared with the test power density values, displayed an impressive $R^2$ score of 0.9644 and a RMSE of 0.1601 at the end of 20 iterations for test data containing power density values. The literature on the other hand reports a value of $R^2$ of 0.6382 and RMSE of 0.9574, attributing the reduction in RMSE values to the larger absolute values of specific capacitance reported by EDLCs. When the tuned ANN model was evaluated for its energy prediction densities it displayed an $R^2$ value of 0.9211 and a RMSE of 0.5764 at the end of 20 iterations for test data containing energy density values. Plots of predicted values of specific capacitance, power and energy densities versus the respective experimental values are shown in figures 2(a)–(c).

Upon inspection, the various hyperparameter values set for building the architecture of the optimal ANN model is given in table 3. Zhou et al reports a $R^2$ value of 0.6891 and RMSE of 38.1331 for specific capacitance prediction using the developed RF model for all 70 vectors without bifurcating into exclusive training and testing sets [22]. The optimized RF model in this work shows an impressive improvement in its performance when evaluated for the test data. The tuned RF model displays an impressive $R^2$ value of 0.9353 and a RMSE of 18.6514 at the end of 20 iterations for the specific capacitance predictions. The performance of the tuned RF model in predicting the specific capacitance values for the test input data at the end of 20 iterations is shown in figure 3(a).

The power density values predicted by the tuned RF model when compared with the actual test data (experimental) displays an $R^2$ value of 0.8612 and a RMSE of 0.2732 at the end of 20 iterations (figure 3(b)). Similarly, when evaluated for the energy density predictions, the tuned RF model displays an $R^2$ of 0.9224 and a RMSE of 0.6471 at the end of 20 iterations (figure 3(c)). Thus, it is seen that the predicted values for specific capacitance, power- and energy-densities have minimal deviation from the actual (experimentally determined) values of respective parameters, thereby emphasising the superior performance of the tuned model over unseen data compared to the poorer performance of the un-tuned model for the entire dataset. Upon inspection, the ideal value of the number of decision trees to be employed for building the efficient RF model is 100. A random forest model employing 100 decision trees depicts the results described above at the end of 20 iterations.
In the computational approach discussed in section 2, we have stated that the model is trained and tested for 20 iterations with completely random train-test splits to obtain an unbiased evaluation of the tuned model performance. As a typical example, the $R^2$ and RMSE values reported by the tuned ANN model for specific capacitance predictions over 20 random training - test splits are shown in figures 4(a) and (b).

**Figure 2.** Tuned ANN correlation plots between experimental and predicted values of (a) Specific Capacitance (b) Power Density and (c) Energy Density.
Figure 4(a) depicts the $R^2$ values reported for 20 random training/testing splits, with 70% of the iterations reporting an $R^2$ value greater than 0.875. The tuned ANN model shows an average $R^2$ value of 0.896 for specific capacitance predictions over 20 iterations. The standard deviation of the $R^2$ values is found to be 0.0861. This leads us to conclude that the tuned ANN model generalizes well over random train/test splits.

Figure 4(b) depicts the RMSE values reported for 20 random training/testing splits, with 65% of the iterations reporting an RMSE value lesser than or equal to 20. The model yields an average RMSE value of 18.566 for specific capacitance predictions over 20 iterations. The standard deviation of the RMSE values is found to be 8.259. This further supports our inference that the tuned ANN model generalizes well over random train/test splits. The performance over 20 iterations for the tuned ANN model for energy density and power density values is reported in the supporting information document. The performance over 20 iterations for the tuned RF model for specific capacitance, power density and energy density are also reported in the supporting information document for reader reference. These results discussed above buttress the superiority of highly tuned ML models namely ANN and RF in making predictions for specific capacitance, power and energy density with significantly greater accuracy for the same electrode configuration reported in previous publications.

Although comparisons between the performance of the highly tuned models employed in our work and the models reported in literature [22–24] have been described, the benchmark for the performance of different ML models employed by the authors has been limited to the $R^2$ and RMSE values computed in this work, as the dataset employed for training and testing is identical to the dataset employed by Zhou et al [22] The other two publications [23, 24] were primarily referenced to study and understand the various input features and output labels employed in building efficient models for predicting supercapacitor performance and the methods adopted to deduce relevant parameters. Besides, the datasets employed in publications [22, 23] were larger compared to the one employed in [21], with 170 and 178 vectors respectively. Since the major focus of this publication is to demonstrate the importance of hyperparameter tuning in building highly efficient models that make accurate predictions despite the limitations of having smaller datasets, the discussion on comparison of our work with the performance of models in publications [22, 23] was limited. However, the $R^2$ and RMSE values reported by the highly-tuned models with smaller datasets were more impressive compared to the $R^2$ and RMSE values reported by models which were trained over larger datasets employed in publications [22, 23], thereby compelling the authors to establish the superiority of the tuned models described and performed in our work.

| Algorithm Used | Parameter  | Literature value | Our work | Literature value | Our work |
|----------------|------------|------------------|----------|------------------|----------|
| ANN            | Specific Capacitance | 0.7167 | 0.9561 | 36.4013 | 18.1322 |
| ANN            | Power Density     | 0.6382 | 0.9644 | 0.9574 | 0.1601 |
| ANN            | Energy Density    | Not Reported | 0.9211 | Not Reported | 0.5764 |
| RF             | Specific Capacitance | 0.6891 | 0.9353 | 38.1331 | 18.6514 |
| RF             | Power Density     | Not Reported | 0.8612 | Not Reported | 0.2732 |
| RF             | Energy Density    | Not Reported | 0.9224 | Not Reported | 0.6471 |

* Literature values reported here correspond to conditions without bifurcating the experimental dataset into training and testing vectors and without the hyperparameter optimization.

Figure 4(a) depicts the $R^2$ values reported for 20 random training/testing splits, with 70% of the iterations reporting an $R^2$ value greater than 0.875. The tuned ANN model shows an average $R^2$ value of 0.896 for specific capacitance predictions over 20 iterations. The standard deviation of the $R^2$ values is found to be 0.0861. This leads us to conclude that the tuned ANN model generalizes well over random train/test splits.

Figure 4(b) depicts the RMSE values reported for 20 random training/testing splits, with 65% of the iterations reporting an RMSE value lesser than or equal to 20. The model yields an average RMSE value of 18.566 for specific capacitance predictions over 20 iterations. The standard deviation of the RMSE values is found to be 8.259. This further supports our inference that the tuned ANN model generalizes well over random train/test splits. The performance over 20 iterations for the tuned ANN model for energy density and power density values is reported in the supporting information document. The performance over 20 iterations for the tuned RF model for specific capacitance, power density and energy density are also reported in the supporting information document for reader reference. These results discussed above buttress the superiority of highly tuned ML models namely ANN and RF in making predictions for specific capacitance, power and energy density with significantly greater accuracy for the same electrode configuration reported in previous publications.

Although comparisons between the performance of the highly tuned models employed in our work and the models reported in literature [22–24] have been described, the benchmark for the performance of different ML models employed by the authors has been limited to the $R^2$ and RMSE values computed in this work, as the dataset employed for training and testing is identical to the dataset employed by Zhou et al [22] The other two publications [23, 24] were primarily referenced to study and understand the various input features and output labels employed in building efficient models for predicting supercapacitor performance and the methods adopted to deduce relevant parameters. Besides, the datasets employed in publications [22, 23] were larger compared to the one employed in [21], with 170 and 178 vectors respectively. Since the major focus of this publication is to demonstrate the importance of hyperparameter tuning in building highly efficient models that make accurate predictions despite the limitations of having smaller datasets, the discussion on comparison of our work with the performance of models in publications [22, 23] was limited. However, the $R^2$ and RMSE values reported by the highly-tuned models with smaller datasets were more impressive compared to the $R^2$ and RMSE values reported by models which were trained over larger datasets employed in publications [22, 23], thereby compelling the authors to establish the superiority of the tuned models described and performed in our work.

| Hyperparameter          | Optimal Hyperparameter value |
|-------------------------|------------------------------|
| Number Of Epochs        | 2000                         |
| Optimization function   | Adam                         |
| Number of neurons in each hidden layer | 50                           |
| Number of hidden layers employed | 2                            |
| Activation function     | Relu (Rectified Linear Unit) |
| Weight Initializer      | Uniform Distribution         |

Table 2. A comparison of $R^2$ and RMSE values between our work and reported in literature [22].

Table 3. Hyperparameter values used in highly-optimized ANN model.
Performance of a machine learning model is deeply intertwined with the nature of the dataset and the amount of data available for training the model. As a result, researchers test a range of ML algorithms/models and compare their relative performance to identify the optimal model for the application. Artificial Neural Networks (ANN) and Random Forests (RF) are two of the most popular machine learning algorithms in use today. ANNs can be considered to be universal function approximators [48, 49] while RFs are ensemble classifiers composed of multiple decision trees [50]. Both ANN and RF are not only capable of learning complex...
relationships between features of the dataset but are also robust, as demonstrated in literature [51, 52]. In this study, hyperparameter tuning is employed to further improve the performance of ANN and bring the R2 value as close to unity as possible. The algorithm developed in this work together with hyperparameter tuning can be directly employed without the need for training to predict more accurately the performance of activated carbon-based electrodes in supercapacitor geometry provided the input and the output features used in the model are available. For other porous materials used in electrode fabrication similar methodology can be adopted to train and test the model, provided datasets consisting of experimentally measured values of surface areas due to micropores and mesopores and scan rates with corresponding output features such as specific capacitance, energy density and power density are available. For non-porous materials used as electrodes in supercapacitor geometry, one may need to train the model with a different set of input features though output parameters remain the same. Yet, the bottom-line is that hyperparameter tuning of the model is essential to predict the performance more accurately.

4. Conclusion

In this work, ANN and RF models were employed to study the dependency of electrochemical performance of EDLC on structural properties of electrodes like mesopore surface area, micropore surface area and in-operando kinetic conditions like scan rate. The processes of Principle Component Analysis (PCA) and Exploratory Data Analysis (EDA) were employed in identifying the appropriate set of input features and their dependencies with EDLC performance. The importance of hyperparameter tuning and the mathematical influence they render in building model architecture and their over-arching role in determining model performance is analysed and experimented. The relevance and application of hyperparameter tuning methodologies like Grid Search Cross Validation in selecting the best set of hyperparameters to maximize model performance has been explored. The highly-tuned ANN and RF models depict improved performance in comparison to model performances reported in previous publications, thereby signifying the importance of hyperparameter tuning in building highly accurate prediction models, with no pre-requisites of learning the physics underlying various processes. Further research into building highly-accurate generalized ML models, predicting performance of other energy storage devices like batteries and fuel cells is identified as potential opportunity to improvise and diversify applications of Machine Learning in electrochemistry.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

ORCID iDs

Rajagopalan Thiruvengadathan https://orcid.org/0000-0001-9609-3245

Figure 4. (a) R2 and (b) RMSE values obtained as a function of number of iterations for 20 random training/testing splits.
References

[1] Frackowiak E 2007 Carbon materials for supercapacitor application Phys. Chem. Chem. Phys. 9 1774–85
[2] Futata D N et al 2006 Shape-engineerable and highly densely packed single-walled carbon nanotubes and their application as supercapacitor electrodes Nat. Mater. 5 987–94
[3] Snoook G A, Kao P and Best A S 2011 Conducting-polymer-based supercapacitor devices and electrodes J. Power Sources 196 1–12
[4] Yasoda K Y et al 2021 Fabrication of MnS/GO/PANI nanocomposites on a highly conducting graphite electrode for supercapacitor application. Today Chem. 19 100394
[5] An K H et al 2001 Electrochemical properties of high-power supercapacitors using single-walled carbon nanotube electrodes Adv. Funct. Mater. 11 387–92
[6] Chen W et al 2011 High-performance nanostructured supercapacitors on a sponge Nano Lett. 11 5165–72
[7] Eldad L et al 2001 Ion sieving effects in the electrical double layer of porous carbon electrodes: estimating effective ion size in electrolytic solutions J. Phys. Chem. B 105 6880–7
[8] Ghosh A and Lee Y H 2012 Carbon-based electrochemical capacitors ChemSusChem 5 480–99
[9] Hasenjäger M et al 2012 New insights into the relationship between capacitance in monolithic carbon electrodes J. Phys. Chem. C 116 26397–203
[10] Lee G J and Pyun S I 2006 Theoretical approach to ion penetration into pores with pore fractal characteristics during double-layer charging/discharging on a porous carbon electrode Langmuir 22 10659–65
[11] Abraham A 2005 Artificial neural networks Handbook of Measuring System Design ed P H Sydenham and R Thorn (New York: Wiley) (https://doi.org/10.1002/0471497398.mm421)
[12] Butler K T et al 2018 Machine learning for molecular and materials science Nature 559 547–53
[13] Gulh K and Pal S 2017 Deep Learning with Keras (United States of America: Packt Publishing) (https://www.packtpub.com/product/deep-learning-with-keras/9781787128422)
[14] Han S J et al 2020 Evidence of scatter in C-band spatio-temporal signals using machine learning models 96th IEEE Int. Conf. on Communication and Signal Processing, ICCSP 2020 (Institute of Electrical and Electronics Engineers Inc) (https://doi.org/10.1109/ICCSSP48568.2020.9182119)
[15] Liu Y et al 2017 Materials discovery and design using machine learning J. Materiomics 3 159–77
[16] Prabhu S S et al 2017 A computational approach to determine shielding effectiveness of carbon nanotube-based nanocomposites for EMC application Comput. Mater. Sci. 126 400–6
[17] Ripley B D 2014 Pattern recognition and neural networks Pattern Recognition and Neural Networks (Cambridge: Cambridge University Press) 1–403
[18] Shalev-Shwartz S and Ben-David S 2013 Understanding machine learning: from theory to algorithms. Understanding Machine Learning: From Theory to Algorithms (Cambridge: Cambridge University Press) 1–397
[19] Surya Natarajan S et al 2020 Detection of Interference in C-Band Signals using K-Means Clustering 9th IEEE Int. Conf. on Communication and Signal Processing, ICCSP 2020 (Institute of Electrical and Electronics Engineers Inc) (https://doi.org/10.1109/ICCSSP48568.2020.9182228)
[20] Tálbi E G 2009 Metaheuristics: from design to implementation Metaheuristics: From Design to Implementation (New York: Wiley)
[21] Witten I H et al 2016 Data mining: practical machine learning tools and techniques Data Mining: Practical Machine Learning Tools and Techniques 4th Edition (United States of America: Elsevier Inc) 1–621 (https://www.elsevier.com/books/data-mining/witten/978-0-12-380429-1)
[22] Zhou M et al 2020 Insights from machine learning of carbon electrodes for electric double layer capacitors Carbon 157 147–52
[23] Zhou M, Vassallo A and Wu J 2020 Data-driven approach to understanding the in-operando performance of heteroatom-doped carbon electrodes ACS Appl. Energ. Mat. 3 5993–6000
[24] Zhu S et al 2018 Artificial neural network enabled capacitance prediction for carbon-based supercapacitors Mater. Lett. 233 294–7
[25] Jiang L et al 2016 Construction of nitrogen-doped porous carbon building blocks using interconnected ultra-small carbon nanosheets for ultra-high rate supercapacitors J. Mater. Chem. A 4 11386–96
[26] Wu H et al 2012 The effect of activation technology on the electrochemical performance of calcium carbide skeleton carbon J. Solid State Electrochem. 16 2941–7
[27] Li Y T et al 2015 Hierarchical porous active carbon from fallen leaves by synergy of K2CO3 and their supercapacitor performance J. Power Sources 299 519–28
[28] Jiang L et al 2013 High rate performance activated carbons prepared from ginkgo shells for electrochemical supercapacitors Carbon 56 146–54
[29] Zhang J et al 2012 Preparation of activated carbon from waste Camellia oleifera shell for supercapacitor application J. Solid State Electrochem. 16 2179–86
[30] Zhang D et al 2017 Scalable synthesis of hierarchical macropore-rich activated carbon microspheres assembled by carbon nanoparticles for high rate performance supercapacitors J. Power Sources 342 363–70
[31] Yang W et al 2017 Template-free synthesis of ultrathin porous carbon shell with excellent conductivity for high-rate supercapacitors Carbon 111 419–27
[32] Liaw A and Wiener M 2002 Classification and Regression by randomForest R News. 2 18–22
[33] Probst P, Boulesteix A L and Bischl B 2019 Tunability: importance of hyperparameters of machine learning algorithms J. Mach. Learn. Res. 20 1934–65
[34] Van Rijn J N and Hutter F 2018 Hyperparameter importance across datasets 24th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD 2018 (Association for Computing Machinery) (https://doi.org/10.1145/3219819.3220558)
[35] Falkner S, Klein A and Hutter F 2018 BOHB: robust and efficient hyperparameter optimization at scale 35th Int. Conf. on Machine Learning, ICML 2018 (Int. Machine Learning Society (IMLS)) (http://proceedings.mlr.press/v80/falkner18a/falkner18a.pdf)
[36] Hertel L et al 2020 Sherpa: robust hyperparameter optimization for machine learning SoftwareX 12 100591
[37] Abas M A H et al 2020 Agarwood oil quality classification using support vector classifier and grid search cross validation hyperparameter tuning Int. J. Emerg. Trends Eng. Res 8 2551–6
[38] Krstajic D et al 2014 Cross-validation pitfalls when selecting and assessing regression and classification models J. Cheminformatics 6 10
[39] Yasin H et al 2016 Prediction of crude oil prices using support vector regression (SVR) with grid search—Cross validation algorithm Global J. Pure Appl. Math. 12 3009–20
[40] Bottou L 1998 Online algorithms and stochastic approximations Online Learning and Neural Networks ed D Saad (Newyork, United States of America: Cambridge University Press) pp 9–42
[41] Hinton G, Srivastava N and Swersky K 2012 Neural networks for machine learning lecture 6a overview of mini-batch gradient descent

Cited 14 249–50

[42] Duchi J, Hazan E and Singer Y 2011 Adaptive subgradient methods for online learning and stochastic optimization J. Mach. Learn. Res. 12 2121–59

[43] Zeiler M D 2012 ADADELTA: An Adaptive Learning Rate Method arXiv:1212.5701

[44] Kingma D P and Ba J L 2015 Adam: a method for stochastic optimization 3rd Int. Conf. on Learning Representations, ICLR 2015 (Int. Conf. on Learning Representations, ICLR)

[45] Wu J et al 2019 Practical multi-fidelity Bayesian optimization for hyperparameter tuning 35th Conf. on Uncertainty in Artificial Intelligence, UAI 2019 (Association For Uncertainty in Artificial Intelligence (AUAI)) (http://auai.org/uai2019/proceedings/papers/284.pdf)

[46] Bernard S, Heutte L and Adam S 2009 Influence of hyperparameters on random forest accuracy 8th International Workshop on Multiple Classifier Systems, MCS 2009 (Reykjavik) p 171–80

[47] Probst P and Boulesteix A L 2018 To tune or not to tune the number of trees in random forest J. Mach. Learn. Res. 18 1–8 (https://www.jmlr.org/papers/volume18/17-269/17-269.pdf)

[48] Leshno M et al 1993 Multilayer feedforward networks with a nonpolynomial activation function can approximate any function Neural Netw. 6 861–7

[49] Sonoda S and Murata N 2017 Neural network with unbounded activation functions is universal approximator Appl. Comput. Harmon. Anal. 43 233–68

[50] Breiman L 2001 Random forests Mach. Learn. 45 5–32

[51] Chiu C T et al 1993 Robustness of feedforward neural networks IEEE Int. Conf. on Neural Networks, ICNN 1993 (Institute of Electrical and Electronics Engineers Inc) (https://doi.org/10.1109/ICCNN.1992.226963)

[52] Roy M H and Larocque D 2012 Robustness of random forests for regression J Nonparametr Stat. 24 993–1006