Chapter-5

Machine learning-based approach for screening of supercapacitor electrode materials and its performance evaluation

5.1 Introduction

The utilization of data science and machine learning (ML) methodologies within the field of materials science has seen significant growth in recent times. By leveraging ML, we can efficiently evaluate materials by utilizing input features obtained from existing databases, thereby facilitating the rapid identification of materials that meet our specific requirements. In recent years, machine learning has garnered considerable attention in the domain of energy material design and optimization, particularly concerning its application to supercapacitors, leading to notable advancements. Numerous studies have explored the impact of various factors on specific capacitance, an essential parameter for evaluating the performance of supercapacitors. Zhu et al. [1] utilized machine learning models to predict the capacitance of carbon materials. The input features they incorporated comprised specific surface area, pore size, voltage window, doping elements, etc. by using Artificial Neuron Network (ANN). Su et al. [2] investigated the influence of porous carbon materials and potential windows on the capacitance of the electric double layer through the application of four machine learning models. Ghosh et al. [3] utilized a hybrid methodology that integrates value and Grade prediction machine learning models to forecast the performance of a new material in supercapacitor applications. A research investigation has shown that artificial neural network (ANN) models may serve as dependable predictive instruments for evaluating the performance of flexible supercapacitors utilizing alkali lignin, in comparison to other machine learning models [4]. Rahimi et al. [5] have forecasted the specific capacitance of Activated Carbon electrodes enriched with heteroatoms in EDLCs by analysing microstructural characteristics, the presence of Nitrogen and Oxygen functional groups, and various operational parameters. The authors employed a Multi-Layer Perceptron (MLP) neural network for their predictive analysis.

Hybrid Machine Learning algorithms have gained significant traction by integrating two distinct algorithms to enhance the performance of the model.

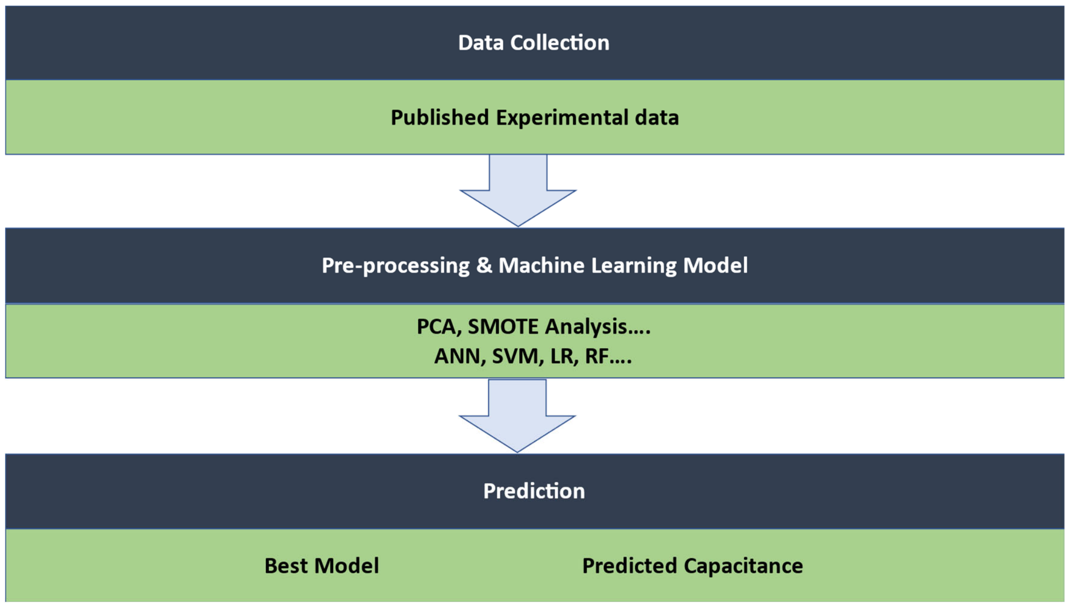

Figure 5.1- *Schematic Illustration of the sequential approach used for predicting supercapacitor capacitance by using machine learning models*

In this study, a data-driven methodology has been employed to forecast the performance of cerium-based electrode material [6]. Here, we have first detailed several important characteristics, such as the morphology and composition of the electrodes, the voltage window, the applied current density, and the electrolytes used, which can significantly affect the capacity retention of the supercapacitor. We subsequently gathered these characteristics from numerous published research articles and established a database. Following the pre-processing stage, which includes the elimination of features with missing values and the prioritization of attributes, the dataset was randomly divided into training and testing sets. We implemented machine learning algorithms on the training dataset, selecting the most suitable algorithms to forecast outcomes for the testing dataset. Our analysis identified the Random Committee and Random Forest algorithms as the most effective performers. The machine learning tasks are performed utilizing WEKA 3.8.6 software [7].

5.2 Dataset description

The dataset used in this work consists of multiple attributes that describe various properties about the material's structure, electrochemical performance, and experimental conditions of cerium-based composite and non-composite materials used in electrochemical applications and prepared from the published literature. Initially, a range of parameters is identified that are deemed essential for assessing the material's performance from a supercapacitive standpoint. These parameters are evaluated from two perspectives: (i) the material perspective, which includes material composition, morphology, and surface area, and (ii) the device perspective, encompassing substrate, current density, potential window, electrolyte, etc. We have used the following attributes in our study:

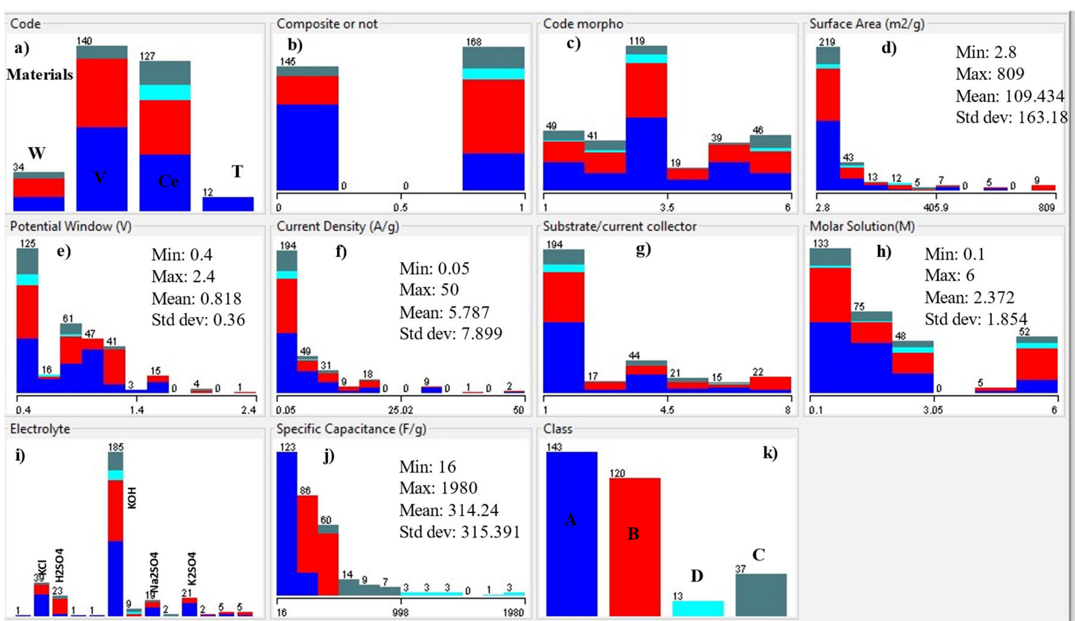

***Figure 5.2**- The features used in the dataset*

Code: Apart from Cerium, Vanadium, Titanium, and Tungsten-based oxide, nitride, oxynitride, and their composites are used to form the database and assign C, V, T, and W codes, respectively.

Composite or Not: Binary attribute indicating whether the material is a composite or not.

Code Morphology: Categorical attributes represent different morphological classifications of the material. 1 - 1D (e.g., quantum dot), 2 - 2D (e.g., rods, tube-like structures), 3 - 3D (3D-nanoparticles with no mainly defined morphology), 4- combinational structures (a structure with multiple morphologies especially in the composites), 5- highly porous structures, 6- 3D nanoparticles with unique morphology (flower-like, spherical-shaped, etc.)

Surface Area: Real-valued attribute measuring the specific surface area of the material in m^2/g.

Potential Window: Real-valued attribute representing the electrochemical potential window in volts.

Current Density: Real-valued attribute that measures the current density of the material in amperes per gram.

Substrate/current collector: Categorical attribute representing different types of substrates or current collectors used in the experiments. 1- Ni-based, 2- Stainless steel, 3- Ti-based, 4- Carbon based substrate, 5- Glassy Carbon, 6- Graphite, 7- Vanadium-based, 8- Unknown

Molar Solution: Real attribute representing the molarity of the solution used.

Electrolyte: Categorical attribute indicating the type of electrolyte used: $CaCl_2$, H_2SO_4, KOH, K_2SO_4, etc.

Specific Capacitance: Real-valued attribute measuring the specific capacitance of the material in farads per gram.

The specific capacitance classification experiment is conducted through a capacitance grade prediction approach. To do this, the data is classified into four distinct classes based on specific capacitance values:

Grade A - specific capacitance less than 200 F/g, **Grade B** – specific capacitance value from 200 to 500 F/g, **Grade C** – specific capacitance value in the range of 501–1000 F/g, and **Grade D** – for the specific capacitance value more than 1000 F/g.

5.3 Methodology

Our study incorporated multiple machine learning algorithms offered by WEKA software, as well as performance metrics that were utilized in the evaluation process. This section has provided an overview of the unique functionalities of some algorithms and the relevant performance metrics. The discussion in this section also includes the strategies for data balancing and feature selection employed during the pre-processing activities.

5.3.1 Machine Learning Algorithms

Machine learning algorithms are computational frameworks that enable computers to recognize patterns and predict outcomes or make decisions based on data, all without the need for explicit programming. Some of the ML algorithms employed in our work are discussed as follows:

5.3.1.1 Multilayer Perceptron (MLP)

A Multi-Layer Perceptron (MLP) is a form of artificial neural network characterized by its composition of several layers of neurons, or nodes, organized in a hierarchical manner. It is recognized as one of the most fundamental and commonly utilized types of neural networks, especially in supervised learning applications such as classification and regression.

The fundamental mechanism that governs the operation of a multilayer perceptron is backpropagation, which is a crucial algorithm employed for training the network. In the backpropagation process, the network modifies its weights and biases by sending the error backward from the output layer to the input layer. This iterative method refines the model's parameters, allowing it to enhance its predictive accuracy over time.

An MLP typically includes the following components:

- **Input layer**: Receives input data and passes it on to the hidden layers.
- **Hidden layers**: Consist of one or more layers of neurons that perform computations and transform the input data.
- **Activation function**: Applies a non-linear transformation to the output of each neuron in the hidden layers.
- **Output layer**: Produces the final output of the network, such as a classification label or a regression target.
- **Weights and biases**: Adjustable parameters that determine the strength of the connection between neurons in adjacent layers and the bias of each neuron.

5.3.1.2 Random Forest

Random Forest (RF) is an ensemble learning model that consists of multiple Classification and Regression Trees (CART), which are trained using techniques such as bagging and the random selection of variables [8,9]. CART employs recursive partitioning to construct decision trees, with the choice of splitting variables being heavily influenced by the characteristics of the learning sample. This process is inherently fragile, as even minor alterations in the learning sample can lead to different selections of splitting variables, thereby altering the entire tree structure. In contrast, RF effectively mitigates this instability by utilizing a collection of diverse trees. It generates predictions by aggregating the outputs from multiple trees. RF enhances tree diversity through the randomization of both the training dataset and the input variables.

Initially, Random Forest (RF) creates several new training datasets through random sampling of the original dataset with replacement, ensuring that these new datasets contain the same number of observations as the original. Prior to the tree splitting phase, RF employs variable set randomization to improve the diversity of the trees, generating a random set of variables for each new training dataset. The trees are then developed by identifying the optimal split within this random variable set, with the expectation that their growth will be independent. Upon completion of the growth process, RF produces predictions by averaging the predictions from all individual trees, thereby enhancing the accuracy of the RF prediction and significantly mitigating the risk of substantial errors.

5.3.1.3 Random Committee

A Random Committee generates several randomizable base classifiers by utilizing distinct random number seed values. The ultimate classification outcome is determined by averaging the predictions provided by each of the individual base classifiers [10]. This algorithm has been selected for our dataset due to its ability to minimize the repeatability of results by employing a distinct random number of seeds. In contrast, the base classifiers would have consistently produced identical predictions.

5.3.1.4 Multiclass Classifier

A Multiclass Classifier (MCC) is categorized as a type of meta classifier. Meta classification entails the integration of several classifiers. This integration process is divided into three distinct phases. Initially, various training subsets are created from the original training data. In the subsequent phase, each classifier operates independently, relying on its specific algorithm and the corresponding training subset. Finally, the outputs from the base classifiers are aggregated in the concluding phase, resulting in the final outcomes produced by the higher-level entity known as the meta classifier [11]. The Multiclass Classifier serves as an enhancement for any binary class classifier. This particular classifier is well-suited for our dataset, as it has the potential to enhance accuracy through the implementation of error correction codes. Given that our dataset exhibits non-uniform instance weights, the MCC proves to be an effective algorithm in this scenario, as it resamples and substitutes data according to these weights prior to its input into the base classifier, thereby improving the predictive accuracy of the classifiers.

5.3.1.5 Random Tree

The Random Tree (RT) algorithm is a form of ensemble learning that creates multiple individual learners. In this context, bagging is employed to generate a collection of random datasets utilized for the construction of decision trees. Each node within a conventional tree is divided based on the optimal split derived from all available variables. Each node in a random forest is divided based on the optimal predictor selected from a randomly chosen subset of predictors at that specific node [12].

Random Tree addresses both classification and regression tasks. In the case of classification, the Random Tree classifier processes the input feature vector by evaluating it against each tree in the forest. The predicted output class is determined based on the majority of votes received from all the trees. For regression tasks, the predicted output is calculated as the average of the predictions generated by each tree. This algorithm is appropriate for our dataset as it falls within the category of ensemble learning algorithms. It is well-established that ensemble methods can enhance predictive accuracy by integrating multiple learning algorithms, surpassing the performance of any individual learning algorithm. Random Trees (RT) presents a viable option for our dataset, as it merges the principles of single model trees with the concepts derived from Random Forest methodologies.

5.3.2 Parameters to validate the models

The primary performance metrics employed in our evaluation are:

5.3.2.1 Cohen's kappa or kappa statistic (κ)

The Kappa statistic serves as a metric that assesses the relationship between Observed Accuracy and Expected Accuracy, which is based on random chance. This statistic is utilized not only for the evaluation of individual classifiers but also for comparing multiple classifiers against one another.

$$k = \frac{observed\ accuracy - expected\ accuracy}{1 - expected\ accuracy} \tag{5.1}$$

Cohen's kappa is always less than or equal to 1. Values of 0 or less, indicate that the classifier is useless. The value needs to be greater than 0 for the classifier to perform better than the chance.

5.3.2.2 Mean absolute error (MAE)

A parameter to measure the closeness of predictions with actual values or results.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y_i}| \tag{5.2}$$

here, $\hat{y_i}$ stands for the prediction value and y_i is the actual outcome, and n is number of observations.

5.3.2.3 Root mean square error (RMSE)

A measurement technique to evaluate the differences between model-predicted values and the actual observed values.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{n}} \qquad (5.3)$$

5.3.2.4 Relative absolute error (RAE)

This ratio compares mean error or residuals to the error generated by a naive model. Hence, a good model is supposed to provide a RAE value less than 1.

$$RAE = \frac{\sum_{i=1}^{n}(P_{(ki)} - T_i)}{\sum_{i=1}^{n}(T_i - \bar{T})} \qquad (5.4)$$

$P_{(ki)}$ is the predicted value of a program 'k' for sample case 'i'. T_i is the target value for the 'i' sample case.

$$\bar{T} = \frac{1}{n}\sum_{i=1}^{n} T_i \qquad (5.5)$$

5.3.2.5 Precision

Represents the proportion of true positive predictions among all predicted positives for each class. A higher precision indicates fewer false positives.

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (5.6)$$

where TP is the number of true positives and FP is the number of false positives.

5.3.2.6 Recall

Reflects the model's ability to correctly identify all relevant samples, measuring the proportion of true positives captured by the model for each class.

$$\text{Precision} = \frac{TP}{TP+FN} \tag{5.7}$$

where FN is the number of false negatives.

5.3.2.7 F_1-Score

The harmonic mean of Precision and Recall, providing a single measure that balances both metrics.

$$F_1\text{-score} = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{5.8}$$

5.3.2.8 Accuracy

The overall percentage of correct predictions across all grades, reflecting the general performance of the model. The formula for Accuracy is:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{5.9}$$

where TN is the number of true negatives.

These performance metrics were computed through a 10-fold cross-validation process, with results for each Grade (A, B, C, D) aggregated into weighted averages to present the overall performance for each algorithm.

5.3.3 Feature selection and data balancing

The distribution of samples of different grades/classes (as per the categorization described above) in the dataset is shown in **Fig. 5.3** (Left). The grade distribution reveals an imbalanced dataset, where Grade A dominates with 143 instances, followed by 119 instances of Grade B samples, with minority samples of Grade C and D with only 37 and 13 instances respectively. To address this, Synthetic Minority Over-sampling Technique (SMOTE) [13] was applied as part of pre-processing to rebalance the datasets. SMOTE helps by generating synthetic samples for the minority classes, thus mitigating the bias introduced by class imbalance and ensuring that the classification models are trained on a more balanced dataset. The SMOTE-rebalanced distribution of samples of different classes is shown in **Fig. 5.3** (Right).

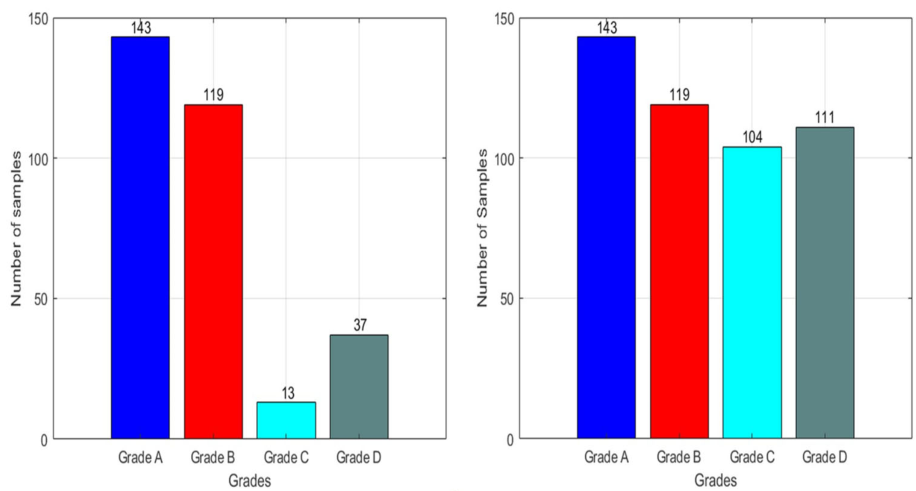

***Figure 5.3**- Original class distribution and class distribution after SMOTE*

Before applying machine learning algorithms, a correlation analysis was performed to explore the relationships among features in the dataset.

The correlation analysis is represented in the heatmap shown in **Fig. 5.4**. Although the correlation coefficients among features were generally low, this does not rule out the potential importance of all features for prediction tasks. As modern machine learning algorithms can capture complex and non-linear relationships, all features were retained for model training. After rebalancing the dataset, different machine learning models were implemented to classify specific capacitance grades and evaluate their predictive performance.

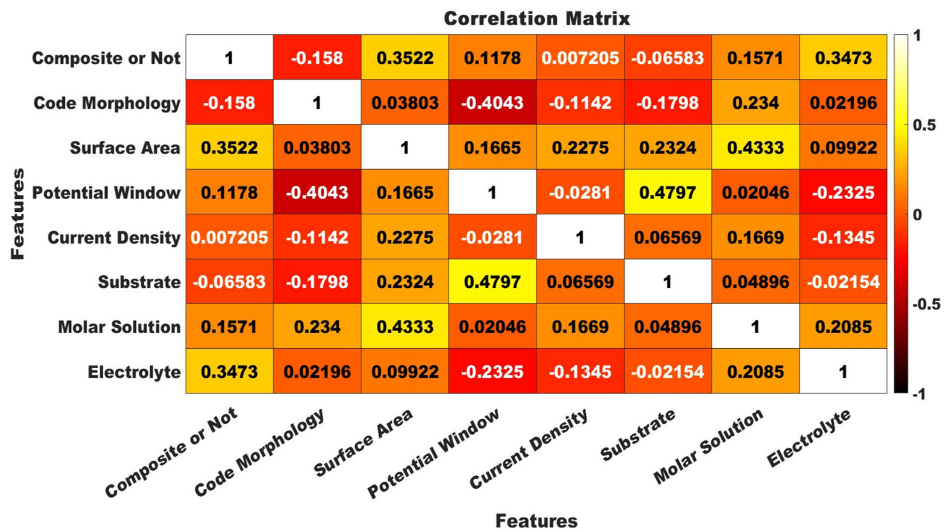

Figure 5.4- *Heatmap of the correlation matrix illustrating the relationships among features*

5.3.4 Electrochemical Experiment

5.3.4.1 Material Synthesis

The Synthesis of highly porous $Ce_2(C_2O_4)_3.10H_2O$ was performed using a precipitation method. The $Ce(NO_3)_3.6H_2O$ was dissolved in 200 ml of deionised water with continuous stirring on a hot plate magnetic stirrer, and 1.27 g of oxalic acid dehydrate ($H_2C_2O_4.2H_2O$) was added in stoichiometric ration in the solution. The entire mixture was stirred vigorously at 80°C for 3 h. After 3 h of stirring, a white coloured precipitate of $Ce_2(C_2O_4)_3.10H_2O$ was obtained.

5.3.4.2 Electrode Preparation

Hydrated $Ce_2(C_2O_4)_3.10H_2O$ working electrodes were prepared using a 7: 2: 1 ratio of active material, AC, and binder (polyvinylidene difluoride, PVDF) in N-methyl-2-pyrrolidone (NMP) solvent. The homogenous slurry was prepared in a mortar, and slurry the (B1 mg) was cast over a 1 cm2 area of Toray carbon paper. The coated electrode was dried at 80°C for 12 h.

5.3.4.3 Electrochemical Testing

All the electrochemical tests were performed in 2M KOH electrolyte, where the $Ce_2(C_2O_4)_3.10H_2O$ and CeO_2, obtained from the TGA studies of the cerium oxalate, were utilised as the working electrodes, using a three-electrode system, where saturated Hg/HgO (1M KOH) was used as a reference electrode, and a platinum wire was used as the counter electrode. Superior specific capacitance equivalent to 78 mA h g-1 (capacitance: 401 F g-1) at 1 A g-1 in the potential window of -0.3 to 0.5 V was observed in an aqueous 2 M KOH electrolyte.

A brief discussion of this work is presented in Chapter Three of this thesis.

5.4 Results and discussion

5.4.1 Grade prediction model

The models are prepared using 10-fold cross-validation. **Table 5.1** and **Fig. 5.6** presents the comparative analysis results of different machine learning algorithms based on different performance measures.

Algorithm	Precision	Recall	F_1 Score	Accuracy
KNN	80.5	80.5	80.5	80.503
Decision Tree	80.9	81.1	80.9	81.132
Logistic Regression	61.9	62.5	61.1	62.473
Naïve Bayes	77.8	78.0	77.8	77.987
CART	79.9	79.9	79.9	79.874
SVM(Linear)	63.0	62.7	61.0	62.683
SVM (RBF)	70.3	69.6	69.8	69.607
SVM (Poly Kernel=2)	72.7	66.7	66.4	66.667
Random Tree	80.6	80.5	80.5	80.503
REP Tree	71.6	72.1	71.5	72.117
MLP	73.3	74.0	73.5	74.004
Voting	85.8	85.7	85.7	85.744
Bagging	84.4	84.3	84.2	84.276
AdaBoost	85.6	85.5	85.5	85.534

Table 5.1*: Precision, Recall, F1 Score, and Accuracy of Various Algorithms*

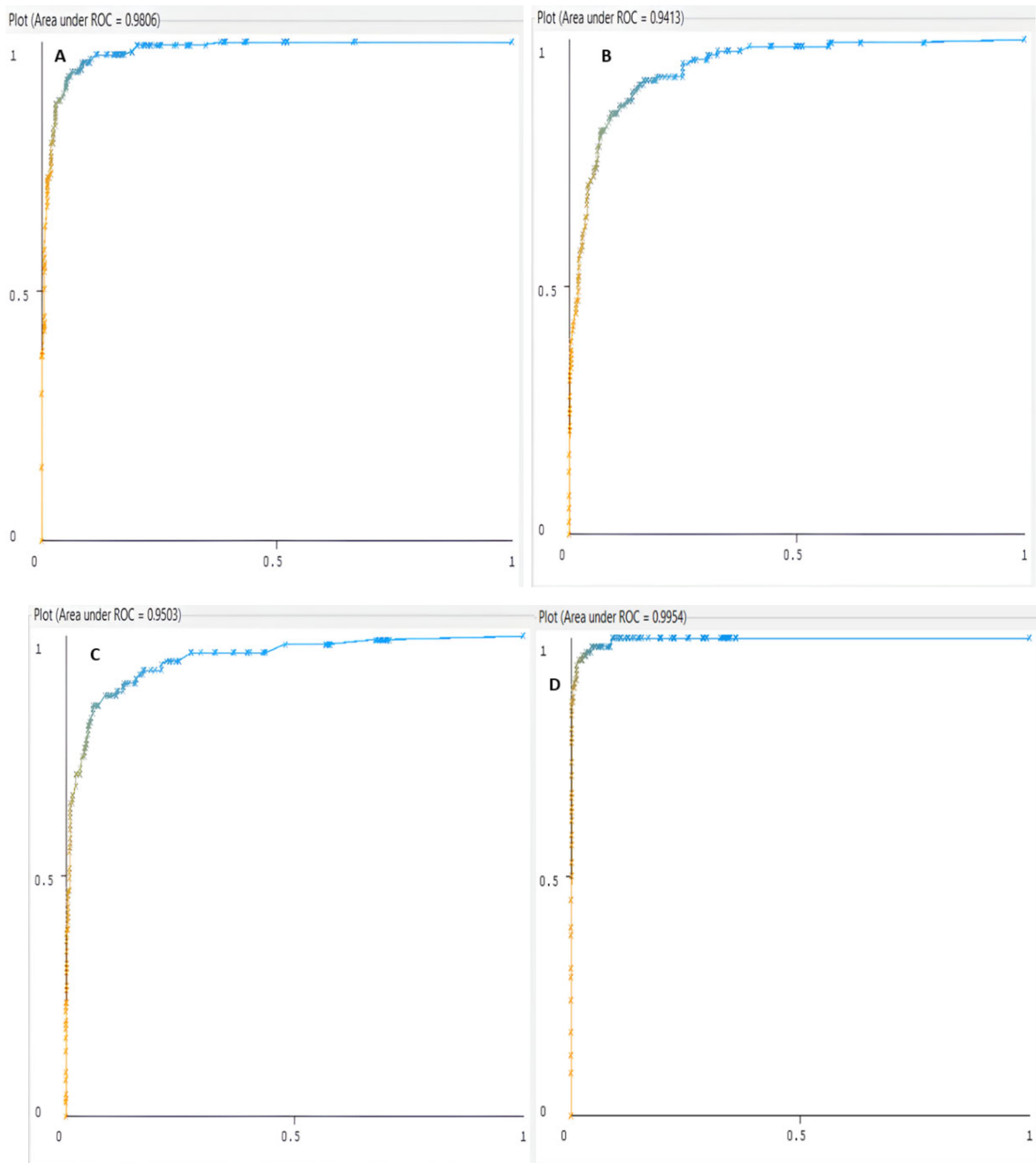

***Figure 5.5**- ROC curve for grade A, B, C and D*

The results highlight significant variability in the performance of the evaluated algorithms. The Voting classifier stands out as the most effective model in this study, achieving a Precision of 85.80, Recall of 85.70, F_1-score of 85.70, and an Accuracy of 85.744. By integrating the predictions from multiple classifiers, it harnesses the strengths of each model, thereby enhancing overall performance. This ensemble method effectively reduces bias and variance, leading to improved reliability in predictions. The consistent high scores in different metrics demonstrate its robustness in handling various complexities within the dataset. The receiver operating characteristic (ROC) curves for the voting classifier for each grade are shown in **Fig.5.5.**

The area under the curve (AUC) values are notably high: 0.9806 for Grade A, 0.9413 for Grade B, 0.9503 for Grade C, and an exceptional 0.9954 for Grade D. These results indicate the model's effectiveness in distinguishing between grades, with particularly robust classification for Grades A and D. Overall, the high AUC values validate the decision to utilize all features for training the Voting Classifier, showcasing its capability to handle class imbalances and optimize predictive accuracy.

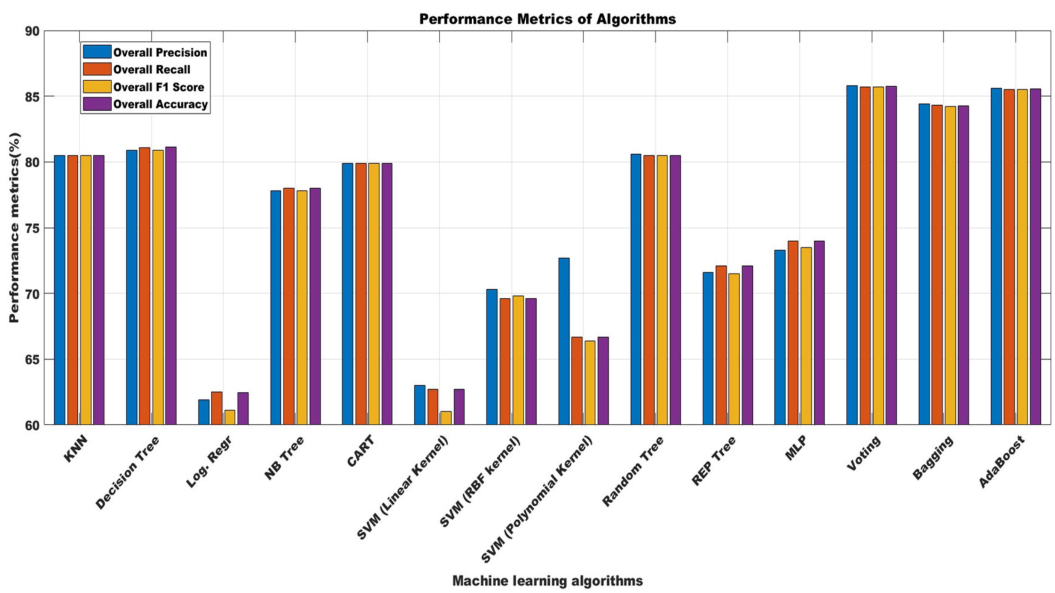

Figure 5.6- *Comparative analysis of different machine learning algorithms*

Decision Tree algorithm employs a tree-like model of decisions, enabling it to classify data by splitting it based on feature values. Its structure allows for clear interpretation of decision pathways, which is critical in applications like material grading where understanding the rationale behind predictions is necessary. The strong Precision and Recall indicate its ability to balance between correctly identifying grades and minimizing misclassifications.

Random Tree a variant of decision trees, uses random feature selection at each split. This approach reduces overfitting and enhances generalization by promoting diversity among individual trees, contributing to its solid performance metrics.

KNN classifies instances based on the majority label among the closest data points in the feature space. This instance-based learning captures local patterns effectively, leading to good performance in distinguishing between grades. However, KNN may become computationally intensive with larger datasets due to its reliance on distance calculations for all instances. Ensemble methods, such as Bagging and AdaBoost, enhance model performance by leveraging the strengths of multiple classifiers. Bagging (Bootstrap Aggregating) improves stability and accuracy by creating multiple versions of a training dataset through resampling and averaging the predictions from separate models, which is particularly beneficial for decision trees that are prone to overfitting. In contrast, AdaBoost (Adaptive Boosting) focuses on misclassified instances by adjusting their weights in subsequent iterations, allowing the model to progressively improve upon weaker classifiers and achieve high Precision and Recall scores.

Lower performing models include Logistic Regression, Support Vector Machines (SVM), Multilayer Perceptron (MLP), and Naive Bayes. Logistic Regression's dependence on linear relationships limits its effectiveness in datasets with complex, non-linear dynamics, resulting in subpar performance metrics despite its simplicity and interpretability. The SVM with linear kernel struggles with non-linear data due to its linear decision boundary, while with the polynomial kernel with degree = 2, despite using a polynomial kernel, fails to significantly improve performance, highlighting the need for appropriate kernel selection based on data characteristics. The MLP may benefit from further tuning of hyperparameters and larger datasets to enhance its ability to learn complex patterns, whereas the Naive Bayes Tree's assumption of feature independence may hinder its effectiveness, suggesting that incorporating techniques to account for feature dependencies could improve its performance.

5.4.2 Value prediction model

The next phase entails projecting the value of specific capacitance, commonly termed specific capacity. In pursuit of this goal, the selected algorithms include Random Forest and Random Committee and the models are prepared using 10-fold cross-validation. The correlation coefficient (R) and mean absolute error (MAE) are calculated for both models. A higher correlation coefficient and lower mean absolute error denote a better model. R is

marginally higher for RF (0.92) than RC (0.91). The value is closer to 1, which indicates a high positive correlation. MAE is lower for RC (68.59) compared to RF (70.2744). Hence, the value-prediction model based on random committee algorithms looks better. **Fig. 5.7** illustrates the relationship between predicted values and actual values of specific capacitance derived from three different value-prediction models. The statistical

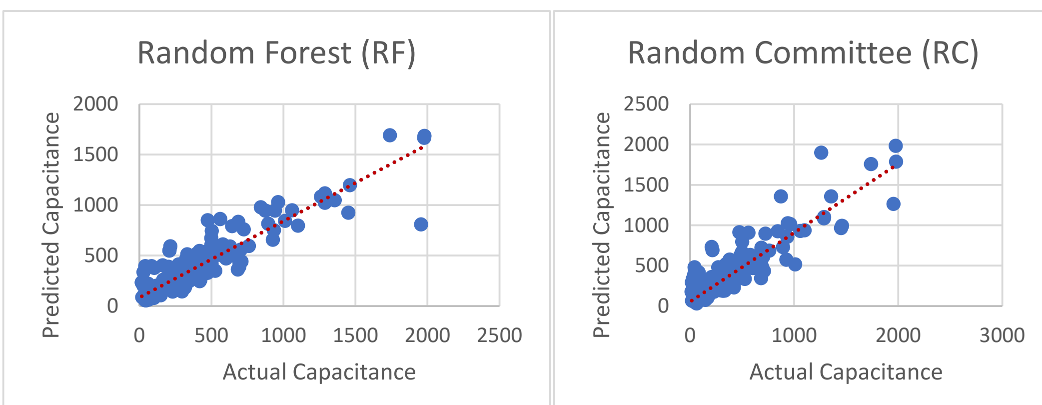

***Figure 5.7**- Predicted value versus actual values of specific capacitance*

parameters are displayed in the table 5.2. From the perspective of primary data science and statistics, the Mean Absolute Error (MAE) may appear substantial when evaluating the models. However, it is important to recognize that the dependent variable in this context, specifically capacitance, is influenced by various attributes previously discussed. Additionally, in a research laboratory setting, even minor alterations in the features can lead to a statistically significant change in the final results, although such changes may not hold practical relevance. Therefore, if a model can forecast outcomes within ± 100 Fg^{-1} of the actual value, it could be deemed significant from an engineering standpoint concerning material selection. While the current model may not be optimal, it remains applicable for practical purposes based on domain knowledge regarding the performance of dedicated materials.

Algorithm	**R**	**MAE**	**RAE**
Random Forest	0.92	70.27	33.00
Random Committee	0.91	68.59	32.21

Table 5.2: *Statistical parameters of ML Algorithms*

5.5 Summary

This work emphasizes the importance of selecting suitable machine learning algorithms for predicting specific capacitance grades based on the unique characteristics of the materials through an extensive comparative analysis of different machine learning algorithms. The ensemble methods, particularly the Voting classifier, demonstrated superior performance, highlighting their effectiveness in accurately classifying materials into grades A, B, C, and D. In the value prediction approach, RC algorithms predict the value of capacitance ~361 F/g. The experimental results (~401 F/g, Grade B) considerably validate the predictive approach presented here. This suggests that further investigation into ensemble techniques may yield enhanced predictive capabilities, especially when combined with hyperparameter tuning and cross-validation strategies.

This research presents data-informed insights regarding material compositions, morphology, electrolytes, surface area, and pore networks, along with their importance in the performance of supercapacitors. The findings from this study can inform experimental efforts related to the synthesis of materials and the fabrication of devices aimed at enhancing the cyclability of supercapacitors. Additionally, this work may encourage researchers to adopt data-driven approaches in the design of empirical investigations.

References

1. Zhu, S., Li, J., Ma, L., He, C., Liu, E., He, F., ... & Zhao, N. (2018). Artificial neural network enabled capacitance prediction for carbon-based supercapacitors. Materials Letters, 233, 294-297.

2. Su, H., Lin, S., Deng, S., Lian, C., Shang, Y., & Liu, H. (2019). Predicting the capacitance of carbon-based electric double layer capacitors by machine learning. Nanoscale Advances, 1(6), 2162-2166.

3. Ghosh, S., Rao, G. R., & Thomas, T. (2021). Machine learning-based prediction of supercapacitor performance for a novel electrode material: Cerium oxynitride. Energy Storage Materials, 40, 426-438.

4. Jha, S., Bandyopadhyay, S., Mehta, S., Yen, M., Chagouri, T., Palmer, E., & Liang, H. (2021). Data-driven predictive electrochemical behavior of lignin-based supercapacitors via machine learning. Energy & Fuels, 36(2), 1052-1062.

5. Rahimi, M., Abbaspour-Fard, M. H., & Rohani, A. (2022). Synergetic effect of N/O functional groups and microstructures of activated carbon on supercapacitor performance by machine learning. Journal of Power Sources, 521, 230968.

6. Singh, M., Mondal, R., Singh, P., & Sharma, N. (2023). Framework structured $Ce_2(C_2O_4)_3\cdot 10H_2O$ as a pseudocapacitive electrode of a hybrid (asymmetric) supercapacitor (HSC) for large scale energy storage applications. Physical Chemistry Chemical Physics, 25(16), 11429-11441.

7. Ding, Z., Chen, Z., Ma, T., Lu, C. T., Ma, W., & Shaw, L. (2020). Predicting the hydrogen release ability of LiBH4-based mixtures by ensemble machine learning. Energy Storage Materials, 27, 466-477.

8. Parsaei, M. R., Rostami, S. M., & Javidan, R. (2016). A hybrid data mining approach for intrusion detection on imbalanced NSL-KDD dataset. International Journal of Advanced Computer Science and Applications, 7(6), 20-25.

9. Wang, Z., Wang, Y., Zeng, R., Srinivasan, R. S., & Ahrentzen, S. (2018). Random Forest based hourly building energy prediction. Energy and Buildings, 171, 11-25.

10. Niranjan, A., Nutan, D. H., Nitish, A., Shenoy, P. D., & Venugopal, K. R. (2018, April). ERCR TV: Ensemble of random committee and random tree for efficient anomaly classification using voting. In 2018 3rd international conference for convergence in technology (I2CT) (pp. 1-5). IEEE.

11. Devi, T. S., & Sundaram, K. M. (2016). A comparative analysis of meta and tree classification algorithms using WEKA. Int. Res. J. Eng. Technol, 3, 77-83.

12. Kalmegh, S. (2015). Analysis of weka data mining algorithm reptree, simple cart and randomtree for classification of indian news. International Journal of Innovative Science, Engineering & Technology, 2(2), 438-446.

13. Nitesh, V. Chawla. "SMOTE: synthetic minority over-sampling technique." J Artif Intell Res 16.1 (2002): 321