REVISTAMATĒRIA V.29 N.3
ISSN 1517-7076 articles e20240256, 2024

An optimization-based stacked ensemble regression approach to predict the compressive strength of self-compacting concrete

Kokila Sekar[1], Rajagopalan Varadarajan[1], Venkatesan Govindan[1]

[1]Anna University, University College of Engineering, Department of Civil Engineering. Tiruchirappalli, India.
e-mail: kokila21@gmail.com, vrg@aubit.edu.in, gv@aubit.edu.in

ABSTRACT

This research paper presents a study on predicting the compressive strength of self-compacting concrete (SCC) containing glass aggregate. A stacked ensemble approach was employed, which is a method of combining multiple models to improve the overall performance. The ensemble consisted of gradient boosting, extreme gradient boost, random forest, and K-nearest neighbors regressors as base learners, and linear regression as the meta learner. The SCC components, namely, water-binder ratio (w/b), total binder content, fine aggregates, fine glass aggregates (FGA), coarse aggregates, coarse glass aggregates (CGA), and superplasticizer were taken as input variables and compressive strength as output variables. The hyperparameters of the base learners were optimized using tree based pipeline optimization (TPOT). The ensemble's accuracy was evaluated using the K-fold cross-validation technique and statistical metrics. The performance of the stacked ensemble models is found to be better than other machine learning models. Permutation feature importance was used to determine the importance of the features in predicting compressive strength. The results demonstrate that the stacked ensemble approach with $R^2 = 0.9866$, RMSE = 1.4730, and MAE = 1.0692 performed better than the individual base learners and the other machine learning models. The water-binder ratio has the highest impact on predicting the compressive strength of SCC containing glass aggregate.

Keywords: Self-compacting concrete; Extreme gradient boot; K-nearest neighbors regressors.

1. INTRODUCTION

In the current situation, the rapid worldwide growth of data prediction models and analytical approaches plays a significant role in almost every field. These prototypes and methods are augmented by using data science concepts because data science helps in doing smart and sensible work and exposure in various sectors like healthcare, learning associations, sensor-based smart farming, weather forecasting, prediction, etc. [1]. However, data sciences have handled every corner of growing technologies and have become a resilience in science and engineering. Moreover, data science has also provided various divisions such as machine learning (ML), artificial intelligence (AI), artificial neural network (ANN) and deep learning, etc. [2].

These divisions give a better understanding of learning, causal relationship, working over precious datasets, multiple and heterogeneous data sources, a computing system for data-intensive applications, data privacy, etc. However, above these divisions, machine learning is one of the most powerful and trending technologies [3]. It is also a well-known application of artificial intelligence (AI). Machine learning generally focuses on developing the learning skills into a machine or a model through observations, past experiences, and training datasets [4]. It encompasses different programs and algorithms to design a learning model and improve itself whenever exposed to new datasets. However, machine learning has a considerable application domain, such as classification, prediction, data analytics, regression, extraction, clustering, learning association, and image and speech recognition [5]. Among these applications, prediction and analysis is one of the most suitable application domains, which are based on historical datasets to identify future probability and give a more accurate evaluation of future probabilities [6]. Moreover, machine learning has become a vital part of every discipline, and worldwide, researchers have concentrated more on machine learning applications in each field. In recent years, construction and infrastructure development have been growing worldwide. Machine learning technologies help researchers, engineers, and concrete practitioners working in this sector and develop the spread of knowledge on their material [7].

Corresponding Author: Kokila Sekar Received on 14/05/2024 Accepted on 03/07/2024

DOI: https://doi.org/10.1590/1517-7076-RMAT-2024-0256

One of the progressive revelations in the construction industry in recent decades is the introduction of self-compacting concrete (SCC) [8]. SCC has an exciting property that enables it to flow freely and fill in the mould under its own weight without getting segregated, even below heavily reinforced sections [9–11]. A prominent claim against SCC is that it is more expensive than vibrational concrete [12]. However, it is likely to be more cost-effective because it saves money on labor, vibrating equipment, and casting time compared to conventional concrete [13]. Various waste materials such as fly ash, copper slag, plastic waste, waste tyre rubber, and rice husk ash are being used as by-products in concrete [14–18]. These materials can substitute any of the basic ingredients of self-compacting concrete. If the precise proportion is determined, this approach saves a significant amount of total construction cost and improves specific qualities of the self-compacting concrete so formed [14].

Global annual production of glass is about 130 million metric tons. Despite the efficient recoverability of glass, only 27 million tons are recovered worldwide, about 21% of the total glass produced [19]. It is reported that India produces more than 21 million tons of glass products every year. However, only 45 percent of the total volume is recycled [20]. It is evident from these figures that the waste glass must be properly managed and procured. Glass is non-biodegradable and comparatively more stable than paper, rubber, and plastic waste [21]. It takes over a million years to decompose if embedded in the soil. So, a gainful use of waste glass in self-compacting concrete (SCC) production would be captivating [22]. Glass is a unique inert material that could be recycled many times without changing its chemical properties. In other words, bottles can be crushed into cullet, then melted and made into new bottles without significant changes to the glass properties [23]. Glass is produced in many forms, including packaging or container glass (bottles and jars), flat glass (windows and windscreens), bulb glass (light globes) and cathode ray tube glass (TV screens, monitors, etc.), all of which have a limited life in the form they are produced and need to be reused/recycled to avoid environmental problems that would be created if they were to be stockpiled or sent to landfill [23]. Recycling this waste by converting it to aggregate not only saves landfill space but also reduces the demand for the extraction of natural raw materials for construction activity [24]. Recycled glass is one such material used by numerous researchers worldwide to successfully replace coarse and fine aggregates in developing sustainable self-compacting concrete [25–29].

MALIK *et al.* [28] and ADAWAY *et al.* [29] reported an increase in slump values in the case of crushed glass-incorporated concrete mixes compared to the control mix. The results obtained by incorporating crushed glass in self-compacting concrete were favorable concerning flow-ability properties. LIU *et al.* [30] observed an increase in the flow-ability of self-compacting concrete when replaced cement and sand partially with 2.2% ground glass, while a reduction in slump flow diameter was observed in replacing sand and cement with 4.3% recycled glass. WANG and HUANG [31] used waste liquid crystal display (LCD) glass to replace different proportions of fine aggregates in SCC. The authors concluded that the increase in flowability with the incorporation of LCD glass sand was primarily because of the hydrophobic nature of the glass. KOU and POON [32] used recycled glass to replace up to 30% of the river sand and up to 15% of coarse aggregate in making the SCC concrete mixtures. It is observed that there is no significant change after 28 days. Compressive strength was also observed at 15% total glass aggregate replacement.

As per Indian standards, strength tests are usually performed from 3 to 28 days after pouring the self-compacting concrete in a cubic (15 × 15 × 15 cm) sample [33]. The period of 28 days causes a delay in the construction work, but neglecting the test would be limited. Therefore, developing rapid and reliable prediction techniques of strength properties in construction industries for pre-design and quality control is needed. However, machine learning provides better prediction models and algorithms such as support vector machine (SVM), artificial neural network (ANN), regression trees (RL), random forest (RF), linear regression (LR), water cycle algorithm (WCA), and decision trees (DT), etc. [3, 34].

These models and algorithms have become more accessible and highly impact in civil engineering. Few studies have been done using machine learning techniques to predict recycled glass aggregate strength properties based on self-compacting concrete. These studies have helped civil engineers estimate various parameters, such as project scheduling, quality control, time, cost, etc., in the construction and infrastructure sectors [35]. Recently, CatBoost regressor have been shown to have better performance than artificial neural network (ANN), and gradient tree boosting in predicting the compressive strength from the SCC mixes [36]. In another study, multiple linear regression, random forest regression, decision tree regression and support vector regression have been exploited for the accurate prediction of compressive strength [37]. Apart from ANN, adaptive fuzzy inference system and extreme learning machine regressor have been utilized for the real-time prediction of compressive strength in SCC [38].

The prediction of the compressive strength of self-compacting concrete is a crucial task for ensuring the durability and structural performance of concrete structures. Although several studies have explored various methods for predicting the compressive strength of self-compacting concrete containing glass aggregate, no research has yet investigated the use of stacked ensemble models. Therefore, in this research, we propose a stacked ensemble model consisting of Gradient Boosting Regressor, XGBoost Regressor, Random Forest Regressor, K-Nearest Neighbors Regressor as base learners, and Linear Regression as the meta learner.

The ensemble approach combines the strengths of multiple machine learning models to improve the accuracy and robustness of the predictions. In addition, we compared the performance of the stacked ensemble with other machine learning models, which can provide insight into the effectiveness of the proposed approach. The potential benefits of this research include enhancing the accuracy and reliability of compressive strength predictions and improving the quality control of self-compacting concrete structures in the construction industry.

2. MATERIALS AND METHODS

2.1. Proposed approach

Figure 1 describes the fundamental steps in creating the stacked ensemble to predict the compressive strength of self-compacting concrete (SCC) containing glass aggregate as a partial replacement of fine and coarse aggregate. (1) A dataset is created from the collected data. (2) Preprocessing the dataset. (3) Splitting the dataset into train and test sets. (4) Employing the base learners on the train set. (5) Tuning the hyper parameters of the base learners using a cross-validation technique. (6) Employing the meta-learner to combine the predictions of the base learners. (7) Evaluating the performance of the stacked ensemble on the test set.

2.2. Data collection and preprocessing

The dataset of SCC containing glass aggregate as a partial replacement of fine and coarse aggregate was collected from 11 published literature. Table 1 illustrates the various literature and the number of samples they contributed. Water to binder ratio (w/b), total binder content, fine aggregate, fine glass aggregate (FGA), coarse aggregate, coarse glass aggregate and superplasticizer (%) are chosen as the input features, and compressive strength (CS) is selected as the target feature. Table 2 provides a brief statistical analysis of each feature variable. Pearson's correlation coefficient was used to understand the relationships between the features. Figure 2 shows the correlation coefficient that measures the strength and direction of the linear relationship between two variables. We found a positive correlation between the total binder content and the CS, indicating that the CS increased as the total binder content increased. Additionally, we observed a strong negative correlation between the compressive strength and the water-binder ratio, indicating that the compressive strength increased as the water-binder ratio decreased. It is clear that each input feature correlates with the output feature. The input features in the dataset all have different magnitudes and scales because they are all numerical variables [39]. Therefore, the numerical values are scaled using the maximum absolute value. The entire dataset considered for this study is shown in a supplementary material.

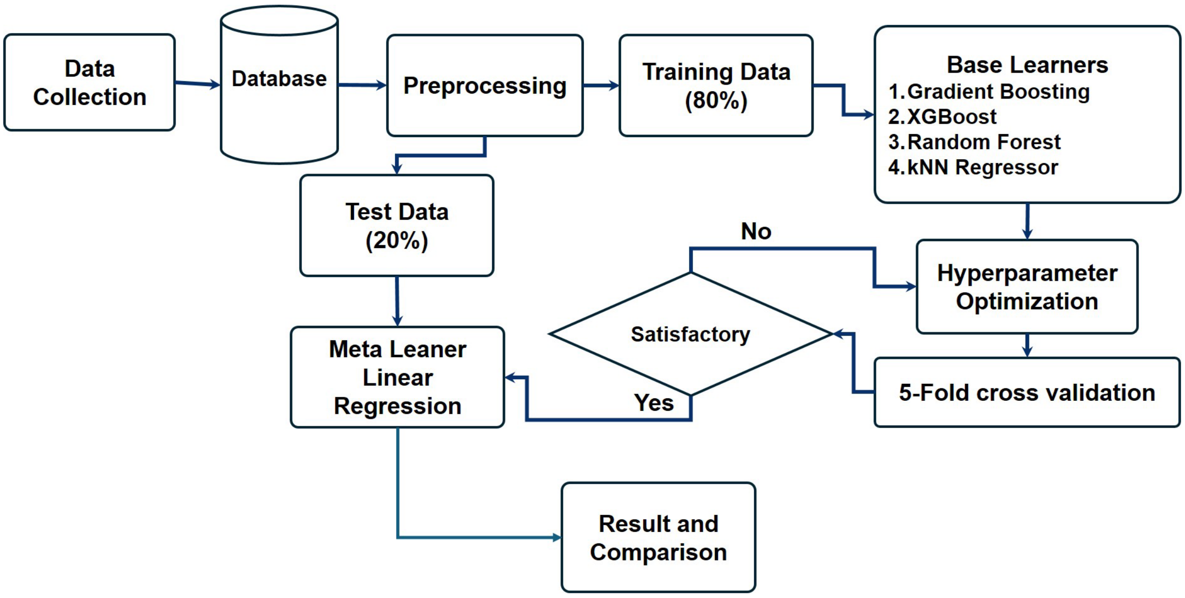


Figure 1: Flowchart of this study.

Table 1: Number of samples and their reference.

S. NO	REFERENCE	NUMBER OF SAMPLES
1	[21]	24
2	[40]	5
3	[41]	9
4	[42]	19
5	[43]	5
6	[44]	16
7	[45]	6
8	[46]	6
9	[47]	18
10	[48]	4
11	[32]	4

Table 2: Statistical analysis.

VARIABLES	TYPE	MEAN	STANDARD DEVIATION	MINIMUM	MAXIMUM
w/b	Input	0.39	0.04	0.28	0.52
Total Binder Content (kg/m^3)	Input	508.8	73.15	365.00	661.00
Fine Aggregate (kg/m^3)	Input	693.14	184.48	0.00	960.00
FGA (kg/m^3)	Input	175.21	179.73	0.00	807.00
Coarse Aggregate (kg/m^3)	Input	722.75	181.45	0.00	936.00
CGA (kg/m^3)	Input	62.73	150.52	0.00	722.70
Superplasticizer (%)	Input	1.24	0.49	0.66	3.10
CS (MPA)	Output	50.77	12.49	23.33	86.50

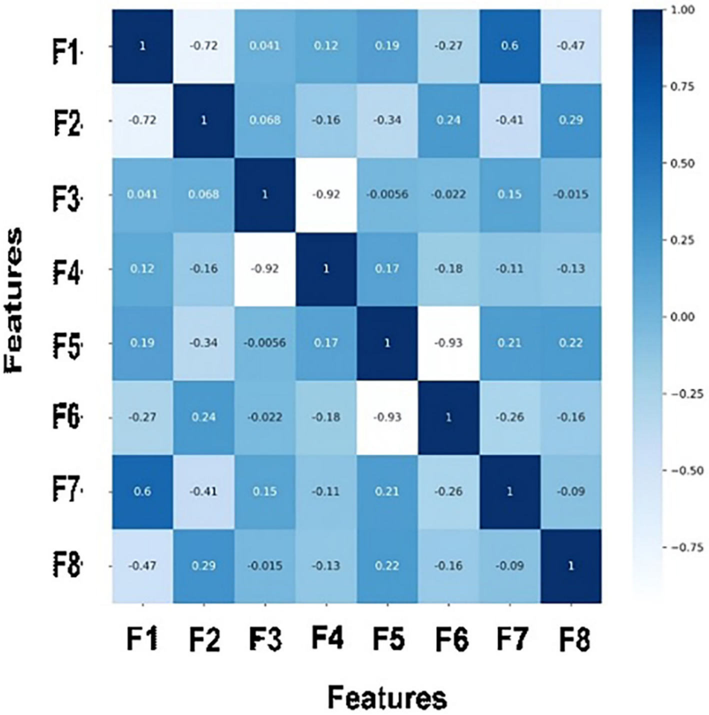


Figure 2: Pearson's correlation matrix for features namely F1: W/b, F2: Total binder content (Kg/m^3), F3: Fine aggregate (Kg/m^3), F4: FGA, F5: Coarse aggregate (Kg/m^3), F6: CGA (Kg/m^3), F7: Super plasticizer (%), and F8: CS(MPA).

2.3. Stacking

Stacking is a method of ensembling, or combining the predictions of multiple models, to improve the overall performance of a model [49–50]. In stacking, a second-level model is trained to learn the relationships between the predictions of the first-level models and the target variables. The first-level models are called base learners, and the second-level model is called the meta-learner. The primary concept behind the stacking model is to first train on the original data with the first layer of base learners, then creates a new dataset from the output of each base learner, and then train on this new dataset and produces the final predictions with the second layer of meta-learners [40]. Figure 3 depicts the framework of creating a stacked ensemble. Figure 4 shows the pseudo-code of stacking algorithms [51]. Creating a new dataset from the first-level classifiers is necessary for the stacking training phase. There is a significant risk of over fitting if the same data used to train the first-level learner is also utilized to produce a new dataset for training the second-level learner. ZHOU [52] suggest a k-fold cross-validation technique to avoid this risk of over fitting, which will be discussed in further sections.

2.4. Selection of learners

The base learners were selected based on model performance and diversity. Therefore, three ensemble models namely gradient boosting (GBR), extreme gradient boosting and random forest regressors are selected. The k-nearest neighbor (k-NN) is included in the combination to increase the diversity of the overall model. By using a diverse set of base models, the stacked model can learn a broader range of relationships in the data and be more robust to over fitting. Linear regression is a simple and effective choice for the meta learner in a stacked model, especially if the base learners are already complex and capture non-linear relationships in the data.

2.5. Gradient boosting regressor (GBR)

Gradient Boosting Regressor (GBR) belongs to the family of boosting algorithms [53]. Figure 4 shows the pseudo-code for boosting algorithms. JEROME FRIEDMAN formulated the first version of gradient boosting in 1999 [54]. Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model stage-wise as other boosting methods do and generalizes them by allowing optimization of an

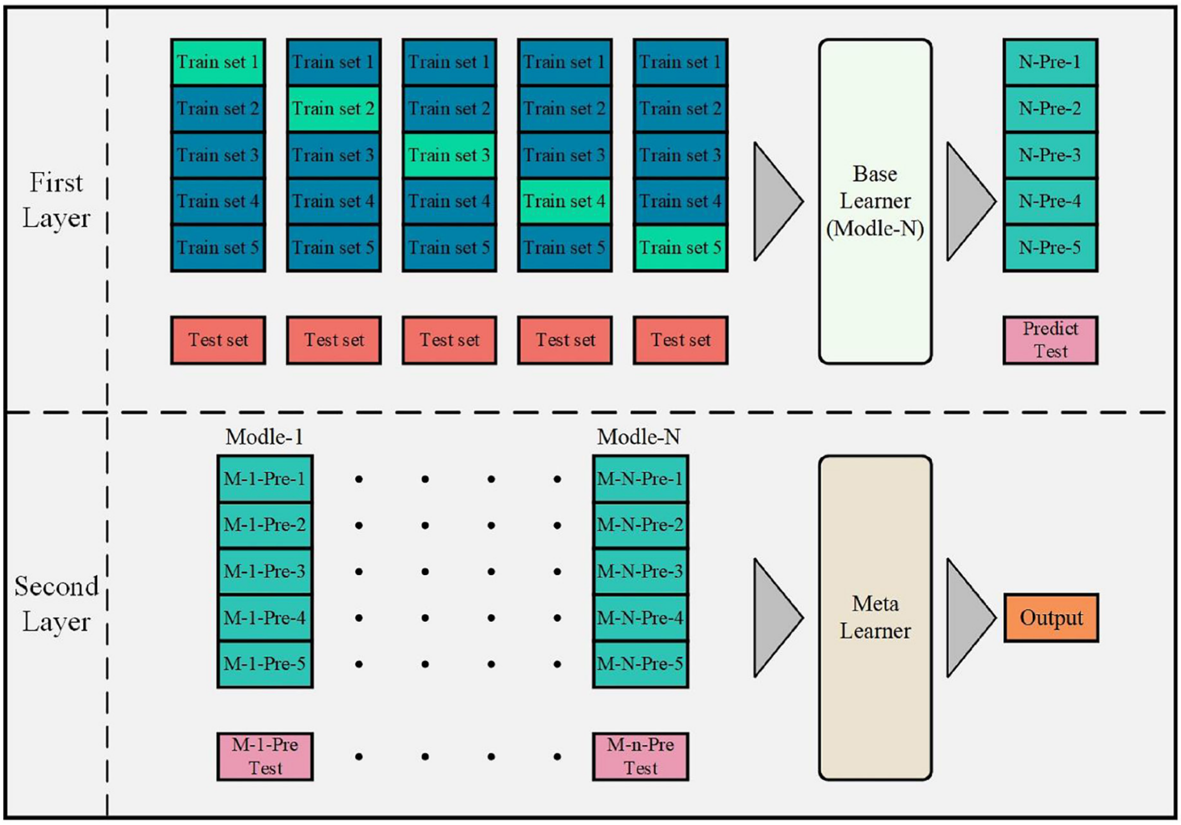
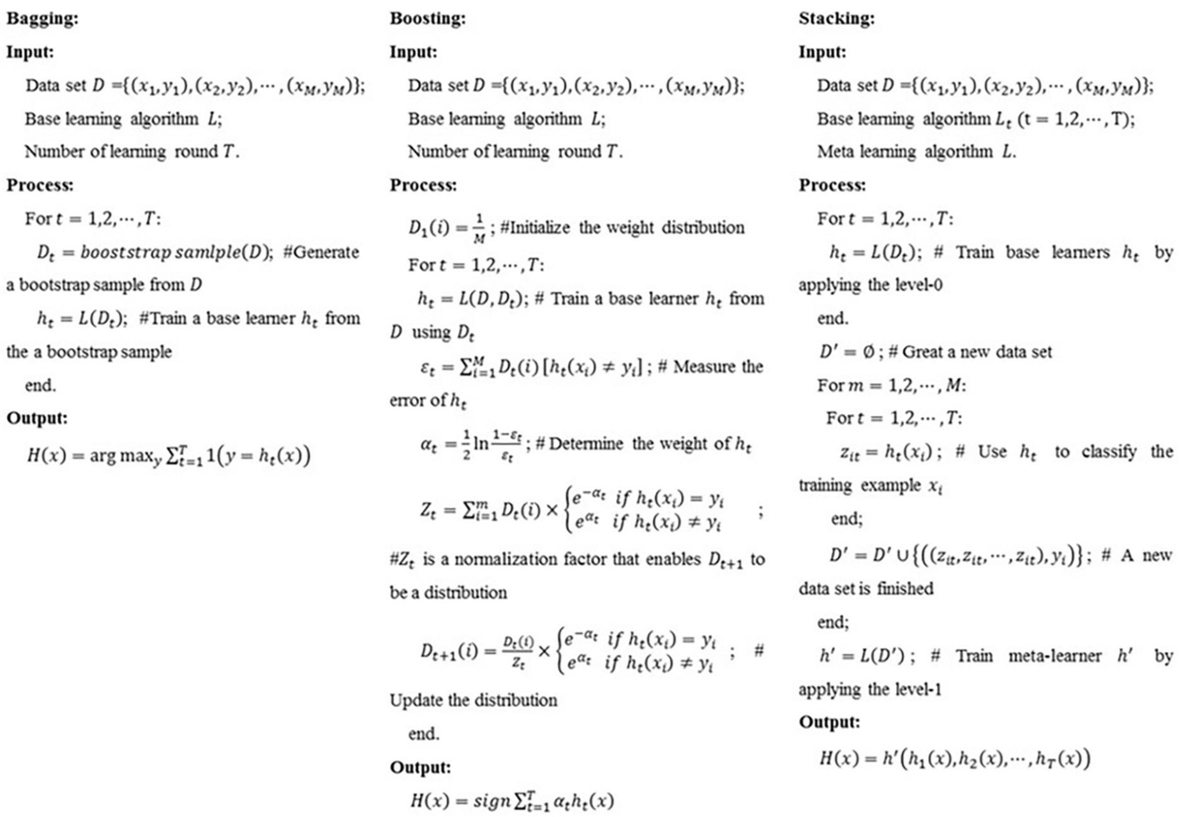


Figure 3: Architecture of stacking with five-fold cross validation.



```
Bagging:
Input:
  Data set $D = \{(x_1,y_1),(x_2,y_2),\cdots,(x_M,y_M)\}$;
  Base learning algorithm $L$;
  Number of learning round $T$.
Process:
  For $t = 1,2,\cdots,T$:
    $D_t = boostsstrap\ samlple(D)$; #Generate a bootstrap sample from $D$
    $h_t = L(D_t)$; #Train a base learner $h_t$ from the a bootstrap sample
  end.
Output:
  $H(x) = \arg\max_y \sum_{t=1}^{T} 1(y = h_t(x))$
```

```
Boosting:
Input:
  Data set $D = \{(x_1,y_1),(x_2,y_2),\cdots,(x_M,y_M)\}$;
  Base learning algorithm $L$;
  Number of learning round $T$.
Process:
  $D_1(i) = \frac{1}{M}$; #Initialize the weight distribution
  For $t = 1,2,\cdots,T$:
    $h_t = L(D, D_t)$; # Train a base learner $h_t$ from $D$ using $D_t$
    $\varepsilon_t = \sum_{i=1}^{M} D_t(i)\,[h_t(x_i) \neq y_i]$; # Measure the error of $h_t$
    $\alpha_t = \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t}$; # Determine the weight of $h_t$
    $Z_t = \sum_{i=1}^{m} D_t(i) \times \begin{cases} e^{-\alpha_t} & if\ h_t(x_i) = y_i \\ e^{\alpha_t} & if\ h_t(x_i) \neq y_i \end{cases}$;
    #$Z_t$ is a normalization factor that enables $D_{t+1}$ to be a distribution
    $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & if\ h_t(x_i) = y_i \\ e^{\alpha_t} & if\ h_t(x_i) \neq y_i \end{cases}$; # Update the distribution
  end.
Output:
  $H(x) = sign \sum_{t=1}^{T} \alpha_t h_t(x)$
```

```
Stacking:
Input:
  Data set $D = \{(x_1,y_1),(x_2,y_2),\cdots,(x_M,y_M)\}$;
  Base learning algorithm $L_t$ (t = 1,2,⋯,T);
  Meta learning algorithm $L$.
Process:
  For $t = 1,2,\cdots,T$:
    $h_t = L(D_t)$; # Train base learners $h_t$ by applying the level-0
  end.
  $D' = \emptyset$; # Great a new data set
  For $m = 1,2,\cdots,M$:
    For $t = 1,2,\cdots,T$:
      $z_{it} = h_t(x_i)$; # Use $h_t$ to classify the training example $x_i$
    end;
    $D' = D' \cup \{((z_{it}, z_{it}, \cdots, z_{it}), y_i)\}$; # A new data set is finished
  end;
  $h' = L(D')$; # Train meta-learner $h'$ by applying the level-1
Output:
  $H(x) = h'(h_1(x), h_2(x), \cdots, h_T(x))$
```

Figure 4: Pseudo-code of ensemble learning algorithm of bagging, boosting, and stacking.

arbitrary differentiable loss function. For regression tasks, the loss function is typically the mean squared error. The mathematics behind the Gradient Boosting Regressor algorithm is represented in equations 1 and 2.

$$F(X) = F(X) + h(X) \tag{1}$$

where F(X)the current approximation of the target function is, h(X) is the new decision tree, and X is the input feature vector. The algorithm iteratively fits new decision trees to the negative gradient of the loss function with respect to the current approximation of the target function F(X), shown in equation 2.

$$-(\partial L(y, F(X)))/\partial F(X) \tag{2}$$

where L(y,F(X)) is the loss function, y is the true output, and F(X) is the predicted output. The idea behind Gradient Boosting is to add weak learners sequentially to the current approximation of the target function, which helps to improve the model's overall performance. The final prediction is obtained by summing the predictions of all decision trees.

2.6. Extreme gradient boosting (XGBoost) algorithm

XGBoost is a scalable ensemble strategy built on gradient boosting that has proven to be a trustworthy and effective machine learning challenge solver [55]. XGBoost was developed by Tianqi Chen and was first released in 2014 [56]. XGBoost is an ensemble additive model made up of several base learners. The XGBoost model, in comparison to the GBDT model, extends the loss function with a second-order Taylor series to increase model accuracy [40]. The XGBoost algorithm uses the complexity of the tree as a constant term, C, in the objective loss function to prevent over fitting [57]. The mathematical expressions are as follows:

$$Object(t) = \sum_{i=1}^{n} l(yi, \hat{y}_l^t) + \Omega(f_t) + C \tag{3}$$

$$\hat{y}_l^t = \hat{y}_l^{t-1} + f_t(x_i) \tag{4}$$

$$\Omega(f_t) = \Upsilon T_t + 1/2\lambda\|w\|^2 \tag{5}$$

The equation consists of variables x_i as the input vector, T_t representing the number of leaves in a tree, hyperparameters Υ and λ, and y_i and $\hat{y}_l^t$ which symbolize actual values and predicted values respectively. The square loss function is represented by l(.) and $f_t(x_i)$ symbolizes a regression tree. From Taylor's formula, the objective function, Object(t) can be approximately expressed as follows.

$$Object(t) \approx \sum_{i=1}^{n}\left[l(y_i, \hat{y}_l^{t-1}) + g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega\left(f_t\right) + C \tag{6}$$

where g_i represents the coefficient of the first term and $\frac{1}{2}h_i$ represents the coefficient of the quadratic term in the Taylor series expansion [58].

2.7. Random forest

Random Forests belong to the bagging family of algorithms. LEO BREIMAN and ADELE CUTLER developed them in the early 1990s. Random forests are a collection of tree predictors where each tree depends on the values of a random vector sampled independently and with the same distribution for all the trees in the forest [59]. Each tree is trained on a random subset of the data, and at each split in the tree, a random selection of features is chosen to consider. This randomization helps reduce the model's variance and generally improves the model's performance. As the number of trees in the forest increases, the generalization error converges a.s. to a limit. The strength of each tree in the forest and the correlation between them affect the generalization error of a forest of tree classifiers [59]. The mathematics behind the Random Forest algorithm is shown in equation 7.

$$F(x) = \frac{1}{T}\sum_{t=1}^{T} f_t(x) \tag{7}$$

where F(x) is the final prediction, T is the number of decision trees in the forest, and $f_t(x)$ is the prediction made by the t-th decision tree.

2.8. K-nearest neighbors

K – Nearest Neighbors (k-NN) is a non-parametric method that uses an instance-based learning approach, or "lazy learning". The k-NN works by finding the k data points in the training set closest to the sample point and then predicting the value of the sample point based on the average of the values of those neighboring points. The k value is the critical parameter that determines the performance of the k-NN algorithm [60]. The k-NN algorithm was first introduced by THOMAS COVER and PETER HART in 1967 [61]. The mathematics behind the k-NN algorithm is shown in equation 8.

$$\hat{y} = 1/\mathrm{k}\sum x \in N_k(x)y_i \tag{8}$$

where $\hat{y}$ is the predicted output, x is the input feature vector, $N_k(x)$ is the set of k-nearest neighbors of x, x_i is an element in $N_k(x)$, and y_i is the corresponding output of x_i. The distance metric used to find the nearest neighbors can be Euclidean, and Manhattan distance. The mathematical equations for the Euclidean, and Manhattan, distances are shown in equations 9, and 10 respectively.

$$d(x, x') = \sqrt{}\sum (x_i - x'_i)^2 \tag{9}$$

$$d(x, x') = \sqrt{}\sum |x_i - x'_i| \tag{10}$$

where x and x' are the input feature vectors, x_i and x'_i are the individual elements of x and x', n is the number of features and p is a parameter that can be any positive real number.

2.9. Tree-based pipeline optimization (TPOT)

Tree-based Pipeline Optimization Tool (TPOT) is a state of art automated machine learning framework implemented in the python programming languages as Deep Evolutionary Algorithms in Python (DEAP). It optimizes machine learning pipelines using genetic algorithm. The machine learning pipeline operators include feature pre-processing, feature selection and classification.

The feature pre-processing operators include StandardScaler, RobustScaler, MinMaxScaler, Max AbsScaler, RandamizedPCA. In this study, MaxAbsScaler is adapted to normalize the features. VarianceThreshold, SelectKBest, SelectPercentile, and Recursive Feature Elimination are the feature selection operators used to reduce the dimension of the feature set. The classification operators contain a group of conventional machine learning algorithms, such as, random forest, decision tree, k-nearest neighbour, and logistic regression. These machine learning pipeline operators are used by genetic algorithm to construct tree-based pipelines. The trees constructed by genetic programming has shown to have flexible representation of machine learning pipelines. The genetic algorithm optimizes the learning models by performing bioinspired operations known as mutation, crossover and selection. The algorithm searches for the best combination of machine learning pipelines rather than relying on a predefined set of hyperparameters.

In this work, the TPOT was executed on the dataset for 10 generations with 25 population size. For further details, the readers can refer the article by the authors OLSON and MOORE [62].

2.10. Cross validation

Cross-validation is a resampling technique used to assess the performance of machine learning models on a limited data sample. Cross-validation tests the model's ability to generalize to an independent dataset to tune the model's hyperparameters and validate its performance. One popular method of cross-validation is k-fold cross-validation, where the data samples are splitted into k equal size r sample sets. Each sample set is used for testing exactly once, and the model is trained on the remaining sample sets [63]. This procedure is repeated k times, using a different sample set as the testing data. The performance measure is then averaged across all k iterations. Figure 5 shows the schematic representation of k-fold cross-validation [64].

2.11. Evaluation metrics

Evaluation metrics measure the quality of the statistical or machine learning models. The metrics used in this study are the coefficient of determination (R^2), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). R-squared is a measure of the goodness of fit of the regression model. The value of R^2 ranges from 0 to 1. The calculation of R^2 is shown by equation (11). MAE is the average of the absolute differences between predicted and actual values. The calculation of MAE is shown by equation (12). RMSE is the square root of the mean squared error. It is often used because it is in the same units as the original data, making it easier to interpret. The calculation of RMSE is shown by equation (13). Here, n stands for the sample count in the dataset, y_i for the actual value, $\hat{y}_i$ for its predicted value, and $\overline{y}$ for the average of the sample's predicted values.

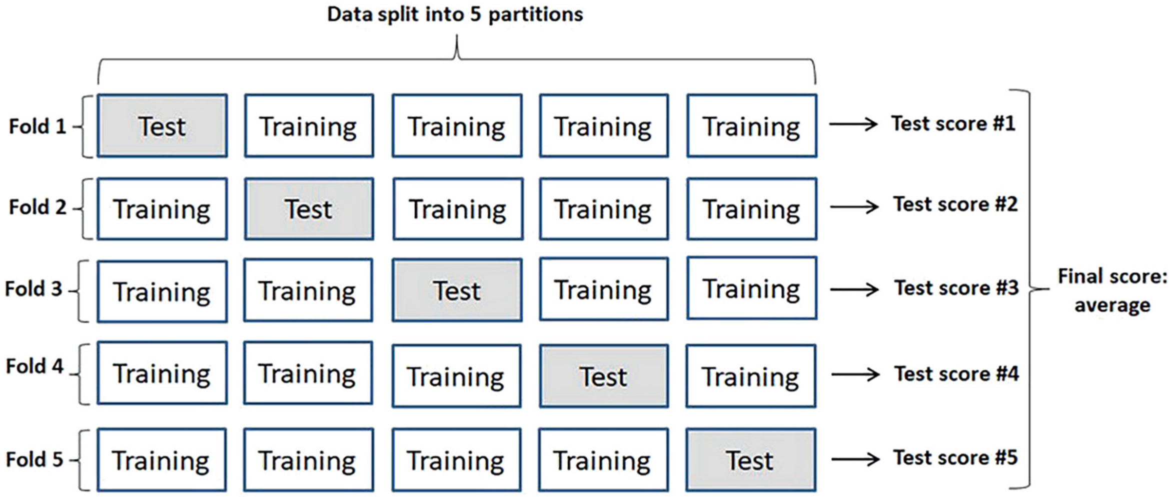


Figure 5: Schematic representation of k-fold cross-validation.

$$R^2 = 1 - \sum_{i=1}^{n}(y_i - \hat{y}_l)^2 / \sum_{i=1}^{n}(y_i - \overline{y})^2 \tag{11}$$

$$MAE = \frac{1}{n}\sum_{\mathrm{i}=1}^{\mathrm{n}}| y_i - \hat{y}_1 | \tag{12}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_l)^2} \tag{13}$$

2.12. Permutation feature importance

Permutation feature importance is a method for estimating the importance of individual features in a machine-learning models. It works by shuffling the values of a single feature and then calculating the change in the model’s performance. This change in performance is then used to infer the importance of the feature. The idea behind permutation feature importance is that if a feature is important, then shuffling its values should significantly degrade the model’s performance, whereas shuffling the values of an unimportant feature should have little to no effect on the model’s performance. The technique is proposed to measure the importance of features in random forests [59]. It is also reported that the permutation approach yields better results than in-built feature importance techniques [65].

3. RESULTS AND DISCCUSSION

3.1. Experiment setup

All the experiments are carried out using JupyterLab (Bisong, 2019) and python 3 is used as the programming language. A computer with 8GB RAM and AMD Ryzen 5 4600H CPU processor working at a clock frequency of 3.00 GHz processor is used for this research. Matplotlib and seaborn libraries are used for visualization, and the scikit-learn [66] and XGBoost libraries are used for employing the machine learning models.

3.2. Hyperparameter optimization results

A genetic search algorithm as a TPOT implementation with 5-fold cross-validation and the R^2 score as the evaluation metric was used to optimize the hyperparameters of the base learners. It has been reported that simple TPOT configurations perform better and takes less time to execute [62]. The optimized hyperparameters of each model are shown in the Table 3.

3.3. Performance of individual base learners

The properties of the base learners can significantly affect the prediction properties of the final stacking model, i.e., the base learners’ overall accuracy will affect the stacking model’s accuracy. Suppose the base learners are

Table 3: Optimized hyperparameters.

BASE MODEL	HYPERPARAMETERS
GBR	‘alpha’: 0.8, ‘learning_rate’: 0.5, ‘loss’: ‘lad’, ‘n_estimators’: 100
XGBR	‘n_estimators’: 1910, ‘learning_rate’: 0.09
RFR	‘n_estimators’: 100
KNN	‘n_neighbors’: 3
GBR	‘alpha’: 0.8, learning_rate’: 0.5, ‘loss’: ‘lad’, ‘n_estimators’: 100

diverse and make predictions using different algorithms or approaches the ensemble can capture a broader range of patterns in the data, leading to improved performance. Therefore, this research first analyzed the performance of the base learners employed independently.

Figure 5 compares the 5-fold cross-validation results of the optimized base learners on the training set. Table 4 compares the performance of the base learners on the test set before and after hyperparameter optimization. The table also shows that the XGBoost model performs the best, followed by GBR and KNN, which performs the worst. Figure 6 illustrates the correlation between the predicted and actual values on the test set. Despite having less accuracy, KNN is included in the stacked ensemble because KNN assumes that points that are similar in the feature space are likely to have the same label; it can be effective at identifying patterns in the data that more complex models do not easily capture.

3.4. Different combinations of base learners

This section tests the performance of different combinations of base learners to evaluate the best possible combination. The LR model was used as the meta learner. Table 5 shows the prediction results of stacked ensembles with different combinations of base learners. k-NN performs well in combination with XGBR. This might be because KNN is a non-parametric method that relies on similarity. At the same time, XGB is a tree-based model capable of learning complex non-linear relationships. The stacked model can thus benefit from the strengths of both approaches. The combination of XGBR, GBR, and k-NN has similar accuracy to that of GBR, XGBR,

Table 4: Performance of the base learners before and after hyperparameter optimization.

BASE MODELS	BEFORE HYPERPARAMETER OPTIMIZATION			AFTER HYPERPARAMETER OPTIMIZATION		
	R^2	RMSE (MPA)	MAE (MPA)	R^2	RMSE (MPA)	MAE (MPA)
GBR	0.964	2.387	2.077	0.982	1.6889	1.320
XGBR	0.976	1.948	1.604	0.983	1.6183	1.285
RFR	0.940	3.118	2.709	0.959	2.5758	2.129
KNN	0.450	9.453	7.842	0.846	4.9947	3.771

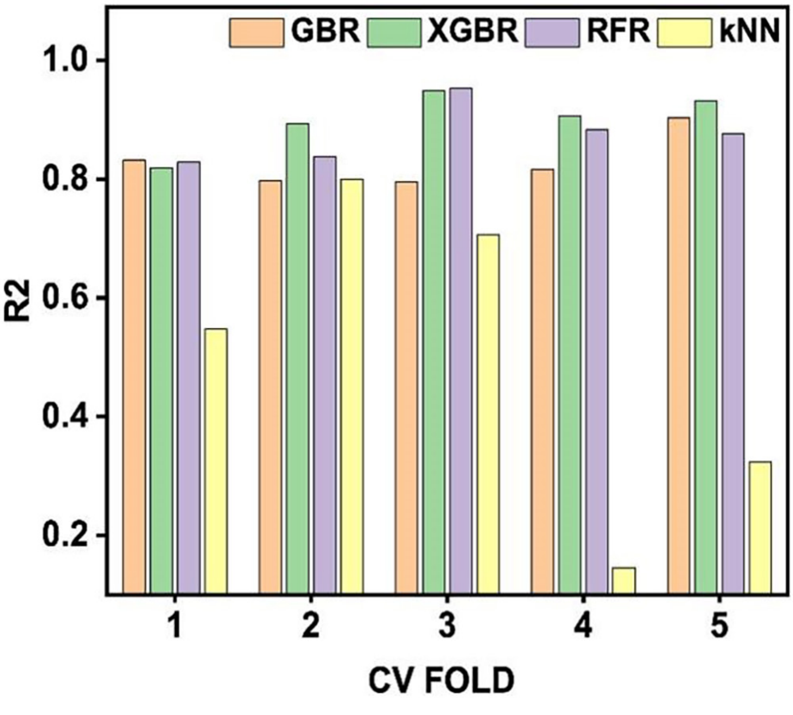


Figure 6: Five-fold CV results of the optimized base learners on the training set.

RFR, and k-NN. Since the combination of all four models has the least RMSE, it is chosen as the best model because RMSE is usually preferred over MAE as RMSE is known to place a greater weight on larger errors, as the errors are squared before being averaged, compared to mean absolute error (MAE), which averages the absolute values of the errors.

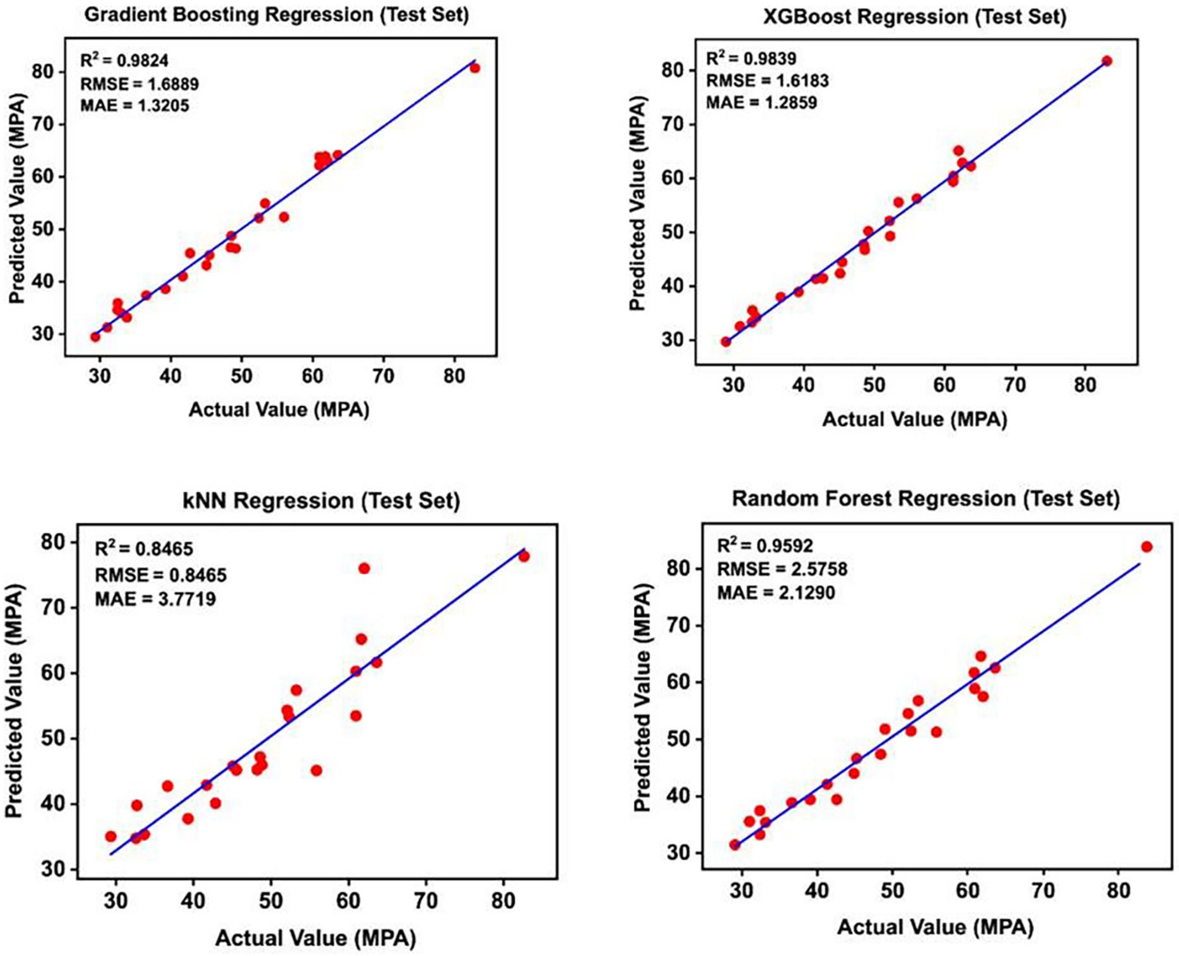
3.5. Performance of proposed stacked ensemble approach

GBR, XGBR, RFR, and KNN were chosen as the base learners, according to section 4.3. The LR model was used as the meta learner. The 5-fold cross-validation results of the proposed stacked ensemble are shown in Figure 7.

The performance results of the stacked ensemble on the test are as follows: R^2 = 0.9866, RMSE = 1.4730, and MAE = 1.0692. The correlation between the predicted and actual values is shown in Figure 8. On comparing

Table 5: Performance of different combinations of base learners.

BASE LEARNER COMBINATIONS	R^2	RMSE (MPA)	MAE (MPA)
XGBR + RFR + KNN	0.9841	1.6054	1.2291
GBR + XGBR + RFR	0.9853	1.5451	1.1764
GBR + RFR + KNN	0.9759	1.9793	1.5612
XGBR + GBR + KNN	0.9866	1.4739	1.0292
GBR + XGBR + RFR + KNN	0.9866	1.4730	1.0692



Gradient Boosting Regression (Test Set)

XGBoost Regression (Test Set)

kNN Regression (Test Set)

Random Forest Regression (Test Set)

Figure 7: Base learner performance on the test set.

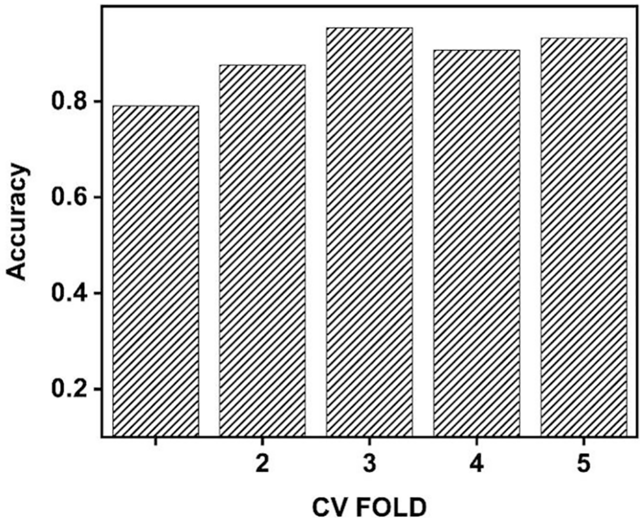


Figure 8: 5-fold CV results of the stacked ensemble on the training set.

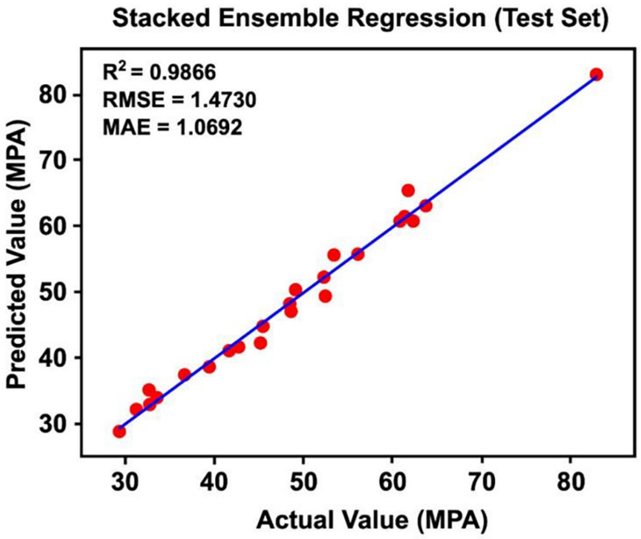


Figure 9: Stacked ensemble performance on the testing set.

the performance results of the proposed state-of-the-art stacked ensemble with the base learners (GBR, XGBR, RFR, KNN), the stacked ensemble outperforms each base learner taken individually. The superior performance of the stacked model might be that it can learn from the strengths of multiple individual models rather than relying on the predictions of a single model. The stacked model combines the predictions of these individual models in a more accurate way than any of the models alone, resulting in improved overall performance. Figure 9 shows the performance of stacked ensemble regression model evaluated on test data.

3.6. Comparison with other machine learning models

This section demonstrates the stacked ensemble's superiority by contrasting its performance with other machine learning models. The metrics used for the model evaluation are R^2, RMSE, and MAE. 80% of the data were utilized as the training set and 20% as the test set after the data had been normalized. The hyperparameters optimization technique discussed in Table 3 is used to tune the hyperparameters of the models. Table 6 compares the performance results of the stacked ensemble with that of LR and SVR. The RMSE and MAE were 85% and 87% less in the stacking model compared to the LR model, while the R^2 was found to be 137% high. The stacked ensemble and SVR comparison results revealed that the RMSE and MAE decreased by 72% and 75%, respectively, while R^2 is increased by 18%. The results show that the stacked model significantly outperformed both SVR and LR.

Table 6: Performance of different combinations of base learners.

BASE LEARNER COMBINATIONS	R^2	RMSE (MPA)	MAE (MPA)
XGBR + RFR + KNN	0.9841	1.6054	1.2291
GBR + XGBR + RFR	0.9853	1.5451	1.1764
GBR + RFR + KNN	0.9759	1.9793	1.5612

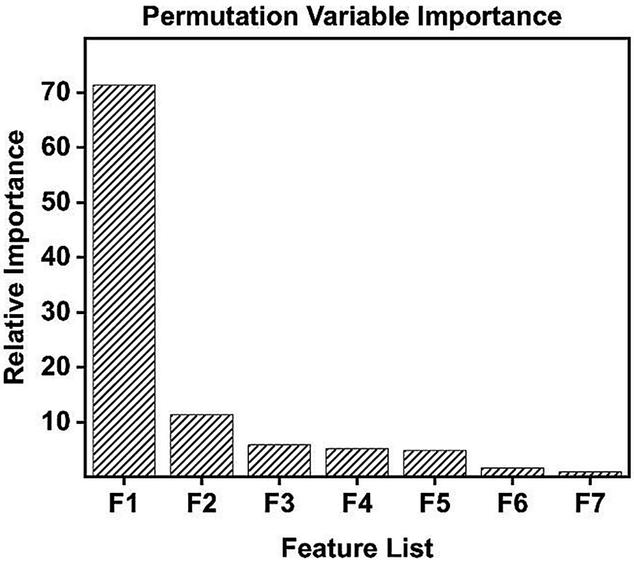


Figure 10: Importance of the input features.

3.7. Permutation feature importance

Permutation feature importance (PFI) was performed on the stacked ensemble for 100 iterations on the test set. Permuting a feature and calculating the resulting change in model performance can be a noisy process. Performing PFI for 100 iterations, the noise associated with a single permutation is averaged out, resulting in a more stable estimate of feature importance.

At each iteration, the same permuted dataset is fed into each base learner, and the PVI is the reduction in the meta-learner mean squared error. Therefore, the output will be a combined variable importance, boosting performance and robustness in a manner like output prediction [62]. Figure 10 show the results of PFI. From the figure, that the most important features in our model were the water-to-binder ratio (w/b), followed by the coarse aggregate, and fine glass aggregate (FGA). This suggests that these features strongly influence the target variable and should be considered when developing strategies for improving model performance. On the other hand, features fine aggregate, superplasticizer (%), total binder content, and coarse glass aggregate (CGA) were found to have minimal impact on the model's prediction accuracy, indicating that they may not be as relevant for predicting the target variable.

4. CONCLUSIONS

In conclusion, the research paper presents a study on predicting the compressive strength of self-compacting concrete (SCC) containing glass aggregate using a stacked ensemble approach. The ensemble consisted of GBR, XGBR, RFR, and k-NN as base learners and LR as meta learner and was optimized using the TPOT Regressor. The results demonstrate that the stacked ensemble approach performed better than the individual base learners and other machine learning models, with an R^2 value of 0.9866, RMSE of 1.4730, and MAE of 1.0692. Additionally, the study found that the water-binder ratio had the highest impact on predicting the compressive strength of SCC containing glass aggregate. This research highlights the effectiveness of using a stacked ensemble approach and the importance of the water-binder ratio in predicting the compressive strength of SCC containing glass aggregate. In addition to the findings of this study, there are several areas for future research. The study could be expanded to include other types of concrete, such as high-performance concrete, to determine if the stacked ensemble approach is effective for predicting the compressive strength of these materials as well. Furthermore, exploring different ensemble methods, such as bagging and boosting, to compare the performance of the stacked ensemble approach with other ensemble methods. Finally, in order to improve the prediction accuracy, incorporating more features and using more sophisticated models, such as deep neural networks, can be considered as future research directions.

5. BIBLIOGRAPHY

[1] KAYA KELEŞ, M., "An overview: the impact of data mining applications on various sectors", *Tehnički Glasnik*, v. 11, n. 3, pp. 128–132, 2017.

[2] ONGSULEE, P., "Artificial intelligence, machine learning and deep learning", In: *Fifteenth International Conference on ICT and Knowledge Engineering*, pp. 1–6, 2017. doi: http://doi.org/10.1109/ICTKE.2017.8259629.

[3] GARCÍA, J.A., VARÓN, F.A.P., "Neural network-based model to predict compressive strength of concrete incorporating supplementary cementitious materials and recycled aggregates", *Matéria (Rio de Janeiro)*, v. 27, pp. e13218, 2023.

[4] JORDAN, M.I., MITCHELL, T.M., "Machine learning: trends, perspectives, and prospects", *Science*, v. 349, n. 6245, pp. 255–260, Jul. 2015. doi: http://doi.org/10.1126/science.aaa8415. PubMed PMID: 26185243.

[5] LIBBRECHT, M.W., NOBLE, W.S., "Machine learning applications in genetics and genomics", *Nature Reviews. Genetics*, v. 16, n. 6, pp. 321–332, May. 2015. doi: http://doi.org/10.1038/nrg3920. PubMed PMID: 25948244.

[6] RAMPRASAD, R., BATRA, R., PILANIA, A., *et al.*, "Machine learning in materials informatics: recent applications and prospects", *NPJ Computational Materials*, v. 3, n. 1, pp. 1–13, Dec. 2017. doi: http://doi.org/10.1038/s41524-017-0056-5.

[7] SUN, L., KOOPIALIPOOR, M., JAHED ARMAGHANI, D., *et al.*, "Applying a meta-heuristic algorithm to predict and optimize compressive strength of concrete samples", *Engineering with Computers*, v. 37, n. 2, pp. 1133–1145, Apr. 2021. doi: http://doi.org/10.1007/s00366-019-00875-1.

[8] MEMON, S.A., SHAIKH, S.K., AKBAR, H., "Utilization of rice husk ash as viscosity modifying agent in self compacting concrete", *Construction & Building Materials*, v. 25, n. 2, pp. 1044–1048, Feb. 2011. doi: http://doi.org/10.1016/j.conbuildmat.2010.06.074.

[9] SINGH, G., SIDDIQUE, R., "Strength properties and micro-structural analysis of self-compacting concrete made with iron slag as partial replacement of fine aggregates", *Construction & Building Materials*, v. 127, pp. 144–152, Nov. 2016. doi: http://doi.org/10.1016/j.conbuildmat.2016.09.154.

[10] SHARMA, R., KHAN, R.A., "Durability assessment of self-compacting concrete incorporating copper slag as fine aggregates", *Construction & Building Materials*, v. 155, pp. 617–629, Nov. 2017. doi: http://doi.org/10.1016/j.conbuildmat.2017.08.074.

[11] SILVA, P., DE BRITO, J., "Experimental study of the mechanical properties and shrinkage of self-compacting concrete with binary and ternary mixes of fly ash and limestone filler", *European Journal of Environmental and Civil Engineering*, v. 21, n. 4, pp. 430–453, Apr. 2016. doi: http://doi.org/10.1080/19648189.2015.1131200.

[12] PATHAK, N., SIDDIQUE, R., "Effects of elevated temperatures on properties of self-compacting-concrete containing fly ash and spent foundry sand", *Construction & Building Materials*, v. 34, pp. 512–521, Sep. 2012. doi: http://doi.org/10.1016/j.conbuildmat.2012.02.026.

[13] MOHAN, A., MINI, K.M., "Strength and durability studies of SCC incorporating silica fume and ultra fine GGBS", *Construction & Building Materials*, v. 171, pp. 919–928, May. 2018. doi: http://doi.org/10.1016/j.conbuildmat.2018.03.186.

[14] GUPTA, N., SIDDIQUE, R., BELARBI, R., "Sustainable and greener self-compacting concrete incorporating industrial by-products: a review", *Journal of Cleaner Production*, v. 284, pp. 124803, Feb. 2021. doi: http://doi.org/10.1016/j.jclepro.2020.124803.

[15] SAFI, B., SAIDI, M., ABOUTALEB, D., *et al.*, "The use of plastic waste as fine aggregate in the self-compacting mortars: effect on physical and mechanical properties", *Construction & Building Materials*, v. 43, pp. 436–442, Jun. 2013. doi: http://doi.org/10.1016/j.conbuildmat.2013.02.049.

[16] RAHMAN, M.M., USMAN, M., AL-GHALIB, A.A., "Fundamental properties of rubber modified self-compacting concrete (RMSCC)", *Construction & Building Materials*, v. 36, pp. 630–637, Nov. 2012. doi: http://doi.org/10.1016/j.conbuildmat.2012.04.116.

[17] ABDELALEEM, B.H., HASSAN, A.A.A., "Development of self-consolidating rubberized concrete incorporating silica fume", *Construction & Building Materials*, v. 161, pp. 389–397, Feb. 2018. doi: http://doi.org/10.1016/j.conbuildmat.2017.11.146.

[18] GILL, S., SIDDIQUE, R., "Strength and micro-structural properties of self-compacting concrete containing metakaolin and rice husk ash", *Construction & Building Materials*, v. 157, pp. 51–64, Dec. 2017. doi: http://doi.org/10.1016/j.conbuildmat.2017.09.088.

[19] PADMALATHA, N.A., PRABHISH, S., "Impact of recycling in glass industry: a project management study", *Journal of Social Science Research*, v. 1, pp. 2455–4839, 2016.

[20] GOWTHAM, R., PRABHU, S.M., GOWTHAM, M., et al.., "A review on utilization of waste glass in construction field", *In: Proceedings of IOP Conference Series on Materials Science and Engineering*, v. 1130, 2021. doi: 10.1088/1757-899X/1130/1/012010.

[21] SINGH, H., SIDDIQUE, R., "Utilization of crushed recycled glass and metakaolin for development of self-compacting concrete", *Construction & Building Materials*, v. 348, pp. 128659, Sep. 2022. doi: http://doi.org/10.1016/j.conbuildmat.2022.128659.

[22] SHAYAN, A., XU, A., "Value-added utilization of waste glass in concrete", *Cement and Concrete Research*, v. 34, n. 1, pp. 81–89, Jan. 2004. doi: http://doi.org/10.1016/S0008-8846(03)00251-5.

[23] RAKSHVIR, M., BARAI, S.V., "Studies on recycled aggregates-based concrete", *Waste Management & Research*, v. 24, n. 3, pp. 225–233, Jul. 2016. doi: http://doi.org/10.1177/0734242X06064820. PubMed PMID: 16784165.

[24] KIM, S., CHOI, S.Y., YANG, E.I., "Evaluation of durability of concrete substituted heavyweight waste glass as fine aggregate", *Construction & Building Materials*, v. 184, pp. 269–277, Sep. 2018. doi: http://doi.org/10.1016/j.conbuildmat.2018.06.221.

[25] SONG, W., ZOU, D., LIU, T., *et al.*, "Effects of recycled CRT glass fine aggregate size and content on mechanical and damping properties of concrete", *Construction & Building Materials*, v. 202, pp. 332–340, Mar. 2019. doi: http://doi.org/10.1016/j.conbuildmat.2019.01.033.

[26] EKOP, E., OKEKE, C.J., INYANG, E.V., "Comparative study on recycled iron filings and glass particles as a potential fine aggregate in concrete", *Resources*", *Conservation & Recycling Advances*, v. 15, pp. 200093, Nov. 2022. doi: http://doi.org/10.1016/j.rcradv.2022.200093.

[27] LI, S., ZHANG, J., DU, G., *et al.*, "Properties of concrete with waste glass after exposure to elevated temperatures", *Journal of Building Engineering*, v. 57, pp. 104822, Oct. 2022. doi: http://doi.org/10.1016/j.jobe.2022.104822.

[28] MALIK, M.I., BASHIR, M., AHMAD, S., "Study of concrete involving use of waste glass as partial replacement of fine aggregates", *IOSR Journal of Engineering*, v. 6, pp. 8–15, 2013.

[29] ADAWAY, M., "Recycled glass as a partial replacement for fine aggregate in structural concrete – Effects on compressive strength", *Electronic Journal of Structural Engineering*, v. 14, n. 1, pp. 116–122, Jan. 2015. doi: http://doi.org/10.56748/ejse.141951.

[30] LIU, M., "Incorporating ground glass in self-compacting concrete", *Construction & Building Materials*, v. 25, n. 2, pp. 919–925, Feb. 2011. doi: http://doi.org/10.1016/j.conbuildmat.2010.06.092.

[31] WANG, H.Y., HUANG, W., "A study on the properties of fresh self-consolidating glass concrete (SCGC)", *Construction & Building Materials*, v. 24, n. 4, pp. 619–624, Apr. 2010. doi: http://doi.org/10.1016/j.conbuildmat.2009.08.047.

[32] KOU, S.C., POON, C.S., "Properties of self-compacting concrete prepared with recycled glass aggregate", *Cement and Concrete Composites*, v. 31, n. 2, pp. 107–113, Feb. 2009. doi: http://doi.org/10.1016/j.cemconcomp.2008.12.002.

[33] BUREAU OF INDIAN STANDARDS, *Methods of tests for strength of concrete*, New Delhi, BI, 2006.

[34] CHOU, J.S., CHIU, C.K., FARFOURA, M., *et al.*, "Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques", *Journal of Computing in Civil Engineering*, v. 25, n. 3, pp. 242–253, May. 2011. doi: http://doi.org/10.1061/(ASCE)CP.1943-5487.0000088.

[35] ASSI, L.N., DEAVER, E., ELBATANOUNY, M.K., *et al*., "Investigation of early compressive strength of fly ash-based geopolymer concrete", *Construction & Building Materials*, v. 112, pp. 807–815, Jun. 2016. doi: http://doi.org/10.1016/j.conbuildmat.2016.03.008.

[36] CHAKRAVARTHY, H.G., SEENAPPA, K.M., "Machine learning models for the prediction of the compressive strength of self-compacting concrete incorporating incinerated bio-medical waste ash", *Sustainability (Basel)*, v. 15, n. 18, pp. 13621, 2023. doi: http://doi.org/10.3390/su151813621.

[37] EL ASRI, Y., AICHA, M.B., ZAHER, M., *et al*., "Prediction of compressive strength of self-compacting concrete using four machine learning technics", *Materials Today: Proceedings*, v. 57, pp. 859–866, 2022. doi: http://doi.org/10.1016/j.matpr.2022.02.487.

[38] GHANI, S., KUMAR, N., GUPTA, M., *et al*., "Machine learning approaches for real-time prediction of compressive strength in self-compacting concrete", *Asian Journal of Civil Engineering*, v. 25, n. 3, pp. 2743–2760, 2024. doi: http://doi.org/10.1007/s42107-023-00942-5.

[39] LI, Q., SONG, Z., "Prediction of compressive strength of rice husk ash concrete based on stacking ensemble learning model", *Journal of Cleaner Production*, v. 382, pp. 135279, Jan. 2023. doi: http://doi.org/10.1016/j.jclepro.2022.135279.

[40] SHIVA, K.S., LALITHA, G., "Mechanical performance of sustainable ternary blended self-compacting concrete with waste crushed glass", *Materials Today: Proceedings*, v. 60, pp. 394–398, Jan. 2022. doi: http://doi.org/10.1016/j.matpr.2022.01.259.

[41] RAHAT DAHMARDEH, S., SARGAZI MOGHADDAM, M.S., MIRABI MOGHADDAM, M.H., "Effects of waste glass and rubber on the SCC: rheological, mechanical, and durability properties", *European Journal of Environmental and Civil Engineering*, v. 25, n. 2, pp. 302–321, 2021. doi: http://doi.org/10.1080/19648189.2018.1528891.

[42] OULDKHAOUA, Y., BENABED, B., ABOUSNINA, R., *et al*., "Effect of using metakaolin as supplementary cementitious material and recycled CRT funnel glass as fine aggregate on the durability of green self-compacting concrete", *Construction & Building Materials*, v. 235, pp. 117802, Feb. 2020. doi: http://doi.org/10.1016/j.conbuildmat.2019.117802.

[43] ARABI, N., MEFTAH, H., AMARA, H., *et al*., "Valorization of recycled materials in development of self-compacting concrete: mixing recycled concrete aggregates – Windshield waste glass aggregates", *Construction & Building Materials*, v. 209, pp. 364–376, Jun. 2019. doi: http://doi.org/10.1016/j.conbuildmat.2019.03.024.

[44] AL-BAWI., R.K., KADHIM, I.T., AL-KERTTANI, O., "Strengths and failure characteristics of self-compacting concrete containing recycled waste glass aggregate", *Advances in Materials Science and Engineering*, n. 1, pp. 6829510, 2017.

[45] DAHMARDEH, S.R., "Advances in environmental biology utilization of waste glass in architectural self-compacting concrete: a novel approach for waste management", *Advances in Environmental Biology*, 2014.

[46] SHARIFI, Y., HOUSHIAR, M., AGHEBATI, B., "Recycled glass replacement as fine aggregate in self-compacting concrete", *Frontiers of Structural and Civil Engineering*, v. 7, n. 4, pp. 419–428, Dec. 2013. doi: http://doi.org/10.1007/s11709-013-0224-8.

[47] ALI, E.E., AL-TERSAWY, S.H., "Recycled glass as a partial replacement for fine aggregate in self compacting concrete", *Construction & Building Materials*, v. 35, pp. 785–791, Oct. 2012. doi: http://doi.org/10.1016/j.conbuildmat.2012.04.117.

[48] WANG, H.Y., HUANG, W.L., "Durability of self-consolidating concrete using waste LCD glass", *Construction & Building Materials*, v. 24, n. 6, pp. 1008–1013, Jun. 2010. doi: http://doi.org/10.1016/j.conbuildmat.2009.11.018.

[49] WOLPERT, D.H., "Stacked generalization", *Neural Networks*, v. 5, n. 2, pp. 241–259, 1992. doi: http://doi.org/10.1016/S0893-6080(05)80023-1.

[50] BREIMAN, L., "Stacked regressions", *Machine Learning*, v. 22, n. 1, pp. 49–64, 1996. doi: http://doi.org/10.1007/BF00117832.

[51] HU, X., MEI, H., ZHANG, H., *et al*., "Performance evaluation of ensemble learning techniques for landslide susceptibility mapping at the Jinping county, Southwest China", *Natural Hazards*, v. 105, n. 2, pp. 1663–1689, Jan. 2021. doi: http://doi.org/10.1007/s11069-020-04371-4.

[52] ZHOU, Z.H., *Ensemble methods: foundations and algorithms*, Hoboken, CRC press, 2012. doi: http://doi.org/10.1201/b12207.

[53] SCHAPIRE, R.E., "A Brief Introduction to Boosting", In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

[54] FRIEDMAN, J.H., "Greedy function approximation: a gradient boosting machine", *Annals of Statistics*, v. 29, n. 5, pp. 1189–1232, 2001. doi: http://doi.org/10.1214/aos/1013203451.

[55] BENTÉJAC, C., CSÖRGŐ, A., MARTÍNEZ-MUÑOZ, G., "A comparative analysis of gradient boosting algorithms", *Artificial Intelligence Review*, v. 54, n. 3, pp. 1937–1967, 2021. doi: http://doi.org/10.1007/s10462-020-09896-5.

[56] CHEN, T., GUESTRIN, C., "XGBoost: a scalable tree boosting system," In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* v. 13-17-, pp. 785–794. Aug.2016. doi: http://doi.org/10.1145/2939672.2939785.

[57] YING, L., "Application of XGBoost algorithm in prediction of students grades performance," In: *Proceedings of IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology*, pp. 93–96., Sep. 2021. doi: http://doi.org/10.1109/CEI52496.2021.9574453.

[58] DAVAGDORJ, V.H., PHAM, N., THEERA-UMPON, N., *et al.*, "XGBoost-based framework for smoking-induced non communicable disease prediction", *International Journal of Environmental Research and Public Health*, v. 17, n. 18, pp. 1–22, Sep. 2020. doi: http://doi.org/10.3390/ijerph17186513. PubMed PMID: 32906777.

[59] BREIMAN, L., "Random forests", *Machine Learning*, v. 45, n. 1, pp. 5–32, 2001. doi: http://doi.org/10.1023/A:1010933404324.

[60] QIAN, Y., ZHOU, W., YAN, J., *et al.*, "Comparing machine learning classifiers for object-based land cover classification using very high resolution imagery", *Remote Sensing (Basel)*, v. 7, n. 1, pp. 153–168, 2015. doi: http://doi.org/10.3390/rs70100153.

[61] COVER, T.M., HART, P., "Nearest neighbor pattern classification", *IEEE Transactions on Information Theory*, v. 13, n. 1, pp. 21–27, 1967. doi: http://doi.org/10.1109/TIT.1967.1053964.

[62] OLSON, R.S., MOORE, J.H., "TPOT: a tree-based pipeline optimization tool for automating machine learning", In: *Workshop on Automatic Machine Learning,* pp. 66–74, 2016.

[63] BERRAR, D. "Cross-validation", *Encyclopedia of Bioinformatics and Computational Biology*, v. 1, pp. 542–545, 2019. doi: http://doi.org/10.1016/B978-0-12-809633-8.20349-X.

[64] PHUNG, V.H., RHEE, E.J., "A High-Accuracy Model Average Ensemble of convolutional neural networks for classification of cloud image patches on small datasets", *Applied Sciences (Basel, Switzerland)*, v. 9, n. 21, pp. 4500, Nov. 2019. doi: http://doi.org/10.3390/app9214500.

[65] BARTON, B., LENNOX, B., "Model stacking to improve prediction and variable importance robustness for soft sensor development", *Digital Chemical Engineering*, v. 3, pp. 100034, Jun. 2022. doi: http://doi.org/10.1016/j.dche.2022.100034.

[66] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., *et al.*, "Scikit-learn: machine learning in Python", *Journal of Machine Learning Research*, v. 12, pp. 2825–2830, 2011.

SUPPLEMENTARY MATERIAL

The following online material is available for this article:

Table S1 – The details of SCC components collected from the literatures are listed below.