Prediction of Effective Stiffness in Rectangular RC Columns Using Ensemble and Boosted Machine Learning Models

Juanjuan Wang[1], Yetao Cong[2], Xine Yan[1,*], Zhanhao Zhang[1]
[1]Xi'an Traffic Engineering University, Xi'an 710300, Shaanxi, China
[2]Xinjiang Beixin Road & Bridge Group Co.,Ltd, Urumqi 830000, Xinjiang, China
E-mail: wang1228109631@163.com
[*]Corresponding author

Keywords: effective stiffness, reinforced concrete, stacking ensemble, averaging ensemble, machine learning

Received: August 10, 2025

The Reinforced Concrete (RC) columns are one of the main structural elements that estimation of their effective stiffness is essential for trustworthy structural system design and analysis. As traditional approaches, both analytical and experimental, often involve simplifications that may not accurately represent real behavior, Machine Learning (ML) approaches offer a viable substitute. This work presents a flexible and high-performance ML-based ensemble framework in phyton environment based on the Stacking and Averaging techniques to improve the prediction of effective stiffness of RC structures. For developing ML models, this study utilizes a comprehensive dataset with 266 samples (80% for training and 20% for testing), which includes axial load capacity, the percentage of reinforcing steel, and the column dimensions and effective stiffness factors. The ML algorithms are trained through k-fold cross-validation and optimized using Grid Search-based hyperparameter tuning to improve prediction accuracy. The Stacking and Averaging ensemble models results the highest predictive accuracy in test-set with R^2 values of 0.9708 and 0.9840, in comparison with Voting, Light Gradient Boosting, DecisionTree, and K-Nearest Neighbors models. Furthermore, the ensembles robustness in capturing nonlinear relationships among features are further confirmed by its consistently low test RMSEs of 0.0182 and 0.0169 values. Moreover, the interpretability analysis makes it clear that the percentage of steel reinforcement is the most influential parameter that have a significant contribution on RC's effective stiffness. This finding demonstrates that such models have great potential for integration into structural design RC processes, enhancing efficiency and reliability in engineering practice.

Povzetek: Predstavljen je ansambelski pristop strojnega učenja za napoved učinkovite togosti armiranobetonskih stebrov, ki združevanjem več modelov in SHAP-razlago.

1 Introduction

The Reinforced Concrete (RC) columns are one of the main structural elements commonly used as vertical load-carrying components in various building and bridge constructions. They transfer loads from the superstructure to the foundation and preserve overall structural stability. These members' stiffness has a major impact on the structure's period, yield displacement, load distribution, and deformation capacity [1]. The ability of a column to withstand deformation under applied loads, particularly under lateral forces caused by wind and earthquakes, is known as effective stiffness [2][3]. Therefore, RC columns must be designed with sufficient stiffness and ductility to withstand seismic events, especially in earthquake-prone regions, as stiffness determines a structure's stability and lateral displacement. While overestimating stiffness can result in unfeasible overdesign, underestimating it can jeopardize safety. Therefore, in RC design, accurate stiffness prediction is essential for both performance and cost-effectiveness.

However, because the stiffness behavior is influenced by many interacting factors, predicting the effective stiffness of RC columns is still a very difficult task [4]. The majority of classical estimation techniques are empirical and rely on condensed formulas that are applied in design codes and obtained from experimental data. The intricate nonlinear interactions between concrete and steel, as well as secondary effects like creep, cracking, shrinkage, and load history, are not captured by these models, which usual take into account parameters like material properties, applied loads, and cross-sectional geometry [5]. The shortcomings of empirical models underscore the need for data-driven techniques that can more accurately model nonlinear, multivariate interactions. Machine Learning (ML) [6] approaches offer a viable substitute in this regard. Multiple interdependent variables can be integrated, hidden relationships between material and geometric factors can be revealed, and more precise and broadly applicable stiffness predictions can be produced using ML-based approaches. ML frameworks can greatly improve the accuracy of stiffness estimation for RC columns by utilizing big datasets and adaptive learning techniques, which will ultimately lead to safer and more effective structural design [7],[8],[9].

Based on the shortcomings of current ML techniques, this study develops a high-performance and flexible

ensemble model based on the Stacking and Averaging techniques. The goal of ensemble models is to accurately predict two crucial structural parameters, namely the radius of gyration about the X-axis (r_x) and the radius of gyration about the Y-axis (r_y), which are linked to the effective stiffness property of RC structures. The suggested model makes use of the stacking and averaging techniques to combine the advantages of several base learners via a meta-model to improve handling of intricate nonlinear and multi-factor interactions, decrease sensitivity to noise, and increase predictive accuracy. The proposed ensemble models are thoroughly compared with other method of Voting, Light Gradient Boosting (LGB), DecisionTree, and K-Nearest Neighbors (KNN), for models' validation and performance benchmarking. It should be mentioned that the incorporation of Grid Search (GS) optimization for hyperparameter tuning of developed ML models is one of the primary innovations of this study, which aims to discover globally optimal hyperparameter configurations. Additionally, to ensure model interpretability and to pinpoint the key characteristics influencing the anticipated responses, a thorough Shapley Additive Explanations (SHAP)-based sensitivity analysis was used. This method not only makes the model more transparent and reliable, but it also offers insightful engineering information about the main structural factors influencing r_x and r_y.

2 Related work

In recent years, the use of ML techniques in RC engineering has grown significantly, providing strong data-driven substitutes for conventional analytical and empirical approaches. For instances, Fan et al. [10] examined the latest developments up to 2021 in ML applications for RC bridges, highlighting how they could revolutionize structural inspection, design, and construction quality control. They discussed about that how ML created new research opportunities in bridge engineering, specifically in the areas of long-span structure form-finding, reinforcement, and structural optimization. Kazemi et al. [11] developed a ML-based framework to predict the maximum-interstory-drift-ratio of RC Moment-Resisting Frames using a large dataset. Among the tested models, improved Artificial Neural Network (ANN) achieved the reasonable accuracy in estimating maximum-interstory-drift-ratio for 3-Story RC frame with three bays, that was introduced to facilitate practical application and reduce computational effort. Yaseen et al. [12] assessed how well three ML models including M5-Tree, Extreme Learning Machine (ELM), and Random Forest (RF) predict the shear strength of reinforced concrete beams. The M5-Tree model outperformed the others with highest accuracy, indicating its high predictive power and the usefulness of ML techniques. Alhalil and Gullu [13] developed ML model to predict torsional irregularity, fundamental period, modal participating mass ratios of RC buildings. Based on the outcomes, the Categorical Boosting (CatBoost) algorithm achieved the best performance with high accuracy in predictions, demonstrating its effectiveness for efficient seismic design. Deger and Kaya [14] introduced a Gaussian Process Regression (GPR)-based ML model to predict backbone curves of RC shear walls, accurately estimating seven critical backbone points. The model achieved highest accuracy with prediction ratios close to 1.0, outperforming traditional methods for seismic performance evaluation. In order to overcome the drawbacks of conventional inspection techniques, Mohammadagha et al. [15] used ANN and multiple linear regression models to forecast condition of RC pipes based on variables like age, material, diameter, and ratings. Based on the outcomes, ANN demonstrated its capacity to model nonlinear deterioration patterns with a higher accuracy. Moreover, Naderpour et al. [16] introduced ML-based methods including Decision Tree and ANN to determine the failure modes in spirally and rectangular RC columns. The Decision Tree model achieved high with simpler computations, making it a practical tool for seismic evaluation, retrofitting, and rehabilitation of RC structures. Table 1 summarize the utilized ML in RC analysis.

Table 1: Summary of ML applications in RC analysis.

References	Model	Reported R^2	Application
Kazemi et al. [11]	ANN	0.868	Estimating maximum-interstory-drift-ratio for 3-story RC frame with three bays
Yaseen et al. [12]	M5-Tree	0.931	Estimating the shear strength of RC
Alhalil and Gullu [13]	CatBoost	0.977 for torsional irregularity, 0.997 for fundamental period, and 0.923 for modal participating mass ratios	Estimating three parameters of torsional irregularity, fundamental period and modal participating mass ratios of RC
Deger and Kaya [14]	GPR	0.90-0.97 for various backbone-curve points	Estimating backbone curves of RC shear walls
Mohammadagha et al. [15]	ANN	0.906	Estimating condition of RC pipes
Naderpour et al. [16]	DecisionTree	0.868-0.935 for spirally RC and 0.950-0.960 for rectangular RC	Estimating failure modes in spirally and rectangular RC columns

Under complex structural behaviors, problems like overfitting, noise sensitivity, higher computational cost and poor management of nonlinear and multi-factor interactions are usual problems of utilized ML models, which can lower prediction accuracy and reliability, particularly during seismic or extreme loading events. In previous works, majority of them utilized deep learning-based ML models on RC applications, because can learn intricate nonlinear relationships between structural and material parameters. The practical application of these models in routine engineering design is limited because they usually require substantial computational resources, sizable datasets, and meticulous tuning to achieve stable and accurate performance. Furthermore, many studies use individual ML models without following strict optimization protocols, which results in less-than-ideal performance, especially when applied to different structural configurations. Furthermore, some studies use tree-based ensemble techniques, like random forest or gradient boosting-based models that usually are based on basic decision tree. Although these ensemble methods somewhat lessen overfitting and increase stability, their dependence on simple base learners' structures frequently limits their ability to fully exploit feature interactions and achieve high predictive accuracy. Lack of advanced interpretability through SHAP Analysis to quantitatively interpret each input feature's contribution to stiffness prediction is one of main drawback of developed models.

3 Methodology

The methodical process used to create ML models for estimating the stiffness factors of RC columns is shown in Fig. 1. In order to ensure that the model accurately depicts the physical behavior of RC members under various loading conditions, the process begins with the formulation of suitable assumptions that define the modeling framework and conditions. Next, Relevant input parameters, such as geometric dimensions along with corresponding target responses, such as stiffness coefficients (r_x and r_y) are identified to create a complete dataset. In the following phase, the dataset is split into 80% for model training and 20% for testing using k-fold cross validation technique. This strategy (with five folds) is used to improve generalization and decrease overfitting, allowing each subset to be validated iteratively. After preparing dataset, the selected models are involved training and testing. In this process, the GS optimization is utilized for hyperparameter tuning of developed ML models. After tarin and test of models, the performance of all potential ML models using a variety of statistical metrics are evaluated to identify the most reliable and accurate predictive approach. The ensemble models are chosen as the best method for estimating effective stiffness. In final step, the SHAP-based sensitivity analysis was used to make the stacking and averaging model more transparent and reliable and offer insightful engineering information about the main structural factors influencing r_x and r_y.

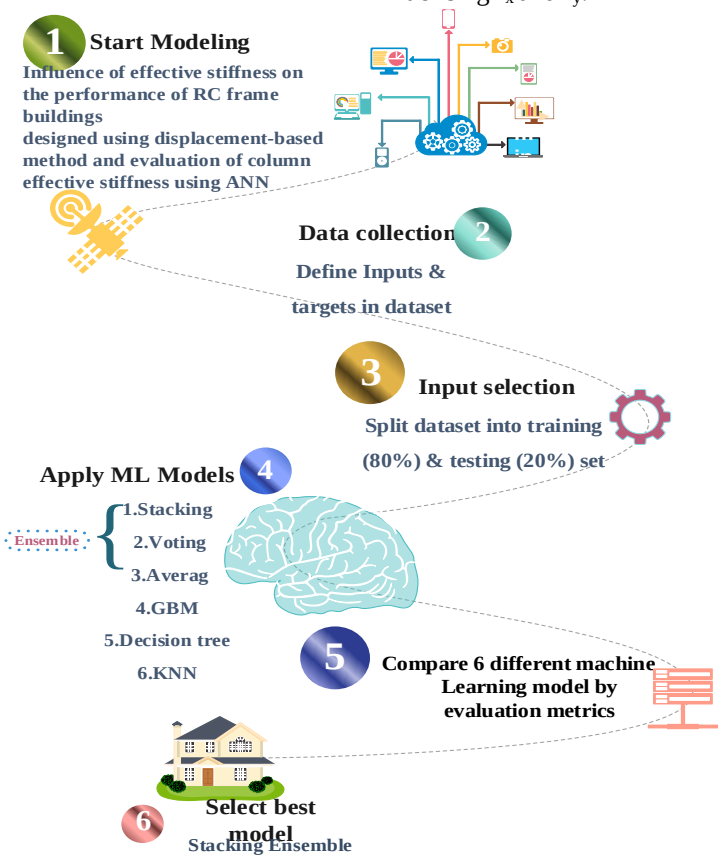


Figure 1: Detailed step-by-step procedure for estimating effective stiffness of rectangular RC column sections by optimizing ML model.

3.1 Implementation information

To guarantee widespread accessibility and reproducibility, the ML framework was created in a flexible Python environment. Python's strong ecosystem for data analysis and model development made it the main programming language. In implementation process, libraries like NumPy, Pandas, Scikit-learn, LightGBM, and Matplotlib are used. Table 2 summarizes the software and hardware specifications utilized for model implementation

Table 2: The software and hardware specifications utilized for model implementation.

Component	Specification
Programming Language	Python 3.9+
Random-seed	Random_state = 42 for all models
Main Libraries	Numpy, Pandas, Scikit-learn, Lightgbm, Matplotlib, Openpyxl
Optional Visualization	Mpl_toolkits.axisartist (for Taylor diagrams)
IDE / Notebook	Jupyter Notebook / JupyterLab
Operating System	Windows 11
CPU	Quad-core Intel i5
RAM	16 GB
Storage	1–2 GB for excel outputs, plots, and models

3.2 Dataset information

This study to develop the predictive models for estimating the effective stiffness of RC under varying loading and geometric conditions utilizes a comprehensive dataset comprising 226 samples that created by Naderpour et al. (2019) [17]. This dataset was derived from the results of RC frame buildings that represents a unique RC column configuration, encompassing both material and geometric properties that significantly influence stiffness behavior. The axial load capacity (P), the percentage of reinforcing steel (p_t), and the column dimensions in the x-axis (D_x) and y-axis (D_y) directions are among the parameters that taken into consideration as input parameter in this study. Furthermore, the radius of gyration in the x-axis and y-axis (r_x and r_y) are regarded as crucial variables that described the column's stiffness that taken into consideration as output parameter in this study. Table 3, lists the essential variables for the effective stiffness estimation model, provides an overview of these parameters.

Table 3: Lists of utilized input parameters for the effective stiffness estimation.

Column	Description	Units
p_t (%)	Percentage of steel reinforcement area	%
D_x (m)	Column width (in X-direction)	meters
D_y (m)	Column depth (in Y-direction)	meters
P ($\times 10^3$ kN)	Axial load capacity	10^3 kilonewtons
r_x (m)	Radius of gyration about X-axis	meters
r_y (m)	Radius of gyration about Y-axis	meters

To guarantee that all input features were standardized to have a mean of zero and a standard deviation of one, the dataset is normalized during the preprocessing stage using the StandardScaler technique. Then, in order to train the model on a representative subset and save a portion for an independent assessment of generalization performance, the dataset is split into 80% for training and 20% for testing after normalization. On the training set, a 5-fold cross-validation scheme is used to reduce the chance of overfitting and increase the reliability of model assessment. In this process, five equal folds are created from the training data in this process that each iteration is used four folds for model training and one for validation. Then, the model performance is averaged across all five folds to provide a more robust and unbiased estimate of predictive accuracy, ensuring that the reported training results reflect consistent and generalizable performance across different data partitions.

4 Preliminaries

4.1 Decision tree

One popular and very interpretable technique for regression tasks is the DecisionTree algorithm. It functions by methodically dividing the dataset into more manageable, uniform groups. A decision path is represented by each branch of the tree, and an outcome or prediction is represented by each leaf. The algorithm chooses the most pertinent attribute at each stage to optimize prediction accuracy, with the root node representing the complete dataset prior to splitting. Measures such as the Gini Index, Entropy, and Information Gain, improve subset purity, and are the foundation of attribute selection.

Entropy: This is a measure in the theory of information that assesses and indicates the degree of uncertainty or disorder of a node. A split that reduces entropy is preferred, as lower entropy indicates smaller,

more ordered, and homogeneous subsets. The mathematical formulation for entropy is [18]:

$$\text{Entropy} = -\sum_{i=1}^{c} p_i \log_2 p_i \quad (1)$$

Here, p_i indicates the probability of occurrence of $i-$th class within the dataset, while c denotes the total number of classes.

Information Gain: The amount of reduced entropy in a dataset after this dataset is divided based on some feature clarification that is measured. A higher information gain is indicative of a more effective split, reflecting the attribute's ability to separate the data into distinct classes, which is represented as follows is [18]:

$$\text{Information Gain} = \text{Entropy}(S) - \sum_{i=1}^{n} \left(\frac{|S_i|}{|S|} \times \text{Entropy}(S_i)\right) \quad (2)$$

Here, S_i refers to the of $i-$th partition of S dataset, n is the number of attribute and $|S_i|$ and $|S|$ indicates the number of cases in the partition S_i and dataset S.

Gini Index: This metric considers the impurity of a node, and the lesser the Gini Index, the more homogeneous the node is, with more samples belonging predominantly to one class, and is defined as is [18]:

$$\text{Gini} = 1 - \sum_{i=1}^{c} p_i^2 \quad (3)$$

Here, p_i indicates the probability of occurrence of $i-$th class within the dataset.

These measures help to find the best splits at each node using them and yield higher predictive accuracy and, therefore, a more stable model. Fig. 2 shows the diagram of the DecisionTree model.

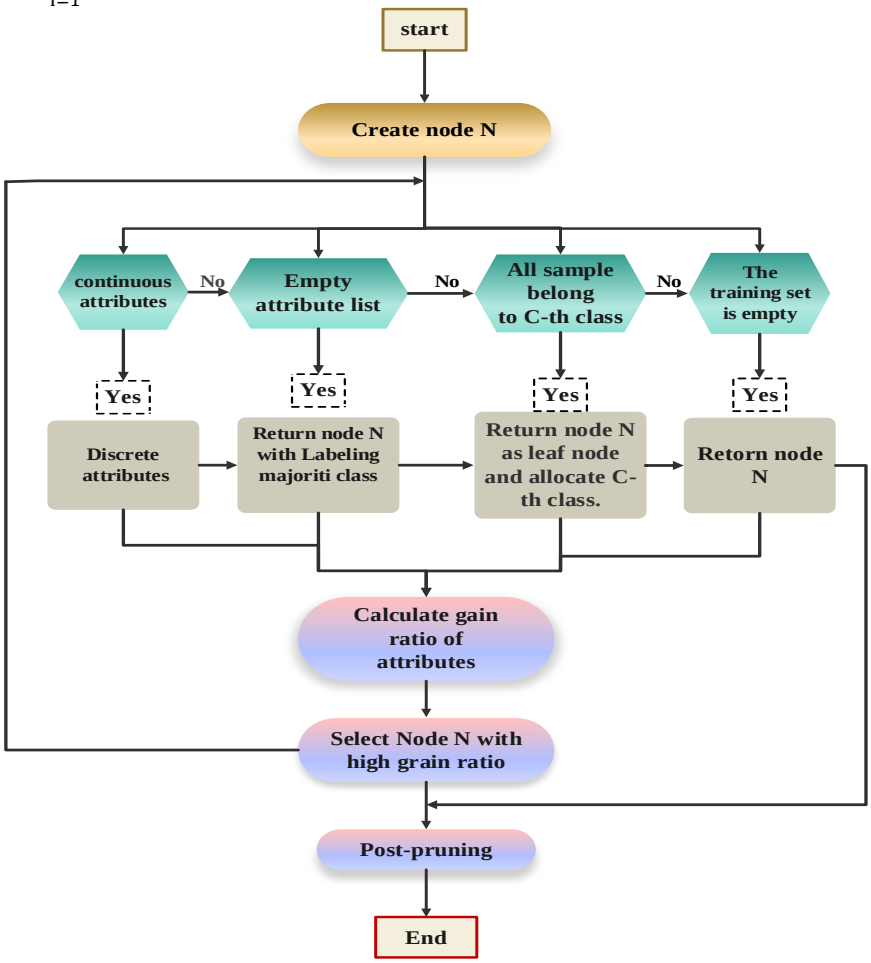


Figure 2: The diagram of the DecisionTree algorithm.

4.2 K-Nearest Neighbors

One of straightforward and effective non-parametric technique for regression problems is the KNN algorithm. It works on the premise that similar data points will probably produce similar results. An unlabeled instance is classified in KNN by locating the k nearest training samples in the feature space, which are typically found using distance metric like Euclidean distance. Then, the average value among these neighbors is assigned [19].

KNN regression predicts the result $\hat{y}$ for a new data x by voting on its KNN in the feature space upshot, given a dataset with n points labeled as $(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i)$, in which each x_i is a feature vector and y_i is the continuous output associated with it. In the KNN algorithm, the similarity between data points is quantified using a Euclidean distance metric. The Euclidean distance between two points x and x_i in the feature space is given by Eq. (4) [20]:

$$d(x, x_i) = \sqrt{(x - x_i)^T (x - x_i)} \quad (4)$$

Using the distance metric, KNN finds the K nearest points to the data point x in the data set. Then Eq. (5) uses the average of these K neighbors' outputs to determine the final prediction [20]:

$$\hat{y} = \frac{1}{k} \sum_{i \in k} y_i \quad (5)$$

Here, y_i refers to the prediction of each $K - th$ nearest points. Fig. 3 indicates the step-by-step flowchart of the KNN model.

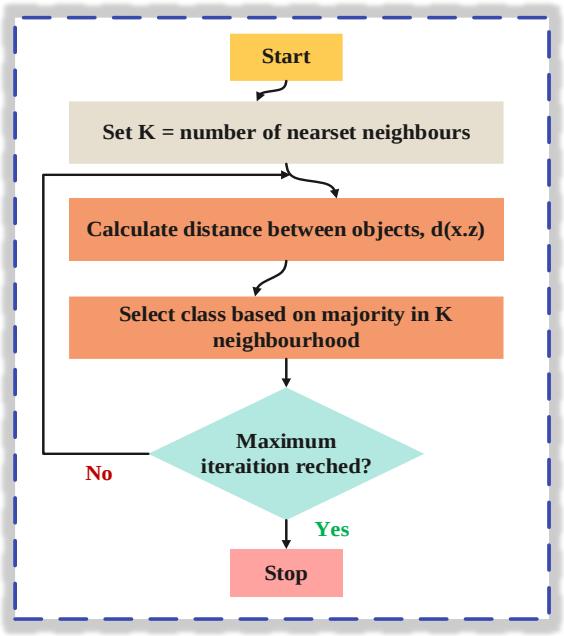


Figure 3: KNN method step-by-step flowchart.

4.3 Light gradient boosting

The LGB is one of the most efficient and fastest implementations of the gradient boosting framework, designed to run efficiently on very large-scale datasets. LGB is a boosting technique that iteratively creates an additive tree, integrating multiple weak learners to generate a strong learner. It primarily minimizes the loss function using gradient-based optimization [21].

It works efficiently for a dataset comprising n samples and m features displayed as (X_i, y_i): $i = 1 \ldots n$, where $X_i \in R^m$ and $y_i \in R$. In the architecture of the LGB model, there is an ensemble of decision trees each of which predicting the output as $\hat{y}_i$. The mechanism for updating the LGB model using boosting technique is given below by utilizing the following equations [22]:

$$\varphi(X_i)^{(t)} = \varphi(X_i)^{(t-1)} + \mu f_t(X_i) = \sum_{i=1}^{t} (L\left(y_i, \hat{y}_i^{(t-1)} + f_t(X_i)\right) + \Omega(f_t)) \quad (6)$$

Here, at each $t - th$ iteration, $\varphi(X_i)$ refers to the prediction function, that is updated using by adding a newly learned weak learner of $f_t(X_i)$. The parameter μ represents the step-size (learning rate) that controls how much each successive tree contributes to the overall model. The iterative optimization minimizes the loss function by incorporating a regularization term $\Omega(f_t)$ to penalize model complexity. This regularization term, defined in Eq. (7) [22]:

$$\Omega(f_k) = \gamma T_i + \frac{1}{2} \lambda \|w\|^2 \quad (7)$$

In addition, LGB has another key distinction: it uses a leaf-wise strategy for growing trees. In this strategy, the tree splits the leaf with the largest loss reduction, as it has the highest gradient. This results in deeper, more unbalanced trees that are often more accurate. The leaf-wise growth is also significantly faster and less memory-intensive compared to the level-wise strategy, where nodes at each tree level are expanded with equal priority. Fig. 4 displays how Light GBM works.

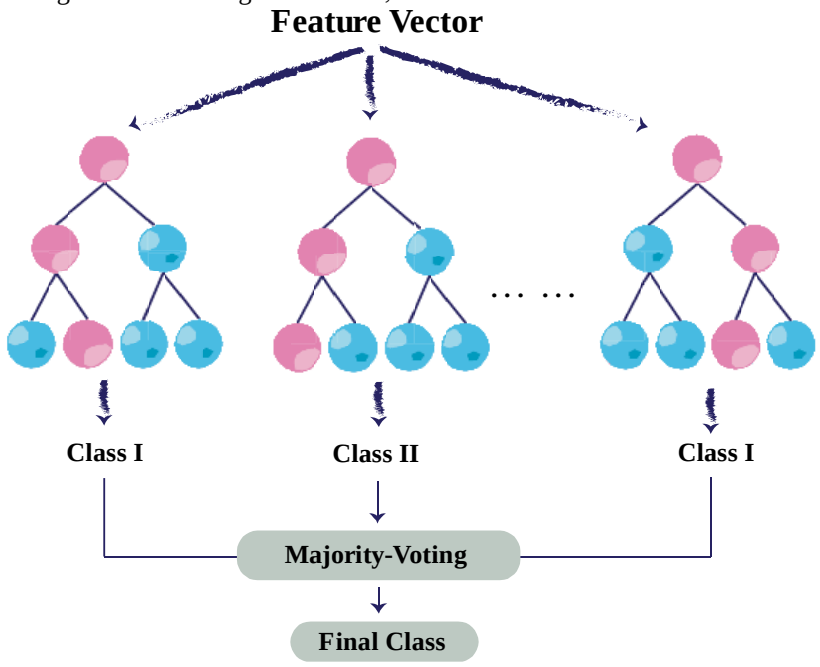


Figure 4: Procedures of the Light GBM model.

4.4 Stacking ensemble

A stacking ensemble is a sophisticated ensemble learning approach that uses a meta-model to aggregate predictions from many base models. The basic concept is to train a series of base models first and then feed the meta-model with their predictions. Stated differently, the meta-model is trained using the basic models' predictions and learns how to efficiently integrate them to enhance overall outcomes. The accumulation model process can be seen in Fig. 5.

It first initiates by training different base models on the same training input data (x), and generating the predictions set $(\hat{y}_1(x), \hat{y}_2(x), \hat{y}_3(x), \ldots \hat{y}_m(x))$ that represents the prediction of each model. Therefore, for the Stacking model on input data (x), the predictions of all base models are calculated to form the meta-feature vector of F(x). The meta-model takes this input vector in order to make the final prediction as follows [23], [24]:

$$F(x) = [\hat{y}_1(x), \hat{y}_2(x), \hat{y}_3(x), \ldots \hat{y}_m(x)] \quad (8)$$

$$\hat{y} = \hat{y}_{meta}\,(F(x)) \quad (9)$$

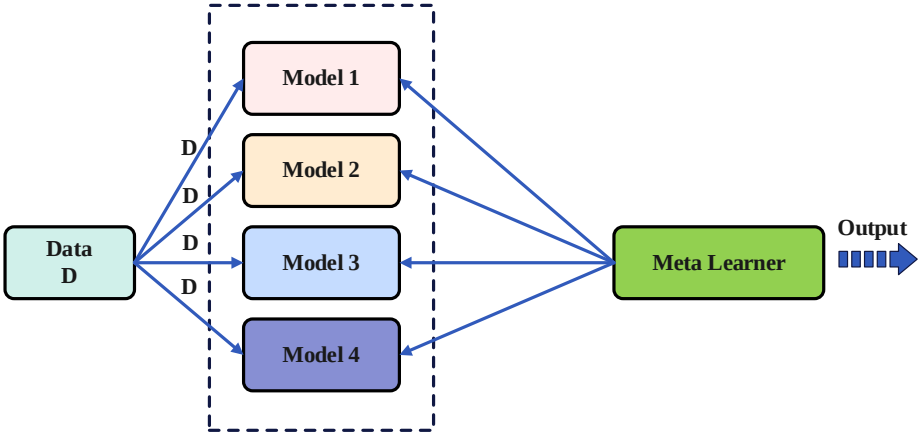


Figure 5: The process of the Stacking model.

4.5 Averaging ensemble

Averaging ensemble technique increases the overall prediction accuracy and resilience by averaging the outputs of several models. By combining the capabilities of many models, this kind of ensemble approach aims to produce forecasts that are more accurate and dependable than any one model could provide alone [25], [26]. The process of the Averaging model is presented in Fig. 6. In the case of regression, the average of the model predictions, the Averaging ensemble makes prediction as below [27]:

$$\hat{y}_{Averaging} = \sum_{i=1}^{n} \frac{\hat{y}_i}{n} \quad (10)$$

In this equation, the $\hat{y}_{Averaging}$ illustrates the output of an Averaging model, $\hat{y}_i$ indicative of the prediction of each $i-th$ model. The variable n denotes the total number of models that are included in the combination method.

Some models may vary in enhanced capabilities, so to achieve the best results, it is necessary to adjust the effect of each model in the set based on its performance. The weighted average of the ensemble prediction, or F(x), is used to do this that mathematically is represented as below [28]:

$$\hat{y}\,(x) = \sum_{i=1}^{n} \omega_i \times \hat{y}_i(x) \quad (11)$$

Here, $\hat{y}_i(x)$ denotes the prediction of the $i-th$ model within the ensemble and ω_i indicates the corresponding weights assigned to the $i-th$ model.

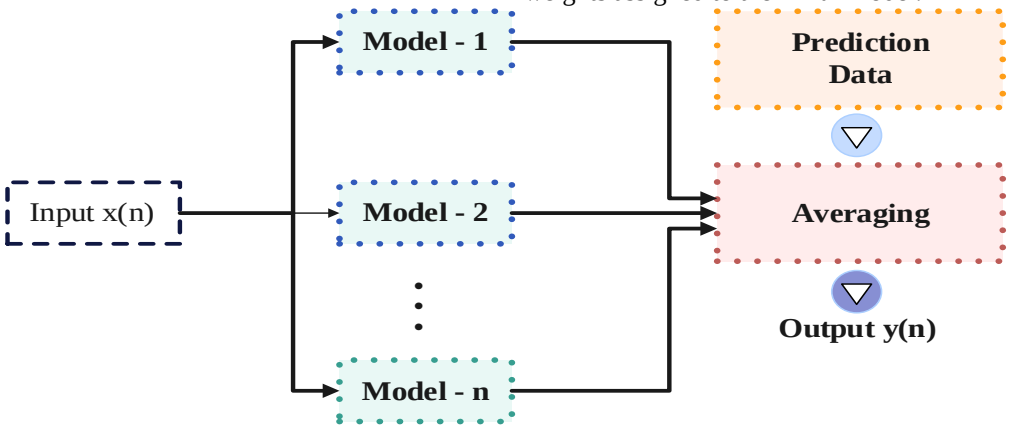


Figure 6: The process of the Averaging model.

4.6 Voting ensemble

A voting ensemble model is another sophisticated ML technique that by combining the results of several base models intends to improve predictive performance. To get more accurate and consistent results, this approach combines the predictions of multiple models, each trained using a different algorithm, to help reduce overfitting and enhance generalization on unseen data by utilizing the complementary strengths of individual models. Hard voting and soft voting are the two primary methods of voting [29], [30]. The voting model is presented in Fig. 7.

Soft voting considers the predicted probabilities of each model and averages them to produce the final prediction as below [31]:

$$\hat{y}_{soft} = \arg\max_{c \in C} \frac{1}{n} \sum_{i=1}^{n} \hat{y}_i \quad (12)$$

Here, the $\hat{y}_{soft}$ illustrates the output of a voting model, and $\hat{y}_i$ indicative of the prediction of each $i-th$ model. The variable n denotes the total number of models that are included in the combination method and C refers to the set of classes.

In hard voting, the class label that the majority of the base models predict is chosen as the final output as below [31]:

$$\hat{y}_{hard} = \arg\max_{c \in C} \sum_{i=1}^{n} (\hat{y}_i = c) \quad (13)$$

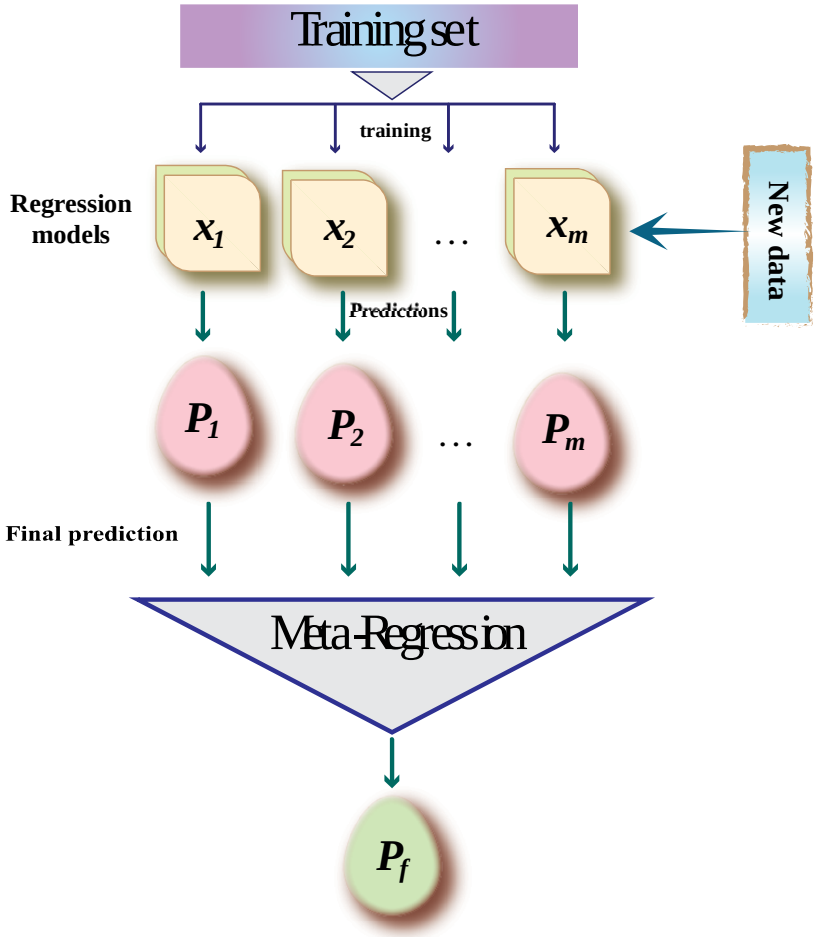


Figure 7: The flowchart of the voting model.

4.7 Evaluation metrics

In this section, the optimization algorithms are compared through statistical evaluation indices. The evaluation indices are measures widely used to assess the operation of the models designed to forecast the numerical variables. Various evaluation indices are available; in this study, the selected indices include R^2, $RMSE$, MAE, STD, VAF, FI, $MDAPE$, and $NMSE$. The equations of the preceding indices are presented in Fig. 8.

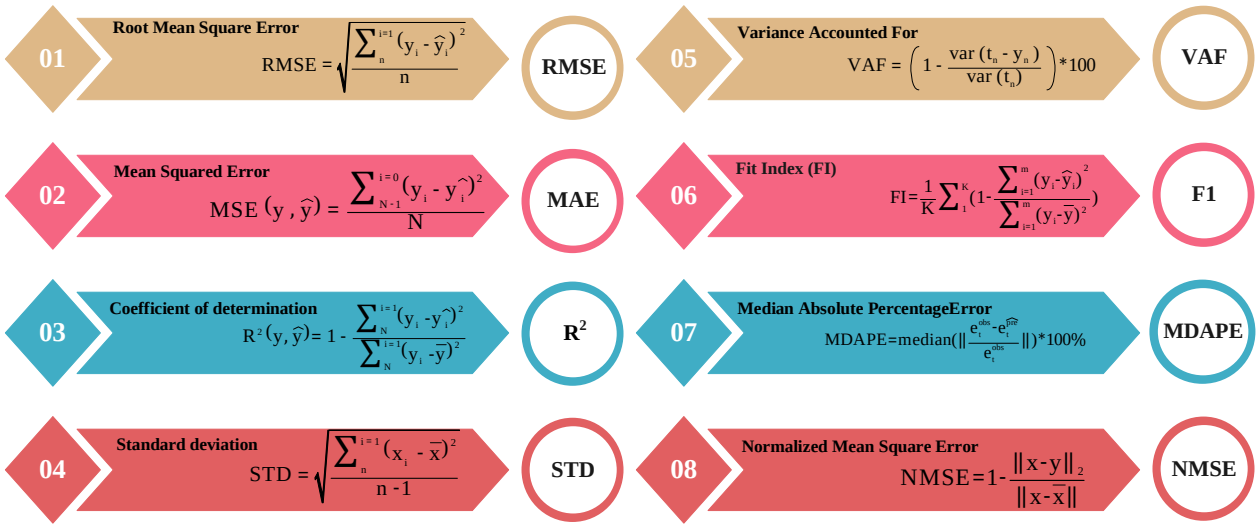


Figure 8: Evaluation equation.

5 Results and discussion

5.1 Dataset analysis

Fig. 9 is a correlation heatmap depicting Pearson correlation coefficients between different variables in a data set. The direction and intensity of linear connections between variables may be determined with the use of a heat map. This heatmap contains several variables that include p_t (%), D_x (m), D_y (m), P (×10^3kN), Avg. D_x (m) and D_y (m) have a very strong positive correlation (0.99). This implies that the displacements in the X and Y directions are approximately proportional. P_t (%) and Avg have a strong positive correlation (0.97). This shows that the reinforcement percentage is highly correlated with the average value, which can mean that the reinforcement percentage strongly affects the overall performance of the structure. P, D_x, and D_y have a moderate positive correlation (0.71 and 0.72, respectively). A higher axial load is somewhat associated with larger displacements in both the X and Y directions. P_t (%) and P have a slight positive correlation (0.2), indicating that axial load may have little effect on percent reinforcement. This correlation matrix gives insight into how various structural or input parameters relate to one another, which may also help determine the consequent analysis or modeling tasks, especially with regard to ML and structural analyses.

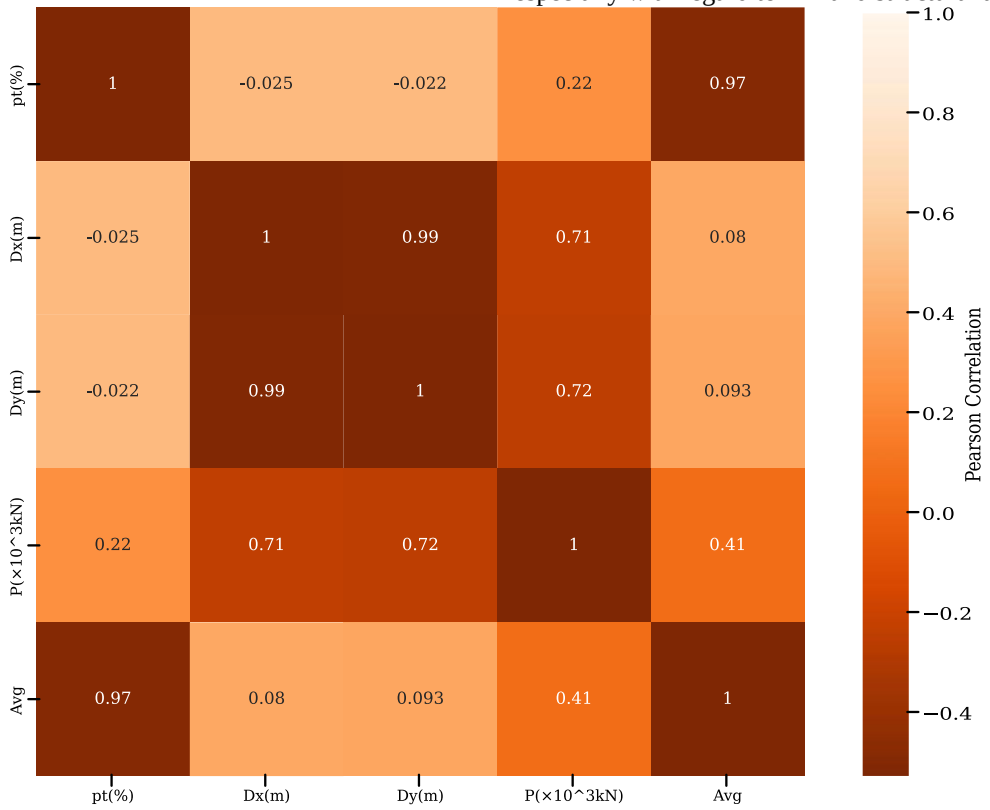


Figure 9: A heatmap of the correlation between variables in a data set.

Fig. 10 is a plot comparing pairs of different input features, with the output variable labeled "Avg" (average), represented on a color scale from light blue (0.4) to dark brown (0.8). Pair plots allow visualization of relationships between multiple variables and help observe pairwise correlations. The variables included in the design are p_t (%), D_x(m), D_y(m), and P (×10^3 kN). The diagonal plots show kernel density estimates (KDEs) or histograms for each individual variable, illustrating their distribution. Off-diagonal scatter plots display pairwise relationships between different variables. As seen in the relationship between "D_x(m)" and "p_t (%)" or between "D_y(m)" and "P (×10^3 kN)". There are some clear patterns, such as the correlation between "p_t (%)" and "D_x(m)", or between "D_y(m)" and "P (×10^3 kN)". These scatterplots show how well the variables are related, and how the color (indicating "Avg") changes according to those relationships. This pair diagram in general helps analyze the dependencies and trends between input variables and how they are related to the output variable "Avg".

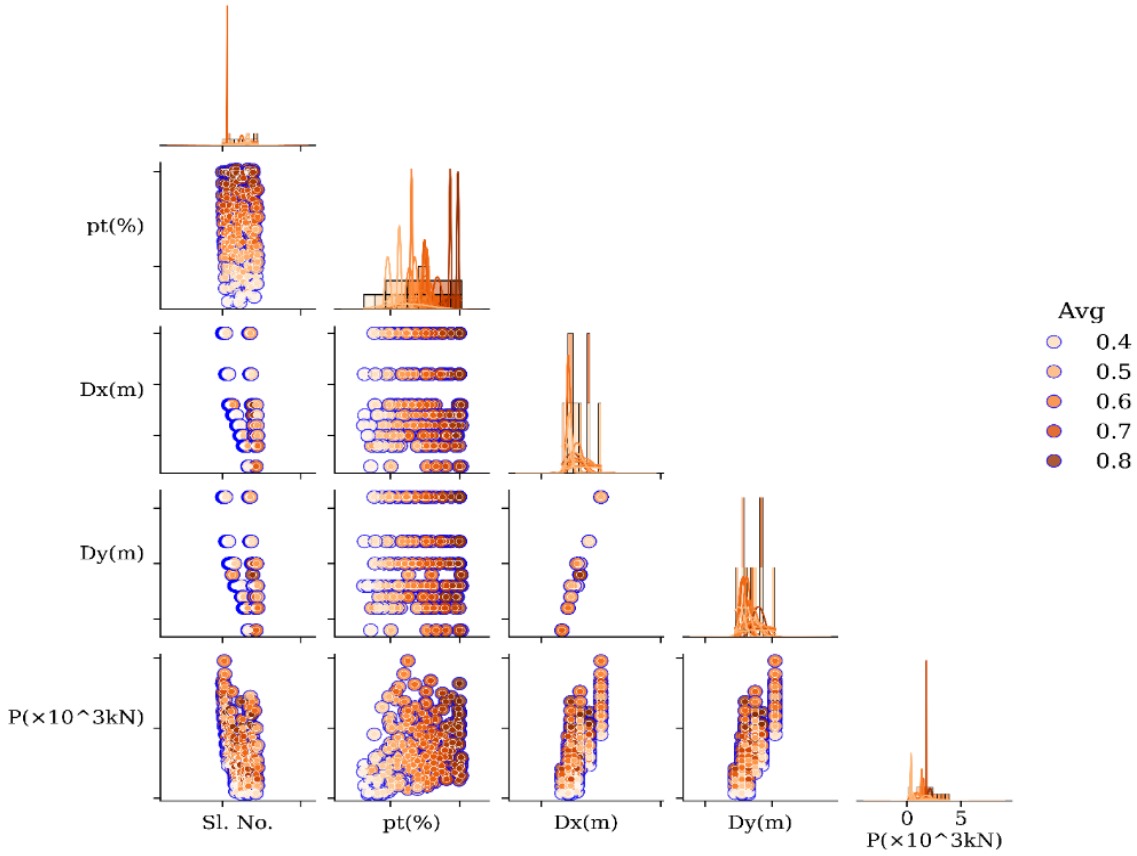


Figure 10: Pair plot with "Avg" (average) as output.

5.2 Hyperparameters tuning

Table 4 indicates the hyperparameter search ranges used to fine-tune each model. To improve prediction accuracy and avoid overfitting, parameters governing model complexity, learning rate, and regularization were methodically changed. To achieve balanced performance, robustness, and generalization across all predictive tasks, both individual models (DecisionTree, KNN and LGB) and ensemble approaches (Stacking and Voting) were optimized using specified parameter ranges.

Table 4: The hyperparameter configuration for each utilized model.

Model	Hyperparameter	Range / Values
DecisionTree	max_depth	3 – 30
	min_samples_split	2 – 20
	min_samples_leaf	1 – 10
	criterion	"squared_error", "friedman_mse", "absolute_error"
KNN	n_neighbors	3 – 15
	weights	"uniform", "distance"
	p	1 – 2
LGB	num_leaves	20 – 200
	max_depth	-1 – 20
	learning_rate	0.005 – 0.3
	n_estimators	100 – 1000

	subsample	0.6 – 1.0
	colsample_bytree	0.6 – 1.0
	reg_lambda	0 – 1
	reg_alpha	0 – 1
Stacking	cv	3 – 10
	final_estimator	Linear-Regression (), LGB ()
Voting	weights	[1, 1, 1] or tuned per model

5.3 Performance evaluation

The effectiveness of several ML models for determining the RC column's effective stiffness factors (r_x and r_y) is compiled in Tables 5 and 6. According to that, ensemble-based models, especially ensemble models show better predictive power than individual learners across both parameters. The Stacking and Averaging models obtains the highest R^2 values 0.9708 for r_x and 0.9840 for r_y on the test sets, showing low prediction error and strong generalization. The ensembles robustness in capturing nonlinear relationships among features was further confirmed by its consistently low RMSEs of 0.0182 and 0.0169 values. The Stacking and Averaging ensemble models' superior performance can be ascribed to their innate ability to incorporate complementary predictive patterns from various base learners, thereby mitigating bias and variance. In contrast the LGB algorithm outperformes traditional learners like DecisionTree and KNN in terms of accuracy, achieving R^2 values of 0.9668 and 0.9525 on the test sets for r_x and r_y. On the other hand, the DecisionTree model shows a perfect fit on the training data with train R^2 of 0.9999 for both, while in the test with R^2 0.9255 and 0.8680 for r_x and r_y shows a decline in generalization. The primary cause of this performance in the DecisionTree model is its unbridled expansion and high structural adaptability, which enable it to fully retain training data instead of picking up generalizable patterns. Near-perfect training accuracy but a noticeable drop in test performance result from the tree capturing noise and small fluctuations in the training data when it grows too deep without pruning or regularization. This suggests that poor generalization to unknown samples results from the model's complexity surpassing that of the underlying data structure. This suggests that poor generalization to unknown samples results from the model's complexity surpassing that of the underlying data structure. Moreover, The KNN model's sensitivity to data noise and local irregularities is demonstrated by its moderate accuracy but higher error dispersion (RMSEs of 0.0325 for r_x and 0.0502 for r_y).

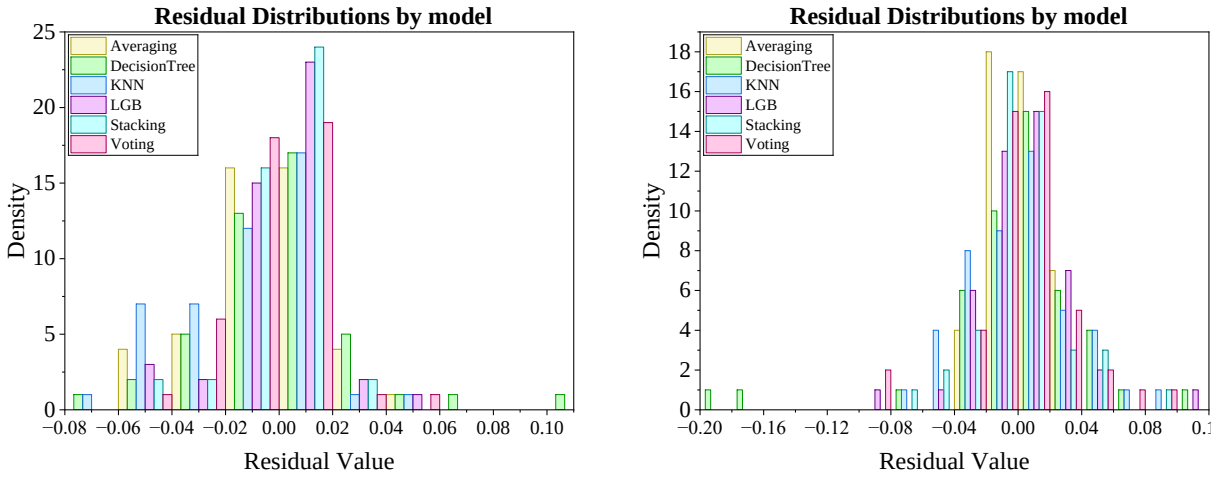
Table 5: The finding related to estimation of effective stiffness factor (r_x) of RC column by ML models.

Model	Set	RMSE	MAE	R^2	NMSE	MDAPE	STD_dev	VAF	FI
Averaging	Train	0.0222	0.0158	0.9680	0.0320	2.2987	0.0222	96.7955	0.9596
	Test	0.0222	0.0172	0.9561	0.0439	2.6409	0.0219	95.7374	0.9581
DecisionTree	Train	0.0000	0.0000	0.9999	0.0000	0.0000	0.0000	99.9999	0.9999
	Test	0.0290	0.0196	0.9255	0.0745	1.8803	0.0290	92.5558	0.9455
KNN	Train	0.0272	0.0207	0.9517	0.0483	3.0580	0.0272	95.1815	0.9504
	Test	0.0325	0.0253	0.9064	0.0936	4.4878	0.0321	90.8643	0.9389
LGB	Train	0.0166	0.0119	0.9819	0.0181	1.5797	0.0166	98.1945	0.9696
	Test	0.0193	0.0141	0.9668	0.0332	2.1360	0.0193	96.6848	0.9636
Stacking	Train	0.0152	0.0111	0.9849	0.0151	1.6039	0.0152	98.4897	0.9722
	Test	0.0182	0.0132	0.9708	0.0292	2.1026	0.0182	97.0755	0.9658
Voting	Train	0.0126	0.0096	0.9897	0.0103	1.4049	0.0126	98.9705	0.9771
	Test	0.0200	0.0147	0.9645	0.0355	2.2456	0.0200	96.4629	0.9623

Table 6: The finding related to estimation of effective stiffness factor (r_y) of RC column by ML models.

Model	Set	RMSE	MAE	R^2	NMSE	MDAPE	STD_dev	VAF	FI
Averaging	Train	0.0264	0.0170	0.9692	0.0308	1.5525	0.0264	96.9249	0.9636
	Test	0.0169	0.0141	0.9840	0.0160	2.0019	0.0167	98.4424	0.9765
DecisionTree	Train	0.0000	0.0000	0.9999	0.0000	0.0000	0.0000	99.9999	0.9999
	Test	0.0486	0.0311	0.8680	0.1320	2.3475	0.0486	86.7996	0.9324
KNN	Train	0.0391	0.0294	0.9327	0.0673	2.9637	0.0385	93.4626	0.9461
	Test	0.0502	0.0372	0.8594	0.1406	4.0270	0.0492	86.4991	0.9302
LGB	Train	0.0213	0.0151	0.9801	0.0199	1.4706	0.0213	98.0100	0.9707
	Test	0.0292	0.0200	0.9525	0.0475	2.0907	0.0290	95.3139	0.9594
Stacking	Train	0.0199	0.0140	0.9826	0.0174	1.4744	0.0198	98.2719	0.9726
	Test	0.0288	0.0198	0.9536	0.0464	1.9463	0.0287	95.3907	0.9599
Voting	Train	0.0178	0.0131	0.9861	0.0139	1.4090	0.0176	98.6300	0.9755
	Test	0.0345	0.0239	0.9334	0.0666	2.0406	0.0345	93.3625	0.9520

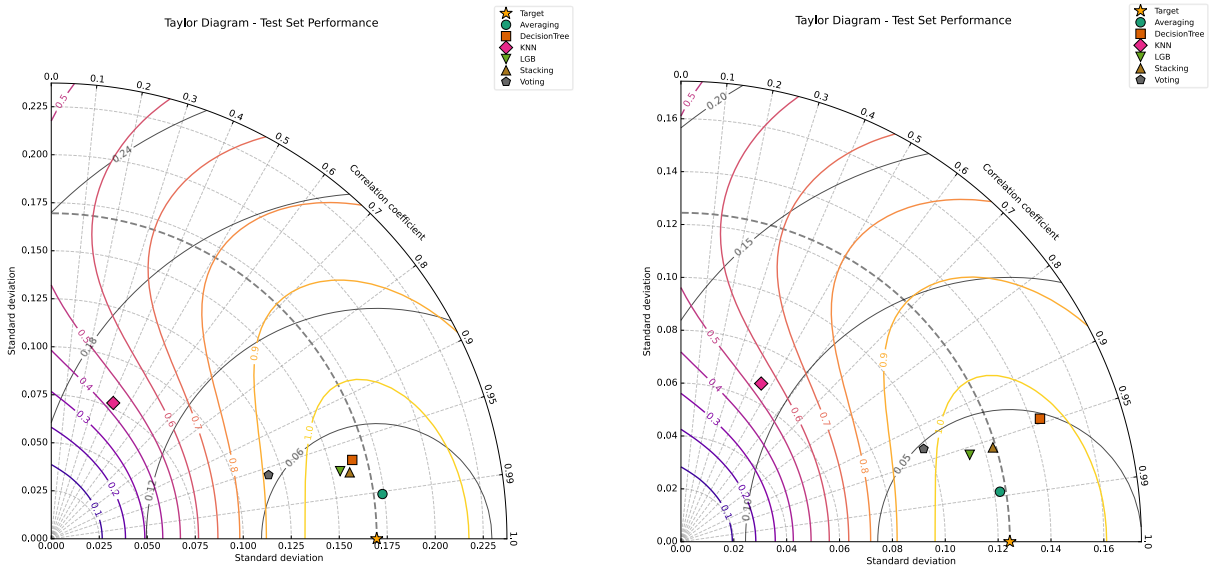
The residual distribution plots for estimating the effective stiffness factors (r_x and r_y) of RC columns using a variety of ML models are shown in Fig. 11. A properly calibrated model should ideally yield residuals that are symmetrically distributed around zero with little dispersion, suggesting that the model consistently follows the target variable. Ensemble-based techniques, such as stacking, voting, and averaging, show compact, bell-shaped residual patterns centered around zero, indicating strong predictive consistency and low bias across samples, as shown in Fig. 11 (a) for r_x and Fig. 11 (b) for r_y. This distributional behavior demonstrates how ensemble frameworks can better integrate individual learners' strengths while addressing their weaknesses, improving generalization ability and resilience to data noise. On the other hand, DecisionTree and KNN models, show more variance and sensitivity to local fluctuations in the data due to their wider and somewhat asymmetric residual distributions. Although the residual spread of the LGB model is smaller than that of these individual learners, it still shows slight departures from perfect normality, indicating limited bias in extreme situations. Out of all the models that were evaluated, Stacking and Averaging models obtained the most concentrated residual density. This suggests that they effectively capture the intricate nonlinear interactions between input features, leading to the lowest overall prediction uncertainty. Their exceptional ability to optimize the bias-variance trade-off is further demonstrated by the ensemble model residuals' consistent alignment near zero.



(a) r_x (b) r_y

Figure 11: Residual distributions by models in the estimation of effective stiffness factors of RC column.

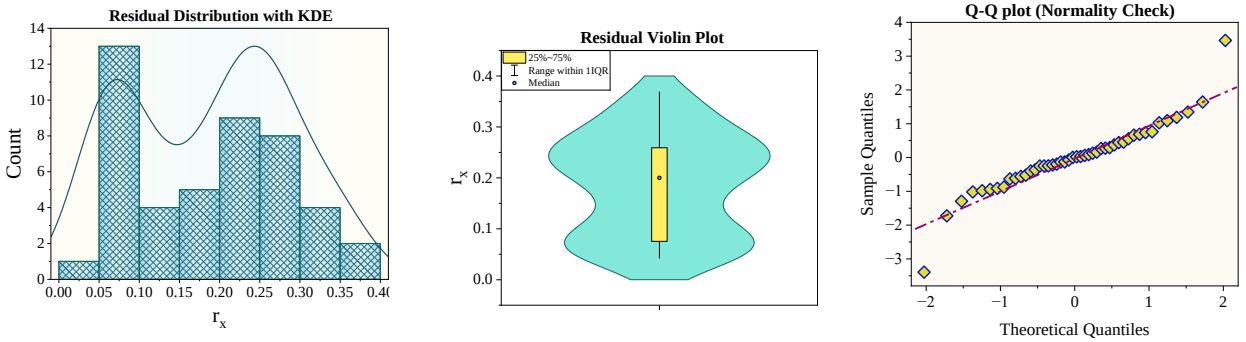
A thorough comparison of the developed ML models' predictive performance against the observed data is shown in Fig. 12 via a Taylor plot. The correlation coefficient, RMSE, and standard deviation of each model are used to illustrate its performance. Strong agreement with the measured values and high predictive reliability are indicated by models with low RMSE and high correlation whose markers are placed near the reference (observed) point. On the other hand, models that are farther away from the reference point exhibit weaker correlation and larger deviations, which indicate decreased accuracy. As shown, the Stacking and Averaging ensemble models are the ones that are closest to the reference point, demonstrating their high consistency, low prediction error, and outstanding ability to replicate the variability of the observed stiffness data.



(a) r_x (b) r_y

Figure 12: Taylor digram for developed models in the estimation of effective stiffness factors of RC column.

Moreover, to further examine the residual distribution and reliability of the ensemble models, Fig. 13 presents the residual histograms, violin plots, Kernel Density Estimation (KDE) curves, and Q–Q plots for the Stacking model in predicting r_x and the Averaging model in predicting r_y. These visual analyses provide insight into the error behavior and normality of the residuals. The residuals' distribution and conformity to the normal distribution are shown by the plots, which offer information on the bias and accuracy of the staking and averaging model. Based on KDE plot, the r_y residuals are more symmetrically centered around zero, and the residuals for r_x with slightly right-skewed distribution also exhibit concentration around zero indicating a balanced prediction behavior by staking and averaging models. The violin plots, in which the dot indicates the median, the central black line shows the range within 1.5×IQR, and the yellow box represents the interquartile range (25th–75th percentile), further illustrate the distribution and concentration of residuals. In both situations, the residuals' concentrated and narrow distribution around the median attests to the model's impartiality and resilience. Moreover, based on the Q–Q plots, the comparing sample quantiles with the theoretical quantiles indicates that the data points in both plots are nearly parallel to the diagonal reference line, suggesting that the residuals for staking and averaging have a distribution that is roughly normal. The extreme tails show only slight deviations, which are normal for empirical data and do not substantially deviate from the normalcy assumption. All things considered, the residual analysis confirms the ensemble model' generalizability and dependability for precise stiffness ratios estimation in RC columns.



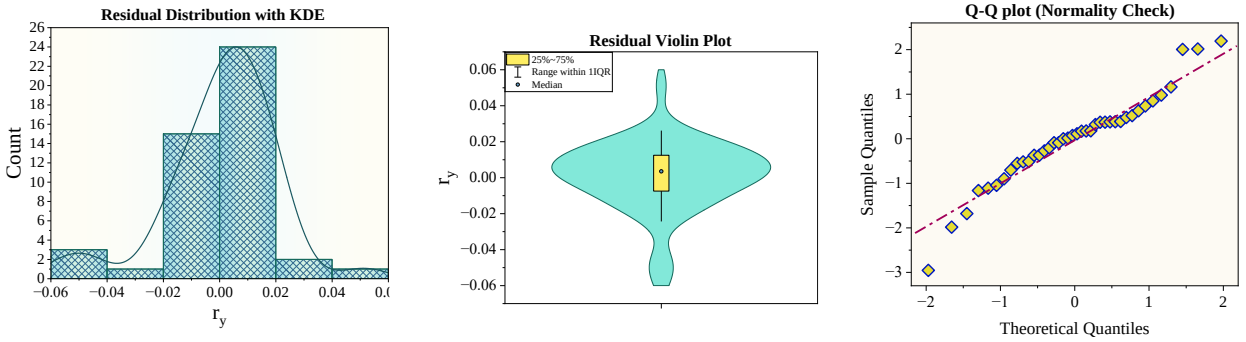
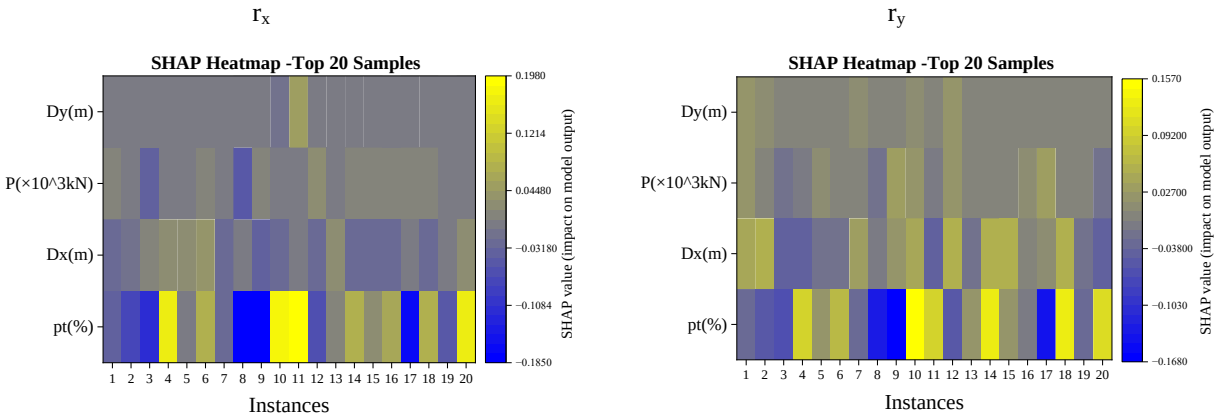


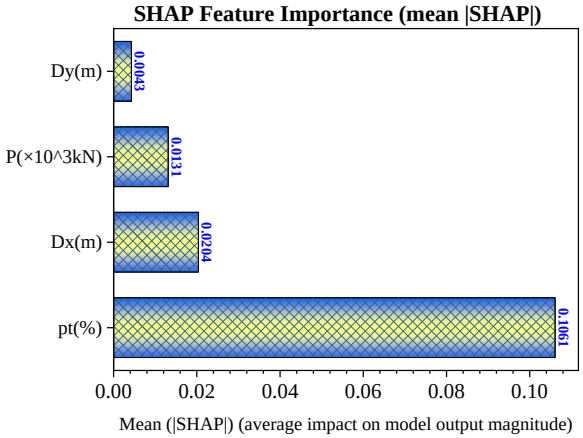
Figure 13: The residual analysis using violin, KDE and Q-Q plots for the Stacking model in predicting r_x and the Averaging model in predicting r_y.

5.4 SHAP-based interpretability analysis

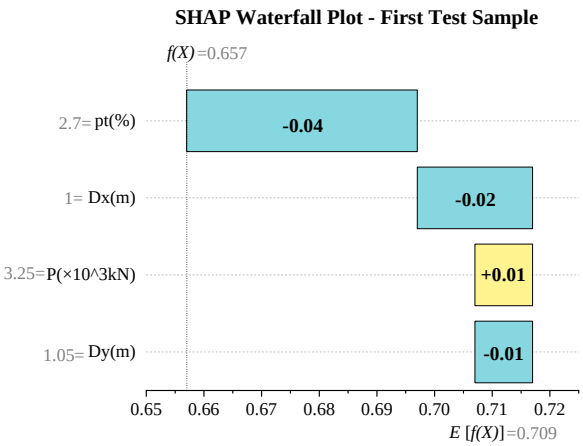
Figs 14 illustrate the SHAP-based plots for heatmap, feature importance, waterfall and values for the effective stiffness ratios r_x and r_y. These plots provide a clear interpretability framework for understanding how individual features influence the model's output predictions. The SHAP heatmaps in Fig. 14 (a) for the 20 samples, indicates the direction and strength of each feature's influence on the model output. In this plot the negative values (blue regions) suggest a decreasing effect, positive SHAP values (yellow regions) indicate that the feature increases the predicted stiffness ratio. The heatmaps make it clear that p_t is the most influential parameter, indicating that the percentage of steel reinforcement have a significant and steady impact on both r_x and r_y. The SHAP feature importance plot in Fig. 14 (b), which quantify the importances score of features across all samples. As depicted in this plot, for both r_x and r_y, the p_t parameter that indicates the reinforcement ratio with the highest importance score of 0.11 and 0.8 emerges as the most influential parameter. Moreover, the SHAP-based waterfall and values in Fig. 14 (c and d) confirm these findings and indicates that the reinforcement ratio is the most influential parameter in determining the effective stiffness. This dominance implies that changes in the percentage of reinforcement have a significant impact on the stiffness of RC columns. While the column width has a comparatively smaller impact, and the column depth and axial load exhibit moderate contributions. This ranking is constant in both directions, indicating that geometry and reinforcement have a greater influence on stiffness behavior than loading circumstances.



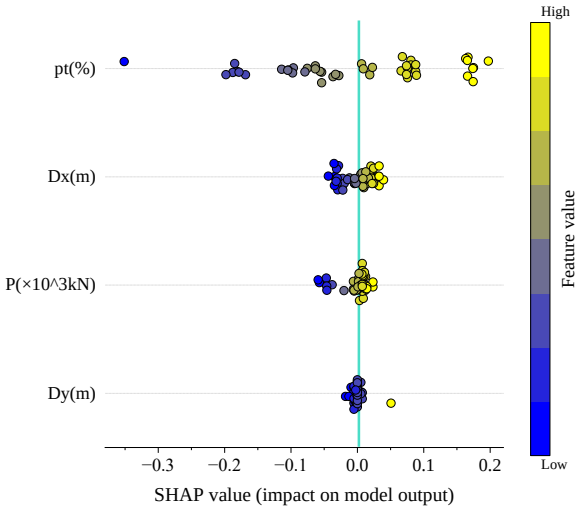
(a) SHAP Heatmap



(b) SHAP Feature Importance



(c) SHAP waterfall plots



(d) SHAP value

Figure 14: The interpretability analysis of developed Staking in predicting r_x and Averaging models in predicting r_y using SHAP-based plots.

5.5 Model's performance discussion

Our analysis showed that the Stacking and Averaging ensemble models had the highest predictive accuracy with test-set R^2 values of 0.9708 for r_x and 0.9840 for r_y, as well as correspondingly low RMSE values. This suggests that there are little generalization gap and strong predictive consistency between the training and test datasets. Indeed, the use of GS-based optimization with 5-fold cross-validation result high generalization ability for these models, as evidenced by the small (<2%) difference between training and testing results. When compared with related studies in Table 1, the obtained results demonstrate

superior or at least comparable accuracy. For instance, previous works reported R^2values of 0.868 for ANN [11], 0.931 for M5-Tree [12], 0.977-0.923 for CatBoost [13], 0.90–0.97 for GPR [14], 0.906 for ANN [15] and up to 0.960 for DecisionTree [16] in estimation of various parameters of RC columns. Thus, the Stacking and Averaging ensembles outperformed most existing ML-based frameworks in predicting RC structural parameters. The Stacking and Averaging ensemble models' superior performance can be ascribed to their innate ability to incorporate complementary predictive patterns from various base learners, thereby mitigating bias and variance. By calculating the mean of the predictions made by several separate learners the Averaging ensemble model aggregates their outputs which evens out individual model fluctuations and present a more stable and universal response results from this aggregation. The averaging process improves robustness against data noise and outliers and reduces the risk of overfitting associated with any one model because each learner captures distinct aspects of the input–output relationship. The Stacking ensemble, on the other hand, uses a more advanced meta-learning framework, where a higher-level meta-learner uses the predictions from multiple base models as new input features. Based on their residual errors, this meta-learner, typically a regression or boosting algorithm, learns how to combine and weight each base learner's strengths in the best possible way. Consequently, stacking is able to model highly nonlinear relationships that individual learners might overlook by learning inter-model dependencies and error correction patterns in addition to aggregating predictions.

Despite these promising outcomes, several limitations are acknowledged. The developed Staking and Averaging models were trained on a static dataset and have not been implemented or evaluated in real-time structural monitoring environments and were trained on a static dataset. Furthermore, even though feature importance analyses and SHAP offer some interpretability, more effort is required to improve explainability using sophisticated model transparency frameworks. To increase deployment viability and interpretive reliability, future studies might concentrate on combining explainable AI modules, real-time data streams, and hybrid physics–ML models.

6 Conclusions

This work presented a flexible and high-performance ensemble framework based on the Stacking and Averaging techniques to improve the prediction of two crucial structural parameters that are directly related to the effective stiffness factors (r_x and r_y) of reinforced concrete structures. The suggested method combines several base learners, utilizing their complementary advantages to better capture intricate nonlinear dependencies and multi-factor interactions, lessen sensitivity to data noise, and increase predictive accuracy. These ensemble models were methodically contrasted with other algorithms, such as Voting, KNN, LGB, and Decision-Tree for thorough validation and benchmarking. Furthermore, the GS-based optimization used in the ML processes enabled further improvements through hyperparameter tuning and leading to more accurate predictions and greater efficiency. The performance evaluation showed that the Stacking and Averaging ensemble models had the highest predictive accuracy with test-set R^2 values of 0.9708 for r_x and 0.9840 for r_y, as well as correspondingly low RMSE values. These optimized ensemble models could help engineers to design safer, more cost-effective, and environmentally friendly structures for RC structure. They These models significantly reduce the time required for structural design and analysis by automating the stiffness prediction process. Engineers can rapidly evaluate multiple design configurations without the need for time-consuming iterative calculations, thus simplifying the RC column design process.

Moreover, using SHAP-based analysis, the models can handle a wide range of input variables and provide insight into which parameters (e.g., reinforcement ratio, concrete dimension, axial load) most affect the stiffness of RC columns. The outcomes make it clear that the percentage of steel reinforcement is the most influential parameter and have a significant and steady contribution on both r_x and r_y values. This understanding supports better decision-making during the design phase. This allows the development of lightweight, cost-effective, and structurally sound designs.

Abbreviations

RC	Reinforced concrete	LGB	Light gradient boosting
KNN	K-Nearest Neighbors	$\hat{y}_m(x)$	The prediction of each $m-th$
GS	Grid Search	$\hat{y}_i$	Variable of the meta-model
SHAP	Shapley Additive Explanations	$\hat{y}_{Averaging}$	The output of an Averaging model
ANN	Artificial Neural Network	$\hat{y}$	Final prediction of model
GPR	Gaussian Process Regression	$f_i(x)$	Prediction of the $i-th$ model
ML	Machine Learning	ω_i	Corresponding weights assigned to the $i-th$ model.
$\Omega(f_t)$	Complexity		

Acknowledgements

We would like to take this opportunity to acknowledge that there are no individuals or organizations that require acknowledgment for their contributions to this work.

Authors' contributions

Xine Yan conducted analysis

Juanjuan Wang performed Data collection also

Yetao Cong carried out simulation

Zhanhao Zhang evaluate the first draft of the manuscript, editing and writing.

Funding

This work was supported by 2023 Youth Innovation Team Scientific Research project of Shaanxi Provincial Education Department. (Project No: 23JP086)

Xi'an Key Laboratory of Track Subgrade Bed Disease Monitoring and Prevention.

The Youth Innovation Team of Shaanxi Universities.

Ethical Approval

The research paper has received ethical approval from the institutional review board, ensuring the protection of participants' rights and compliance with the relevant ethical guidelines.

References

[1] S. Das, I. Mansouri, S. Choudhury, A. H. Gandomi, and J. W. Hu, "A prediction model for the calculation of effective stiffness ratios of reinforced concrete columns," *Materials*, vol. 14, no. 7, p. 1792, 2021. https://doi.org/10.3390/ma14071792

[2] B. Zhao, F. Taucer, and T. Rossetto, "Field investigation on the performance of building structures during the 12 May 2008 Wenchuan earthquake in China," *Eng Struct*, vol. 31, no. 8, pp. 1707–1723, 2009. https://doi.org/10.1016/j.engstruct.2009.02.039

[3] M. Vafaei, M. Baniahmadi, and S. C. Alih, "The relative importance of strong column-weak beam design concept in the single-story RC frames," *Eng Struct*, vol. 185, pp. 159–170, 2019. https://doi.org/10.1016/j.engstruct.2019.01.126

[4] C. Shao, W. Wei, G. Xu, Q. Qi, and C. Wang, "Assessment on effective stiffness of RC hollow columns based upon semi-static experiment," in *Structures*, Elsevier, 2023, pp. 882–894. https://doi.org/10.1016/j.istruc.2023.04.071

[5] C. G. Nechevska and A. Roshi, "Rehabilitation of RC buildings in seismically active regions using traditional and innovative materials," *Građevinski materijali i konstrukcije*, vol. 62, no. 3, pp. 19–30, 2019.

[6] J. Bell, "What is machine learning?," *Machine learning and the city: applications in architecture and urban design*, pp. 207–216, 2022.

[7] D.-C. Feng, W.-J. Wang, S. Mangalathu, G. Hu, and T. Wu, "Implementing ensemble learning methods to predict the shear strength of RC deep beams with/without web reinforcements," *Eng Struct*, vol. 235, p. 111979, 2021. https://doi.org/10.1016/j.engstruct.2021.111979

[8] Y. Li, X. Lu, H. Guan, and L. Ye, "An improved tie force method for progressive collapse resistance design of reinforced concrete frame structures," *Eng Struct*, vol. 33, no. 10, pp. 2931–2942, 2011. https://doi.org/10.1016/j.engstruct.2011.06.017

[9] X. Lu, L. Xie, H. Guan, Y. Huang, and X. Lu, "A shear wall element for nonlinear seismic analysis of super-tall buildings using OpenSees," *Finite Elements in Analysis and Design*, vol. 98, pp. 14–25, 2015. https://doi.org/10.1016/j.finel.2015.01.006

[10] W. Fan *et al.*, "Machine learning applied to the design and inspection of reinforced concrete bridges: Resilient methods and emerging applications," in *Structures*, Elsevier, 2021, pp. 3954–3963. https://doi.org/10.1016/j.istruc.2021.06.110

[11] F. Kazemi, N. Asgarkhani, and R. Jankowski, "Machine learning-based seismic response and performance assessment of reinforced concrete buildings," *Archives of Civil and Mechanical Engineering*, vol. 23, no. 2, p. 94, 2023. https://doi.org/10.1007/s43452-023-00631-9

[12] Z. M. Yaseen, "Machine learning models development for shear strength prediction of reinforced concrete beam: a comparative study," *Sci Rep*, vol. 13, no. 1, p. 1723, 2023. https://doi.org/10.1038/s41598-023-27613-4

[13] I. Alhalil and M. F. Gullu, "Predicting Main Characteristics of Reinforced Concrete Buildings Using Machine Learning," *Buildings*, vol. 14, no. 9, p. 2967, 2024. https://doi.org/10.3390/buildings14092967

[14] Z. Tuna Deger and G. Taskin Kaya, "A Novel GPR-Based Prediction Model for Cylic Backbone Curves of Reinforced Concrete Shear Walls," *arXiv e-prints*, p. arXiv-2111, 2021. https://doi.org/10.1016/j.engstruct.2022.113874

[15] M. Mohammadagha, M. Najafi, V. Kaushal, and A. M. A. Jibreen, "Machine learning models for reinforced concrete pipes condition prediction: The state-of-the-art using artificial neural networks and multiple linear regression in a Wisconsin case study," *arXiv preprint arXiv:2502.00363*, 2025. https://doi.org/10.48550/arXiv.2502.00363

[16] H. Naderpour, M. Mirrashid, and P. Parsa, "Failure mode prediction of reinforced concrete columns using machine learning methods," *Eng Struct*, vol. 248, p. 113263, 2021. https://doi.org/10.1016/j.engstruct.2021.113263

[17] S. Das and S. Choudhury, "Influence of effective stiffness on the performance of RC frame buildings designed using displacement-based method and evaluation of column effective stiffness using ANN," *Eng Struct*, vol. 197, p. 109354, 2019. https://doi.org/10.1016/j.engstruct.2021.113263

[18] P. Gulati, A. Sharma, and M. Gupta, "Theoretical study of decision tree algorithms to identify pivotal factors for performance improvement: A review," *Int. J. Comput. Appl*, vol. 141, no. 14, pp. 19–25, 2016.

[19] A. B. Hassanat, M. A. Abbadi, G. A. Altarawneh, and A. A. Alhasanat, "Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach," *arXiv preprint arXiv:1409.0919*, 2014. https://doi.org/10.48550/arXiv.1409.0919

[20] V. Prasath, H. A. A. Alfeilat, O. Lasassmeh, A. Hassanat, and A. S. Tarawneh, "Distance and similarity measures effect on the performance of K-nearest neighbor classifier—A review. arXiv 2017," *arXiv preprint arXiv:1708.04321*.

[21] R. P. Sheridan, A. Liaw, and M. Tudor, "Light gradient boosting machine as a regression method for quantitative structure-activity relationships," *arXiv preprint arXiv:2105.08626*, 2021. https://doi.org/10.48550/arXiv.2105.08626

[22] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Adv Neural Inf Process Syst*, vol. 30, 2017.

[23] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992. https://doi.org/10.1016/S0893-6080(05)80023-1

[24] B. Pavlyshenko, "Using stacking approaches for machine learning models," in *2018 IEEE second international conference on data stream mining & processing (DSMP)*, IEEE, 2018, pp. 255–258. https://doi.org/10.1109/DSMP.2018.8478522

[25] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[26] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.

[27] D. Arpit, H. Wang, Y. Zhou, and C. Xiong, "Ensemble of averages: Improving model selection and boosting performance in domain generalization," *Adv Neural Inf Process Syst*, vol. 35, pp. 8265–8277, 2022.

[28] M. A. I. Neloy, N. Nahar, M. S. Hossain, and K. Andersson, "A weighted average ensemble technique to predict heart disease," in *Proceedings of the Third International Conference on Trends in Computational and Cognitive Engineering: TCCE 2021*, Springer, 2022, pp. 17–29. https://doi.org/10.1007/978-981-16-7597-3_2

[29] A. Dogan and D. Birant, "A weighted majority voting ensemble approach for classification," in *2019 4th international conference on computer science and engineering (UBMK)*, IEEE, 2019, pp. 1–6. https://doi.org/10.1109/UBMK.2019.8907028

[30] F. Leon, S.-A. Floria, and C. Bădică, "Evaluating the effect of voting methods on ensemble-based classification," in *2017 IEEE international conference on INnovations in intelligent Systems and applications (INISTA)*, IEEE, 2017, pp. 1–6. https://doi.org/10.1109/INISTA.2017.8001122

[31] L. Rokach, "Ensemble-based classifiers," *Artif Intell Rev*, vol. 33, no. 1, pp. 1–39, 2010. https://doi.org/10.1007/s10462-009-9124-7