CIVIL ENGINEERING JOURNAL

Article no. X

THE CIVIL ENGINEERING JOURNAL Y-20ZZ

USING THE AUTOMATED RANDOM FOREST APPROACH FOR OBTAINING THE COMPRESSIVE STRENGTH PREDICTION OF RCA

*Yujie Wu[1, 2, *], Xiaoming He[3]*

1. *School of Civil Engineering, Chongqing University, Chongqing,400044, China ; wuyjcqcn@sina.com*
2. *School of International Business and Management, Chongqing Institute of Foreign Studies, Chongqing,401120, China*
3. *Henan Communications Planning & Design Institute Co., Ltd., Zhengzhou, Henan,451450, China*

ABSTRACT

The intricate relationships and cohesiveness among numerous components make the task of designing mixture proportions for high-performance concrete (HPC) a challenging endeavour. Machine learning (ML) algorithms are indeed efficacious in mitigating this predicament. However, their lack of an explicit correlation between mixture proportions and compressive strength renders them opaque black box models. To surpass this constraint, the present research puts forward a semi-empirical methodology that involves the utilization of tactics such as non-dimensionalization and optimization. The methodology proposed exhibits a remarkable level of accuracy in predicting compressive strength across various datasets, exemplifying its all-encompassing applicability to diverse datasets.Furthermore, the exact association furnished by semi-empirical equations is a valuable asset for engineers and researchers operating in this domain, especially concerning their prognostic capabilities. The compressive strength of concrete holds significant importance in designing high-performance concrete, and achieving an optimal mixture proportion necessitates a comprehensive comprehension of the complex interplay among diverse factors, including the type and proportion of cement, water-cement ratio, size and type of aggregate, curing conditions, and admixtures. The semi-empirical approach put forth in this study presents a potential remedy to the intricate undertaking by establishing a more unequivocal correlation between mixture ratios and compressive strength.

KEYWORDS

High-performance concrete, Random forest, Crystal Structure Algorithm, Bonobo Optimizer, Sunflower Optimization Algorithm

INTRODUCTION

Concrete materials are the predominant construction materials utilized in contemporary engineering structures. The construction of concrete structures in intricate surroundings necessitates the employment of high-performance concrete (HPC), which is characterized by enhanced specifications regarding its strength, durability, and workability. HPC is a composite material comprised of high-grade cement, aggregates, water, and active fine admixtures. This concrete exhibit superior durability, workability, and strength properties [1]. HPC has found diverse applications in the construction industry, including but not limited to the development of houses, bridges, and various components [2, 3]. The utilization of concrete admixtures has the potential to minimize the dimensions of concrete structures, decrease their weight, curtail the material

requirement, enhance the endurance of concrete structures, and extend their serviceability. HPC can be formulated by incorporating a range of enhancing agents such as mineral admixtures, chemical admixtures, and fibrous materials into the concrete mixture [4–7].

The precise anticipation of concrete strengths has a significant impact on the effectiveness of material utilization and the structural safety of civil infrastructure [8]. Moreover, a failure to properly acknowledge the inherent robustness of concrete may result in a superfluous application of cement, thereby intensifying the release of CO2 [9]. In light of this, considerable endeavors have been undertaken over the last several years to establish prognostic models which establish a correlation between the mixture composition of concrete and its corresponding potency. Ideally, a predictive model ought to furnish noteworthy elucidations that culminate in the enhancement of concrete compositions characterized by exceptional constructability and durability at a reduced expenditure [10, 11]. As a consequence, there has been a proliferation of models that are formulated utilizing physics or chemistry-based relationships. Although conventional approaches have been instrumental in establishing strong correlations between critical parameters, including cement dosage, aggregate fraction, and air void content, with concrete strength, analyzing the compounded impacts of these features remains a difficult undertaking [12]. Furthermore, it is imperative to acknowledge that the conventional methods fail to consider the impact of ancillary factors, including but not limited to the chemical admixtures' inherent characteristics and appropriate dosage, the distribution of aggregate sizes, as well as the fineness modulus of aggregates. Ascertaining a potent and comprehensive prognostic model for concrete strength through conventional methods poses a formidable challenge [13–15].

The subfield of artificial intelligence, known as machine learning, endeavors to construct algorithms capable of acquiring knowledge from data sets and enhancing their aptitude over time. Machine Learning (ML) has garnered considerable attention in recent times owing to its inherent capacity to automate and optimize an extensive array of tasks, ranging from image recognition and natural language processing to predictive analytics [16, 17]. One of the significant advantages of ML is its capability to manage and dissect massive and intricate datasets, facilitating the identification of underlying patterns and enabling precise predictions with exceptional accuracy. The emergence of many machines learning (ML) techniques, including supervised and unsupervised learning, reinforcement learning, deep learning, and others, has been observed. ML has garnered widespread attention and is extensively applied across diverse industries, including healthcare, finance, manufacturing, and transportation. One potential application of machine learning (ML) is in the healthcare domain, in which it may be utilized to analyze medical images and identify any possible anomalies. Meanwhile, within the field of finance, ML may prove valuable in identifying instances of fraud and mitigating risks associated with financial activities [18–20].

In the past few decades, engineering has witnessed a surge in the application of machine learning methodologies for analyzing biological data of significant volume and complexity [21, 22]. The Random Forest (RF) methodology, comprising a collection of decision trees and incorporating intrinsic feature selection and interactions within the learning process, is widely preferred. The stated methodology is nonparametric, readily explicable, efficacious, and displays elevated prognostic competence across a variety of data sets. The domain of computational biology has recently witnessed a surge in the adoption of RF due to its distinctive benefits in addressing issues of limited sample size, feature space with high dimensionality, and intricate data structures [23].

The utilization of the Random Forest (RF) classifier has generated considerable attention due to its outstanding classification performance and efficient processing speed. The RF classifier can produce dependable and consistent categorizations by utilizing predictions derived from an ensemble of decision trees [24]. Additionally, this classifier can be effectively employed to select and prioritize variables demonstrating the greatest discriminatory capability between the specified categories. The importance of this resource stems from the extensive nature of remotely sensed data, which presents a challenge in identifying the most relevant variables. Such an undertaking requires significant time, is subject to errors, and can be subjective [25].

In this study, the Random Forest (RF) algorithm was used to forecast High-performance concrete (HPC) due to its ability to handle complex systems and multiple parameters using ML

techniques. Optimization algorithms were also implemented to improve the precision of the HPC systems. The following section describes three pioneering algorithms: the Crystal Structure Algorithm (CSA), Bonobo Optimizer (BO), and Sunflower Optimization Algorithm (SFO). Optimization algorithms refer to mathematical methods that aim to find the best solution to a particular problem and have been widely used to optimize various parameters associated with the design of HPC systems. This study presents a novel approach to predicting CS by integrating RF with three optimization algorithms. The results demonstrate that RF can accurately predict CS, and combining optimization algorithms significantly improves the model's efficiency.

MATERIALS AND METHODOLOGY

Data gathering

Multiple input variables are required for supervised machine learning algorithms to predict the compressive strength of recycled coarse aggregate-based concrete. The data used in this study were obtained from previously published literature and can be found in Appendix A. The models employed nine input variables, including water (W), cement (C), sand (S), natural coarse aggregate (NCA), recycled coarse aggregate (RCA), superplasticizers, size of RCA, the density of RCA, and water absorption of RCA. The outcome variable for the models was the compressive strength. The model's outcome is significantly affected by the number of input parameters and data points. In this study, 344 data points (mixes) were utilized to predict RCA-based concrete. The RF model was run using Python coding on the Anaconda software, and the RF software was utilized to run the model. The relative frequency distribution of each parameter used for the mixes was analyzed, and the descriptive statistical analysis for all parameters is listed in Table 1.

Tab. 1. The statistical properties of inputs and output

Variables	Statistical properties			
	Max	Min	Ave.	St. dev.
Water (kg/m3)	271	117.6	184.62	25.835
Cement (kg/m3)	600	158	386.86	82.160
FA (kg/m3)	1010	0	681.88	205.28
NCA (kg/m3)	1448.25	0	398.07	370.70
RCA (kg/m3)	1574.3	0	596.35	371.69
SP (kg/m3)	7.8	0	1.3241	2.0512
SRCA (mm)	32	10	19.755	4.0201
DRCA (kg/m3)	2661	0	2231.0	580.95
WRCA (%)	10.9	0	4.8046	2.2624
CS (MPa)	108.5	13.4	44.394	15.617

Random forest

principle of RF

A Random Forest classifier consists of a set of tree-structured classifiers represented as $\{\mathrm{d}(\mathrm{x}, \aleph_\mathrm{l}), \mathrm{l} = 1, \ldots\}$, with each tree making a unit vote to determine the most popular class for a given input x. Here, the $\{\aleph_\mathrm{l}\}$ denote independent identically distributed random vectors.

A random forest consists of multiple tree-structured classifiers developed using a training sample set and a random variable, $\{\aleph_\mathrm{l}\}$, for the l -th tree in Breiman's model [26]. The random variables are independent and identically distributed between any two trees, resulting in the creation of a classifier $d(x, \aleph_l)$, where x represents the input vector. By running the algorithm l times, a sequence of classifiers $\{d_1(x), d_2(x), \ldots, d_l(x)\}$ is generated, which can be utilized to create multiple

classification models. The final output of the system is determined by a standard majority vote, and the decision function is calculated accordingly [22].

$$D_{(x)} = argmax_g \sum_{i=1}^{l} F(d_i(x) = M) \quad (1)$$

Each tree has the right to vote for the best classification result for a given input variable, and the combination of these individual decision tree models is denoted as D(x). The output variable is M, and the indicator function is represented as $F(.)$ [27]. The process of selecting the most appropriate classification outcome is illustrated in Fig 2.

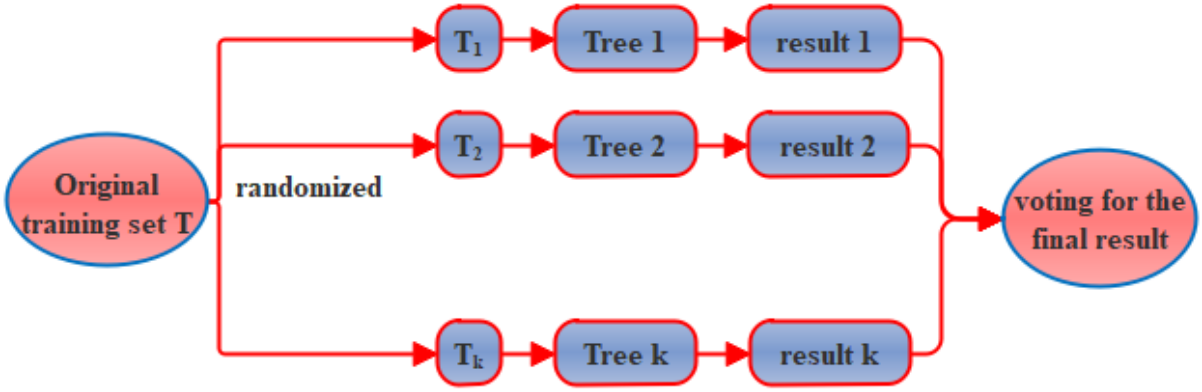

Fig.1_ Schematic of Random Forest

Characters of RF

The margin function [28], which is employed in Random Forest to assess the extent by which the average number of votes for the correct class at X, M, surpasses that for the incorrect class, can be defined as:

$$np(X, M) = av_l F(d_l(X) = M) - max_{j \neq M} av_l F(d_l(X) = j) \quad (2)$$

A higher value of the margin function indicates greater accuracy in the classification prediction and a higher level of confidence in the classification. The generalization error of this classifier can be defined as:

$$QA^* = Q_{X,M}(np(X, F) < 0) \quad (3)$$

Leo Breiman established that the random variable $h_k(X) = h(x, \aleph_k)$, follows the Strong Law of Large Numbers when the number of decision trees is sufficiently large. As the number of decision trees increases, QA* converges to a certain value for almost all sequences of $\aleph_1$ Breiman also demonstrated that Random Forest is not susceptible to overfitting and can yield a limiting value for the generalization error [16].

$$Q_{x,M}(Q_\theta(d_l(x, \theta) = M) - max_{j \neq M} Q_\theta(d(x, \theta) = j) < 0) \quad (4)$$

Another conclusion drawn by Leo Breiman is that there is a maximum limit for the generalization error:

$$QA^* \leq \bar{\rho}(1 - r^2)/r^2 \quad (5)$$

Two factors that influence the generalization error of RF are the strength of each tree in the forest, denoted by (r), and the correlation between the trees, represented by the average correlation value $\bar{\rho}$. A lower correlation value indicates reduced interdependence between the trees, which results in improved performance for the RF [29].

Out-of-bag estimation

The process of building a Random Forest includes growing a tree on a new training set that randomly selects features. The new training set is generated using bagging methods, which involve drawing samples from the original training set. Bagging is used in this process for two main reasons. Firstly, it has been observed that bagging can improve accuracy when random features are

employed. Secondly, bagging produces out-of-bag data, which can be utilized to provide continuous estimates for the QA* of RF, as well as strength and correlation estimates [30, 31].

Approximately 36.8% of the samples in T are not included in the l -th training set, Z_l, which is drawn from the original training set Z using bagging with replacement. Z_l contains N samples, where N is the total number of samples in Z. The probability of any given sample not being present in Z_l is $(1-1/N)^N$, which approaches a^{-1} as N increases. These samples that are not included in Z_l are known as out-of-bag data [32].

The OOB estimation algorithm employs out-of-bag data to estimate the classification performance. Each tree in the forest has an error estimate using the OOB method. The generalization error of the RF is calculated as the average of all tree error estimates for each tree included in the RF. Tibshirani, Wolpert, and Macready suggested using the OOB estimate as a component when estimating the generalization error, as it is faster to compute and less biased compared to cross-validation. Furthermore, the OOB estimate is more accurate than cross-validation. Using the OOB error estimate eliminates the need for setting aside a test set, as Breiman demonstrated that the accuracy of the OOB estimate is comparable to that of using a test set of the same size as the training set. Additionally, the out-of-bag method can be utilized to estimate the strength and correlation, providing an internal estimate that can aid in comprehending classification accuracy and identifying areas for improvement.

Crystal Structure Algorithm (CryStAl)

Minerals that display a regularly repeating or ordered crystalline structure in three dimensions are known as crystals. Crystalline solids can have varying sizes and shapes, and their properties may be either isotropic or anisotropic [33]. Tiny particles with a defined shape make up crystals. Through experimentation, various chemical and physical formulations have been studied and proposed. Additionally, the intricate symmetries and properties of crystals have influenced human creations such as mechanisms, structures, and artworks. This article uses the Bravais model to explain the crystal structure. In this model, infinite lattice geometry is considered, and the periodic structure described by the lattice geometry is specified along with the vector of the lattice positions as follows:

$$z = \sum s_i c_i \tag{6}$$

In the Bravais model, the periodic structure is described by the lattice geometry along with the vector of the lattice positions, where c_i represents the minimum vector of the principal crystal directions, s_i denotes the angular number of the crystal. This basic idea of crystals is presented with appropriate modifications for CryStAl mathematical modeling. In this model, each candidate solution of the optimization method is regarded as a single crystal lattice. An arbitrary number of crystal lattices is selected as initialization for the cycle.

$$\begin{bmatrix} wz_1 \\ cwz_2 \\ \vdots \\ wz_i \\ \vdots \\ wz_s \end{bmatrix} = \begin{bmatrix} x_1^1 \dots & x_1^j \dots & x_1^q \\ x_2^1 \dots & x_2^2 \dots & x_2^q \\ \vdots & \vdots & \vdots \\ x_i^1 \dots & x_i^2 \dots & x_i^q \\ \vdots & \vdots & \vdots \\ x_s^1 \dots & x_s^2 \dots & x_s^q \end{bmatrix}, \begin{cases} i = 1,2,3, \dots, s \\ j = 1,2,3, \dots, q \end{cases} \tag{7}$$

where s is the candidate solution and q is the dimension of the problem. In the search space, the initial positions of these crystals are randomly determined by:

$$x_i^j(0) = x_{i,min}^j + \gamma\left(x_{i,max}^j - x_{i,min}^j\right), \begin{cases} i = 1,2,3, \dots., s \\ j = 1,2,3, \dots, q \end{cases} \tag{8}$$

Where $x_i^j(0)$ characterizes the starting gem position, the least and greatest permitted values are characterized as $x_{i,max}^j$ and $x_{i,min}^j$ separately, the jth choice variable of the i -th candidate arrangement is within the indicated ρ. The primary crystals, according to the crystallographic concept of the "base," consist of all corner crystals. wz_{main} randomly determined considering the first generated crystal. In addition, the z_l the current value is ignored, and a random extraction method is

set for each tread. Crystals with optimal configuration determined by wz_r. D_v represents the mean of randomly selected crystals. To keep track of the position of a candidate solution in the search space, four types of update procedures are defined using fundamental network principles:

Simple cubic; $$wz_{new} = wz_{main} + wz_{old} \quad (9)$$

Best crystal cubicle; $$wz_{new} = z_1 wz_{zmain} + z_2 wz_r + wz_{old} \quad (10)$$

Mean crystal cubicle; $$wz_{new} = z_1 wz_{zmain} + z_2 D_v + wz_{old} \quad (11)$$

M&B crystal cubicle; $$wz_{new} = wz_{old} + z_1 wz_{zmain} + z_2 wz_r + z_3 D_v \quad (12)$$

Bonobo Optimizer (BO)

The BO algorithm, developed by Das et al. [34], is a modern metaheuristic algorithm that draws inspiration from the reproducible approach and social behaviour of bonobos. The BO algorithm is formulated in a population-based structure. Bonobos are typically divided into smaller groups, known as fission, for activities such as foraging for food and sleeping at night. To enhance the effectiveness of the search process, this behaviour was incorporated into the BO algorithm.

The BO algorithm surveys the natural strategies and behaviours of bonobos to achieve optimal responses. Bonobos exhibit various mating strategies, including extra-group mating, promiscuity, restrictiveness, and consortship. These strategies are used to generate new bonobo populations [35]. The mating strategies may be modified based on the phase condition, which can be either negative (NP) or positive (PP). The PP state indicates a favourable condition within the bonobo community characterized by sufficient food, genetic diversity among bonobos, successful mating, and safety from neighbouring communities. Conversely, the NP state represents an unfavourable circumstance within society.

Restrictive and Promiscuous Mating Techniques

The mating strategy of the bonobos is represented by the phase probability parameter (w_w). Initially, w_w is set to 0.5 and is incremented at each iteration. If a randomly generated number, p, falls within the range from zero to one, a new bonobo is created. The value of p is compared to w_w using Eq. (13):

$$s\_Bn_j = Bn_j^i + p_1 h^e \left(e_j^{Bn} - Bn_j^h\right) + (1 - p_1) h^h flag(Bn_j^h - Bn_j^w) \quad (13)$$

Bn = bonobo

s_Bn_j and e_j^{Bn} are the $j-$th new offspring' variables

j is a value that varies between 0 and 1

c is referred to the variables' number

p_1 determines a random value within the range from 0 to 1

Bn_j^i and Bn_j^w determine the values of variables related to $i-th$ and $w-th$ bonobos, respectively.

h^e and h^h are referred to as sharing coefficients for e^{Bn} and $w-th$ bonobos, respectively.

As the best response of $i-th$ bonobo obtains a better consequence than $w-th$ bonobos, promiscuous mating happens. In this situation, the flag is indicated 1. On the other hand, for limited mating e^{Bn} are assigned as -1.

Extra-Group and Consortship Mating Techniques

If part w_w is lesser than p, these types of mating will happen. On the other hand, if p_2 is equal to or less than the extra group (w_{xyw}) probability, this will result in upgrading the solution via extra-group mating.

$$\begin{cases} Bn_j^i + c^{\left(p_3^2+p_3-2p_3^{-1}\right)\left(Var\_max_j - Bn_j^i\right)} & e_j^{Bn} \geq Bn_j^i \\ Bn_j^i - c^{\left(-p_4^2+2p_4-2p_4^{-1}\right)\left(Bn_j^i - Var\_min_j\right)} & p_3 \leq w_c \\ Bn_j^i - c^{\left(p_3^2+p_3-2p_3^{-1}\right)\left(Bn_j^i - Var\_min_j\right)} & e_j^{Bn} \leq Bn_j^i \\ Bn_j^i + c^{\left(-p_4^2+2p_4-2p_4^{-1}\right)\left(Var\_max_j - Bn_j^i\right)} & p_3 \geq w_c \end{cases} \quad (14)$$

The w_c is started with 0.5 with an incremental upgrading related to the evolution's nature, and it optimizes the searching process for the foremost hopeful output. Var_min_j and Var_max_j denote the lowest and highest boundaries of the $j-th$ variable, respectively.

In other cases, using the consortship mating strategy, a novel offspring is generated, where the amount of p_2 is found to be higher than that of w_{xyw}, as Eq. (15):

$$s\_Bn_j = \begin{cases} s\_Bn_j + c^{p_5} flag(1+p_1)\left(Bn_j^i - Bn_j^w\right) & p_6 \leq w_c \\ Bn_j^w & otherwise \end{cases} \quad (15)$$

In these equations, p_1, p_2, p_3, p_4, p_5 are random numbers between 0 and 1.

Sunflower Optimization Algorithm (SFO)

The main reason for employing the SFO algorithm to optimize problems is to leverage the power of soft computing capabilities [36]. The SFO algorithm is a heuristic population-based algorithm inspired by nature, specifically by the concept of simulating the orientation of sunflowers to maximize the amount of radiation received from the sun [37]. Sunflowers exhibit a daily periodic sequence that tracks the sun's movement to maximize radiation intake and reverse direction in the evening [38]. The orientation of a sunflower towards the sun is determined by Eq. (16), which assumes that each sunflower produces only one pollen gamete according to the model.

$$R_i = \frac{X^* - X_i}{\|X^* - X_i\|}, \qquad i = 1,2,\dots,z_c \quad (16)$$

The direction r in which sunflowers take a step is shown by Eq. (17):

$$b_i = \beta + G_i(X_i + X_{i-1}) \times \|X_i + X_{i-1}\| \quad (17)$$

The pollination probability is represented by $G_i(X_i + X_{i-1})$, while β denotes a constant that characterizes the "inertial" movement of the sunflowers. Individuals that are closer to the sun exhibit smaller movements as they refine their local position, while those that are farther away move normally. Constraints on the magnitude of these steps are introduced by Eq. (18):

$$b_{max} = \frac{\|X_{max} - X_{min}\|}{2 \times V_{pop}} \quad (18)$$

The lower and upper limits are represented by X_{min} and X_{max}, respectively, while V_{pop} denotes the total number of plants in the population. The new plant is determined using the following Eq:

$$X_{i+1} = X_i + b_i + R_i \quad (19)$$

The SFO algorithm involves the following simple steps:

1) Identify the individual within the population that receives the highest evaluation as the sun.
2) Randomly generate the population.
3) Adjust the orientation of the remaining population members to maximize their exposure to the sun. [39].

In addition, Fig 2 has determined the flowchart of SFO.

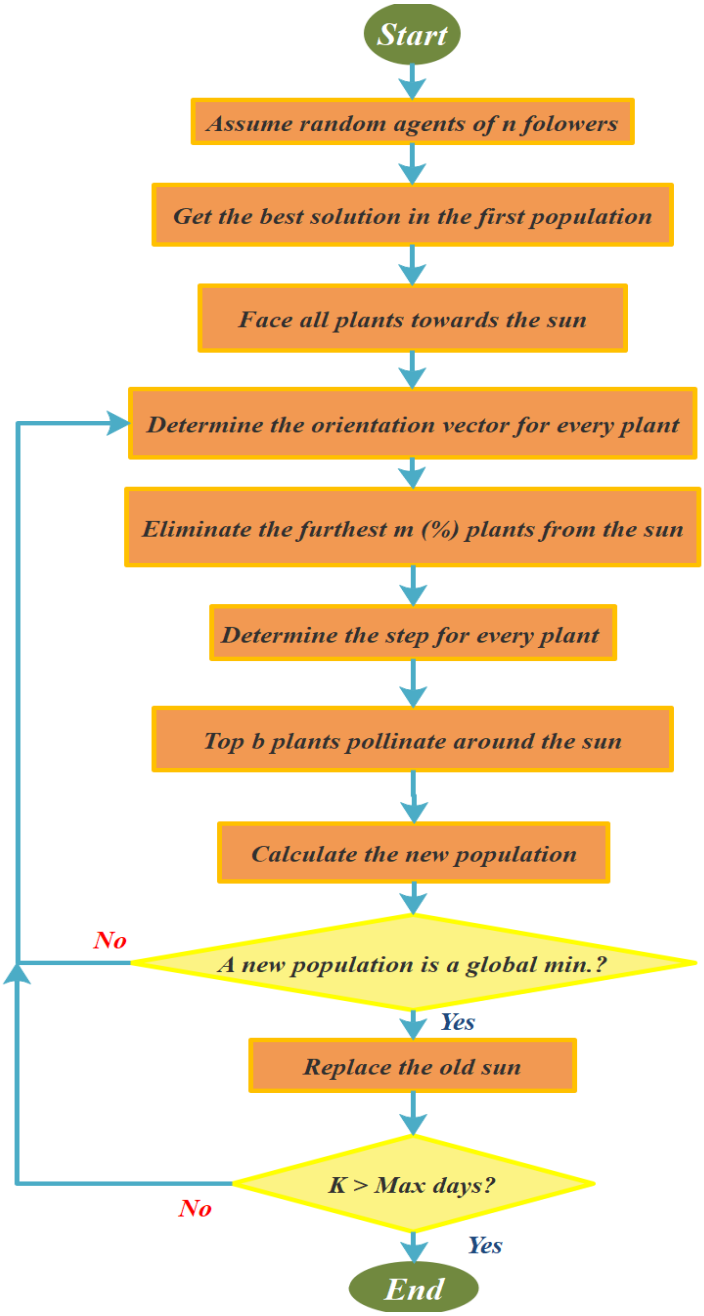

Fig. 2_SFO Algorithm.

Performance evaluation methods

As mentioned earlier, this article uses various metrics to evaluate the models, such as the root mean square error (RMSE), correlation coefficient (R^2), mean square error (MSE), median absolute percentage error (MDAPE), and ratio of RMSE (RSR). These metrics are calculated using equations (20) through (24):

$$R^2 = \left(\frac{\sum_{i=1}^{p} (a_i - \bar{a})(k_i - \bar{k})}{\sqrt{\left[\sum_{i=1}^{p} (a_i - a)^2\right]\left[\sum_{i=1}^{p} (k_i - \bar{k})^2\right]}} \right)^2 \quad (20)$$

$$RMSE = \sqrt{\frac{1}{P}\sum_{i=1}^{p}(k_i - a_i)^2} \quad (21)$$

$$RSR = \frac{RMSE}{St.Dev.} \quad (22)$$

$$MSE = \frac{1}{P}\sum_{i=1}^{p} d_i^2 \quad (23)$$

$$MDAPE = 100 \times median\left(\frac{|k_i - \bar{k}|}{|a_i - \bar{a}|}\right) \quad (24)$$

In this context, a_i and k_i refer to the predicted and experimental values, respectively. The mean values of the predicted and experimental samples are represented by a and k. Alternatively, P denotes the number of samples being considered.

RESULTS AND DISCUSSION

This section pertains to the evaluation of the hybrid models that have been introduced. The performance metrics have been divided into training and testing, with 70% of the data samples allocated for training and the remaining 30% for testing. For the R^2 metric, a higher value is considered better, while for the other metrics, the objective is to minimize the error and achieve the most favorable outcome. Any improvement or deterioration in the performance metrics during the testing phase indicates the efficacy or inadequacy of the training of the model during the training phase. The performance evaluation of the models is presented in Table 2. The R^2 the highest value was in $RFCS_{test} = 0.9969$, the obtained value that was the lowest by$RFSO_{test} = 0.9738$. In RMSE and RSR, the most suitable values of 1.112501 and 0.0654 were acquired by $RFCS_{test}$, correspondingly and the RMSE value for $RFSO_{test} = 4.913095$, while the RSR value for $RFSO_{train} = 0.293$, indicating that the performance of the model is weakest in these two metrics. In MDAPE, like the other two error assessors, $RFSO_{test}$ had the lowest value of 6.7167, while $RFCS_{test}$ attained the highest value of 1.3275. In terms of MSE, which represents the highest value of the pertinent performance standard, the most acceptable outcome was achieved by$RFSO_{test}$ with a 24.1385, whereas the poorest result was attained by $RFCS_{test}$ with a score of 1.2377.

Tab. 2: The results achieved from the hybridized models

Models	RFCS		RFSO		RFBO	
Status	Train	Test	Train	Test	Train	Test
RMSE	1.584589	1.112501	4.401471	4.913095	1.71693	2.794363
R2	0.991554	0.996913	0.980004	0.973828	0.992097	0.982405
MSE	2.5109	1.2377	19.3729	24.1385	2.9478	7.8085
MDAPE	1.9341	1.3275	6.357	6.7167	2.397	3.6994
RSR	0.1055	0.0654	0.293	0.289	0.1143	0.1644

Figure 3 is a scatter plot of the predicted values versus the actual values for three hybrid models: RFCS, RFBO, and RFSO. The scatter plot has a centerline and two linear fits that represent the training and testing phases. The scatter plot shows that all three models have a strong positive correlation between the predicted and actual values, meaning that the models are capable of accurately predicting the values. However, the scatter plot also shows that RFCS has the tightest clustering of data points around the linear fit lines, indicating that it is the most accurate of the three models. RFBO and RFSO also demonstrate a strong correlation, although with slightly more scattered data points. The linear fit lines for both models exhibit a similar slope and intercept, indicating that their predictive capabilities are comparable.

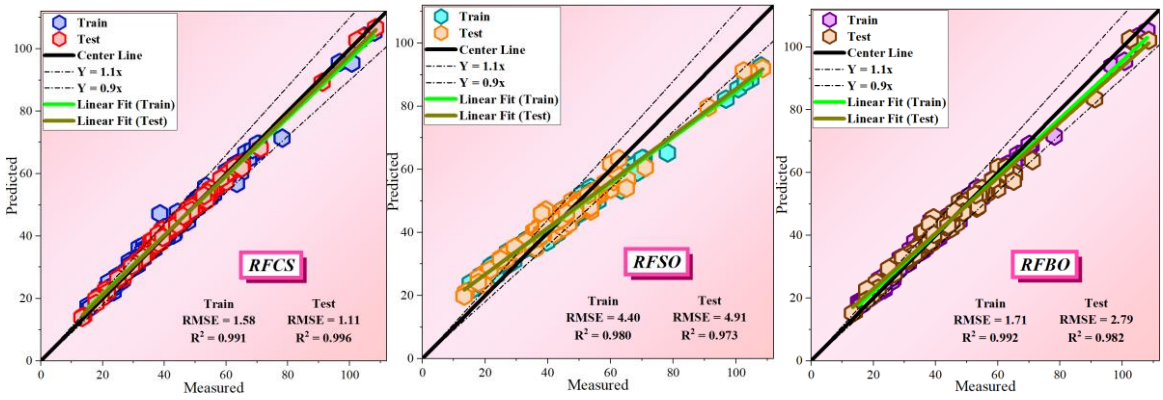

Fig.3_ The scatter plot for developed hybrid models.

Figure 4 is a line-symbol plot comparing the predicted and measured samples for the three hybrid models: RFCS, RFBO, and RFSO. The plot shows how closely the predicted values align with the measured values, highlighting the performance of the models. RFCS exhibits the highest level of accuracy, with the predicted values closely following the measured values throughout the entire dataset. RFBO and RFSO also demonstrate a strong correlation between the predicted and measured values but with slightly more deviations from the measured values. This suggests that while RFBO and RFSO are still effective, they may not be as precise as RFCS.

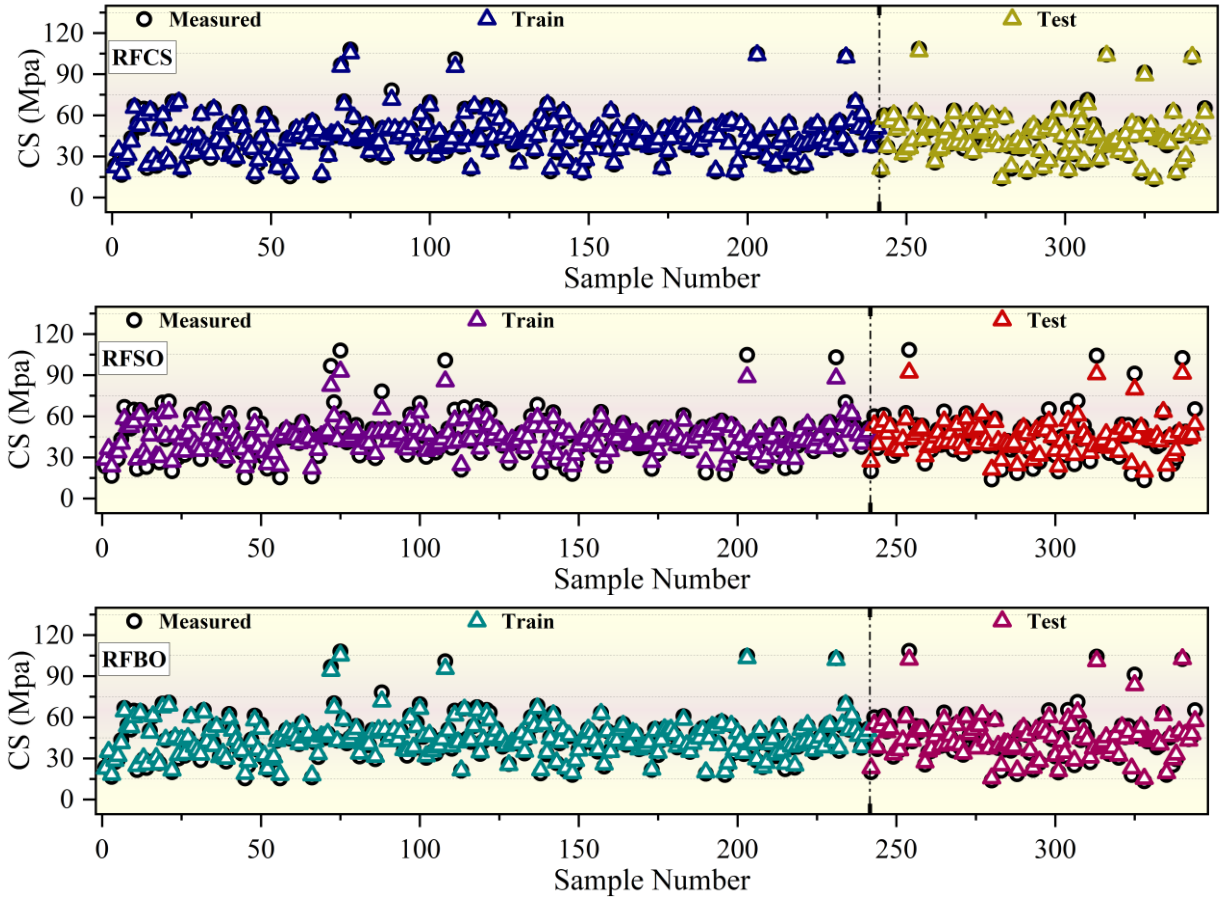

Fig.4_The comparison of predicted and measured samples based on a scatter plot.

Figure 5 illustrates the distribution of error percentages for the developed models. The x-axis represents the error percentage, and the y-axis shows the frequency of occurrence. The graph shows that RFCS has the lowest error percentage, with most error percentages falling within the range of 0-10%. On the other hand, RFBO and RFSO have a wider distribution of error percentages,

with more values exceeding 10%. Furthermore, RFBO and RFSO have a right-skewed distribution, implying that a few data points have relatively high error percentages. The graph also reveals that all three models show lower error percentages during the testing phase than during the training phase, indicating the risk of overfitting the training data. In conclusion, the graph provides a clear visualization of the error percentage distribution for the developed models and highlights the superior accuracy of RFCS.

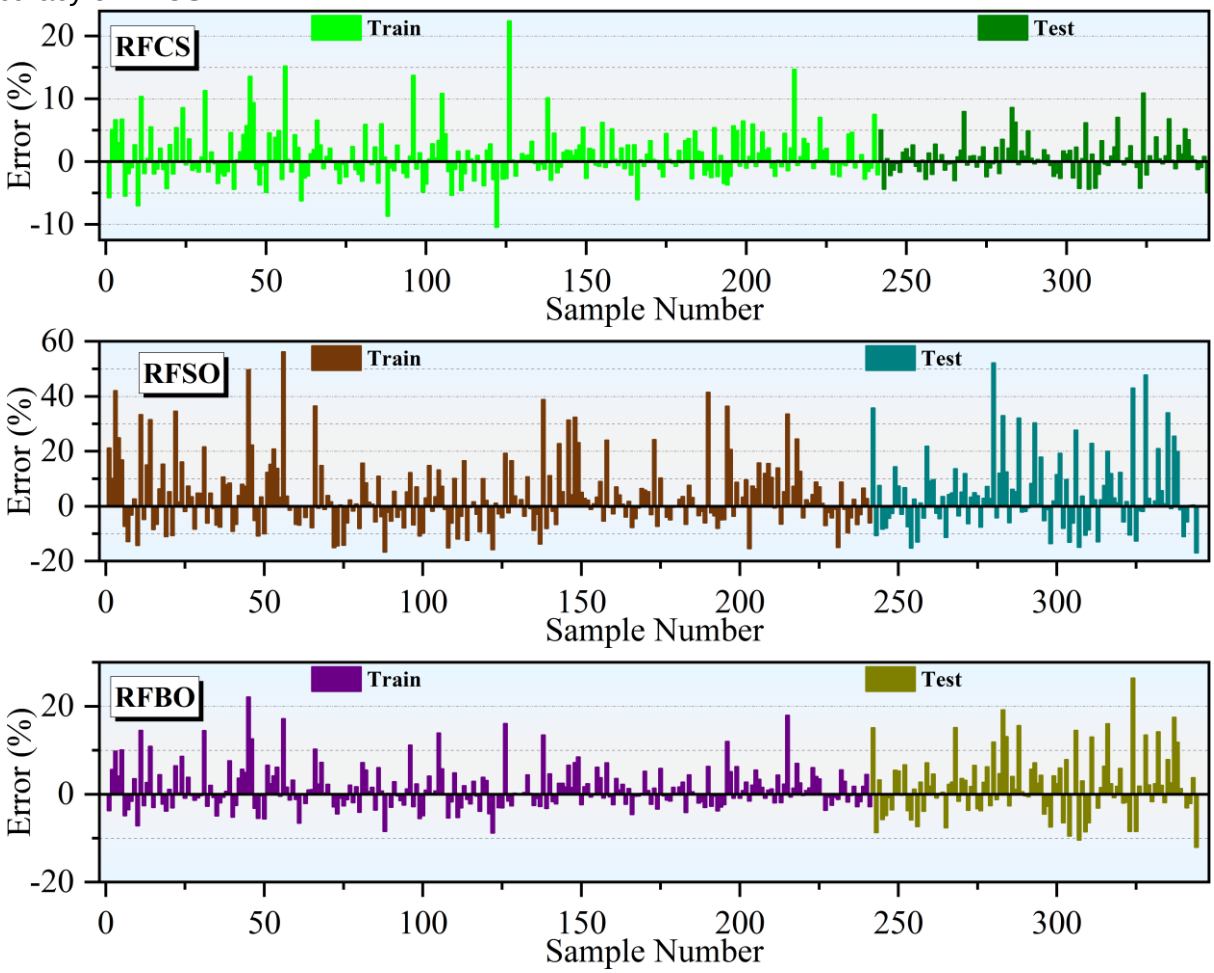

Fig. 5_The error rate percentage for the models being showcased.

Figure 6 shows a box plot of the error percentage for the presented models. RFCS had an average error of 0% during the training phase, with a sharp normal distribution and minimal dispersion observed. The dispersion of errors was also good, with values below 10%. In contrast, RFSO had dispersion in both phases, and a flatter normal distribution was observed. Nevertheless, the model achieved its highest error percentage, below 10%. RFBO had the most significant and diverse errors, but an outlier data point was only collected during the testing phase, exceeding 10% of the data, which is considered uncommon. The Gaussian distribution of RFSO was more widely dispersed compared to the other two models, with a lower frequency of occurrence around zero. In general, all three models performed well, but RFCS yielded the most favorable outcomes.

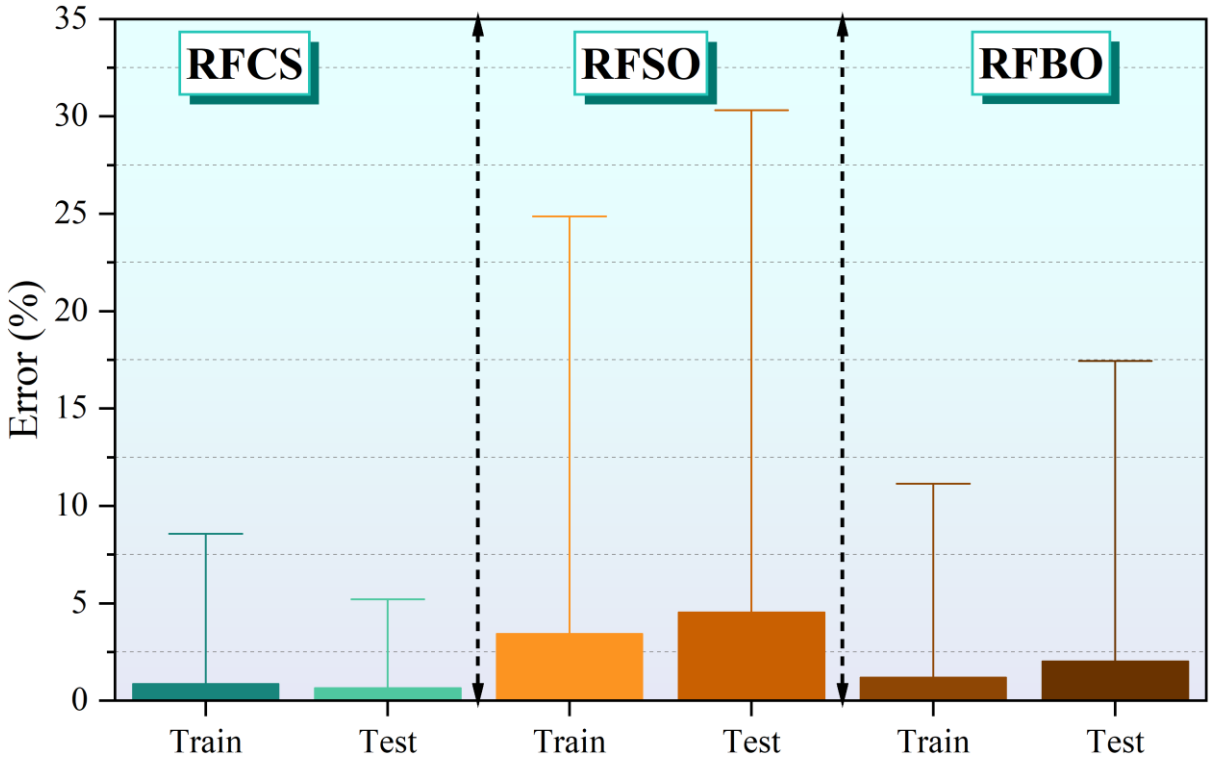

Fig. 6_The error-box for error percentage of presented models.

CONCLUSION

High-performance concrete (HPC) is a form of concrete renowned for its exceptional workability, longevity, and robustness. The CS of concrete is widely regarded as a critical mechanical property. Obtaining a comprehensive understanding of CS through laboratory experimentation is a process that requires significant time and physical labor. The application of machine learning is a potential solution to this challenge. This research endeavor sought to employ the random forest (RF) machine learning algorithm to forecast coiled tubing fatigue life in HPC applications. The study utilized a hybrid approach that combined the RF model with optimization algorithms, such as CSA, BO, and SFO, to enhance accuracy. The models' performance was evaluated using R^2, RMSE, RSR, MSE, and MDAPE. The study's results revealed that the RFCS models had the highest R^2 values, while RFSO had the lowest R^2 value of 0.973828. The error indicators, including RMSE, MSE, RSR, and MDAPE, indicated that RFCS models generally had lower error indicators, suggesting better performance compared to RFBO and RFSO models. The lowest RMSE values were observed in both training and testing phases among the RFCS models, with a narrow dispersion range, suggesting consistent and accurate performance in predicting HPC. However, the error percentage values were relatively consistent across all models, indicating the need for further improvements. Overall, the study suggests that the RF hybrid models, especially the RFCS models, are proficient in predicting HPC and can provide precise and reliable results for engineering applications. Additionally, the study demonstrates the effectiveness of the hybrid approach in improving the models' accuracy.

REFERENCES

[1]. Sadowski Ł, Nikoo M, Nikoo M (2018) Concrete compressive strength prediction using the imperialist competitive algorithm. Computers and Concrete, An International Journal 22:355–363

[2]. Afroughsabet V, Biolzi L, Ozbakkaloglu T (2016) High-performance fiber-reinforced concrete: a review. J Mater Sci 51:6517–6551

[3]. Rajasekaran S, Amalraj R (2002) Prediction of strength and workability of high performance concrete composites using artificial neural networks

[4]. Yeh I-C (1999) Predicting the compressive strength and slump of high strength concrete using neural. Journal of Computing in Civil Engineering 13:36–42
[5]. Chou J-S, Pham A-D (2013) Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength. Constr Build Mater 49:554–563
[6]. Mehdi Yaltaghian Khiabani1, Behnam sedaghat 2, Parisa Ghorbanzadeh3, Negin Porroustami4, Seied Mehdy Hashemy Shahdany5, Yousef Hassani6 SN (2023) Application of a Hybrid Hydro-economic Model to Allocate Water over the Micro- and Macro-scale Region for Enhancing Socioeconomic Criteria under the Water Shortage Period. Water Economics and Policy
[7]. Masoumi F, Najjar-Ghabel S, Safarzadeh A, Sadaghat B (2020) Automatic calibration of the groundwater simulation model with high parameter dimensionality using sequential uncertainty fitting approach. Water Supply 20:3487–3501. https://doi.org/10.2166/ws.2020.241
[8]. Aïtcin P-C (1998) High performance concrete. CRC press
[9]. Benhelal E, Zahedi G, Shamsaei E, Bahadori A (2013) Global strategies and potentials to curb CO2 emissions in cement industry. J Clean Prod 51:142–161
[10]. Behnood A, Golafshani EM (2018) Predicting the compressive strength of silica fume concrete using hybrid artificial neural network with multi-objective grey wolves. J Clean Prod 202:54–64
[11]. Lyngdoh GA, Zaki M, Krishnan NMA, Das S (2022) Prediction of concrete strengths enabled by missing data imputation and interpretable machine learning. Cem Concr Compos 128:104414
[12]. Zhang X, Akber MZ, Zheng W (2021) Prediction of seven-day compressive strength of field concrete. Constr Build Mater 305:124604
[13]. Chakraborty I, Bodurtha KJ, Heeder NJ, et al (2014) Massive electrical conductivity enhancement of multilayer graphene/polystyrene composites using a nonconductive filler. ACS Appl Mater Interfaces 6:16472–16475
[14]. Moutassem F, Chidiac SE (2016) Assessment of concrete compressive strength prediction models. KSCE Journal of Civil Engineering 20:343–358. https://doi.org/10.1007/s12205-015-0722-4
[15]. Lasisi A, Sadiq MO, Balogun I, et al (2019) A Boosted Tree Machine Learning Alternative to Predictive Evaluation of Nondestructive Concrete Compressive Strength. In: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). IEEE, pp 321–324
[16]. Biau G (2012) Analysis of a random forests model. The Journal of Machine Learning Research 13:1063–1095
[17]. Kim K (2003) Financial time series forecasting using support vector machines. Neurocomputing 55:307–319
[18]. Mahesh B (2020) Machine learning algorithms-a review. International Journal of Science and Research (IJSR)[Internet] 9:381–386
[19]. Zhou Z-H (2021) Machine learning. Springer Nature
[20]. Wang H, Lei Z, Zhang X, et al (2016) Machine learning basics. Deep learning 98–164
[21]. Breiman L (1996) Bagging predictors. Mach. Learn.
[22]. Liu Y, Wang Y, Zhang J (2012) New machine learning algorithm: Random forest. In: Information Computing and Applications: Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012. Proceedings 3. Springer, pp 246–252
[23]. Breiman L (2001) Random forests. Mach Learn 45:5–32
[24]. Genuer R, Poggi J-M, Tuleau-Malot C (2010) Variable selection using random forests. Pattern Recognit Lett 31:2225–2236
[25]. Du P, Samat A, Waske B, et al (2015) Random forest and rotation forest for fully polarized SAR image classification using polarimetric and spatial features. ISPRS journal of photogrammetry and remote sensing 105:38–53
[26]. Biau G, Scornet E (2016) A random forest guided tour. Test 25:197–227
[27]. Sarica A, Cerasa A, Quattrone A (2017) Random forest algorithm for the classification of neuroimaging data in Alzheimer's disease: a systematic review. Front Aging Neurosci 9:329
[28]. Lin W, Wu Z, Lin L, et al (2017) An ensemble random forest algorithm for insurance big data analysis. Ieee access 5:16568–16575
[29]. Kulkarni AD, Lowe B (2016) Random forest algorithm for land cover classification
[30]. Livingston F (2005) Implementation of Breiman's random forest machine learning algorithm. ECE591Q Machine Learning Journal Paper 1–13

[31]. Khajavi H, Rastgoo A (2023) Predicting the carbon dioxide emission caused by road transport using a Random Forest (RF) model combined by Meta-Heuristic Algorithms. Sustain Cities Soc 93:104503

[32]. Mohapatra N, Shreya K, Chinmay A (2020) Optimization of the random forest algorithm. In: Advances in Data Science and Management: Proceedings of ICDSM 2019. Springer, pp 201–208

[33]. Talatahari S, Azizi M, Tolouei M, et al (2021) Crystal structure algorithm (CryStAl): a metaheuristic optimization method. IEEE Access 9:71244–71261

[34]. Das AK, Pratihar DK (2019) A new bonobo optimizer (BO) for real-parameter optimization. In: 2019 IEEE Region 10 Symposium (TENSYMP). IEEE, pp 108–113

[35]. Das AK, Pratihar DK (2019) A new bonobo optimizer (BO) for real-parameter optimization. In: 2019 IEEE Region 10 Symposium (TENSYMP). IEEE, pp 108–113

[36]. Yuan Z, Wang W, Wang H, Razmjooy N (2020) A new technique for optimal estimation of the circuit-based PEMFCs using developed Sunflower Optimization Algorithm. Energy Reports 6:662–671

[37]. Gomes GF, da Cunha SS, Ancelotti AC (2019) A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates. Eng Comput 35:619–626

[38]. Shaheen MAM, Hasanien HM, Mekhamer SF, Talaat HEA (2019) Optimal power flow of power systems including distributed generation units using sunflower optimization algorithm. IEEE Access 7:109289–109300

[39]. El-Sehiemy RA, Hamida MA, Mesbahi T (2020) Parameter identification and state-of-charge estimation for lithium-polymer battery cells using enhanced sunflower optimization algorithm. Int J Hydrogen Energy 45:8833–8842