Front. Struct. Civ. Eng. 2022, 16(7): 928–945
https://doi.org/10.1007/s11709-022-0837-x

RESEARCH ARTICLE

Assessment of different machine learning techniques in predicting the compressive strength of self-compacting concrete

Van Quan TRAN*, Hai-Van Thi MAI, Thuy-Anh NGUYEN, Hai-Bang LY

Faculty of Civil Engineering, University of Transport Technology, Hanoi 100000, Vietnam
**Corresponding author. E-mail: quantv@utt.edu.vn*

ABSTRACT The compressive strength of self-compacting concrete (SCC) needs to be determined during the construction design process. This paper shows that the compressive strength of SCC (CS of SCC) can be successfully predicted from mix design and curing age by a machine learning (ML) technique named the Extreme Gradient Boosting (XGB) algorithm, including non-hybrid and hybrid models. Nine ML techniques, such as Linear regression (LR), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Trees (DTR), Random Forest (RF), Gradient Boosting (GB), and Artificial Neural Network using two training algorithms LBFGS and SGD (denoted as ANN_LBFGS and ANN_SGD), are also compared with the XGB model. Moreover, the hybrid models of eight ML techniques and Particle Swarm Optimization (PSO) are constructed to highlight the reliability and accuracy of SCC compressive strength prediction by the XGB_PSO hybrid model. The highest number of SCC samples available in the literature is collected for building the ML techniques. Compared with previously published works' performance, the proposed XGB method, both hybrid and non-hybrid models, is the most reliable and robust of the examined techniques, and is more accurate than existing ML methods (R^2 = 0.9644, *RMSE* = 4.7801, and *MAE* = 3.4832). Therefore, the XGB model can be used as a practical tool for engineers in predicting the CS of SCC.

KEYWORDS compressive strength, self-compacting concrete, machine learning techniques, particle swarm optimization, extreme gradient boosting

1 Introduction

Self-compacting concrete (SCC) can flow and compact under the effect of self-weight and does not need mechanical vibration in order to ensure uniformity [1]. In contrast to conventional concrete, SCC more often uses mineral fillers and water-reducing admixtures. In addition, SCC often uses coarse aggregates with smaller particle sizes than conventional concrete [2]. The benefits of SCC include increased economic efficiency due to reduced construction time, labor costs, energy costs, and equipment prices [3]. Furthermore, SCC can be applied in complex construction shapes and difficult construction conditions and can improve the working and living environment because vibration and noise pollution can be avoided. Use of SCC can also favor automation of construction. Because of these benefits SCC has been regarded as a remarkable application in civil engineering, including for bridge building, road construction, and tunnel construction [4,5]. However, constructions using SCC also have disadvantages such as structural defects and failure to guarantee compressive strength [6]. To overcome these disadvantages, one of the most critical mechanical properties needs to be correctly determined: the SCC compressive strength. In particular, due to the complex composition required for SCC to reach its desirable properties, the compressive strength of SCC (CS of SCC) depends on mixed composition design, mineral filler content, one or more chemical admixtures, and an optimum balance between the coarse aggregate and fine aggregate.

For decades, many researchers have determined the CS

Article history: Received Jan 7, 2022; Accepted Feb 22, 2022

of SCC by experimental methods. For example, Bouzoubaâ and Lachemi [7] determined the compressive strength and shrinkage of SCC using fly ash (FA). In this work, 40%, 50%, and 60% of cement content are replaced by FA. The compressive strength of SCC containing 50% FA after 28 d is equal to that of conventional SCC. However, one type of FA was used in this study. Busari et al. [8] conducted experiments to determine the influence of some selected cement brands on the CS of SCC used in pavement construction. The effect of temperature rise on the CS of SCC was studied by Rajah Surya et al. [9]. These tests demonstrate that the CS of SCC is very variable and is influenced by the mixed composition design and experimental settings. Furthermore, many errors can occur during the experimental procedures. and so it is necessary to have other methods to predict the CS of SCC, and machine learning (ML) algorithms appear as a potential alternative.

In recent years, based on available experimental databases, ML algorithms or artificial intelligence (AI) approaches have been widely used in the geotechnical field [10], structural engineering [11–13], fundamental mechanical problems [14], and material sciences [15–17]. In addition to many advanced computational approaches to predict concrete behaviors [18–21], the possibility of applying AI to predict various concrete properties have also been investigated, such as concrete beams' shear strength [22], the later-age CS of concrete [23], recycled aggregate concrete compressive strength [24], concretes containing blast furnace slag (BFS) and FA [25–27], high-performance concrete [28], rubber concrete compressive strength [16].

Concerning SCC, there have been many studies using one of the AI algorithms to predict compressive strength, such as Artificial Neural Network (ANN), Random Forest (RF), Fuzzy Logic (FL), and Support Vector Machine (SVM). Siddique et al. [29] used 80 data with six inputs (namely cement content, FA content, the ratio of water to binder, superplasticizer dosage, sand, coarse aggregate) to develop an ANN model to forecast the CS of SCC. The highest coefficient of determination (R^2) of this architecture was found to be about 0.9187. Using only 69 data samples, Abu Yaman et al. [30] proposed an ANN model with Levenberg Marquardt backpropagation algorithm to study SCC using FA. They used a coefficient of determination to evaluate the performance of such an ANN model, and the highest value was $R^2 = 0.81$. Asteris et al. [31,32], and Malagavell and Manalel [33] collected 205 and 169 data, respectively. They built ANN models for predicting the CS of SCC containing BFS and FA. In particular, Siddique et al. [29] used a Back-Propagation Neural Network (BPNN) and surrogate models [32]. The highest correlation coefficient of their model was $R^2 = 0.9604$. Moreover, some researchers used RF regression to evaluate the uniaxial CS of lightweight SCC [34] and used linear and nonlinear SVM to predict the CS of high volume FA SCC [35]. It is noted that these studies used a limited number of data, with only 130 and 340 data samples, respectively, and the highest value of R^2 was about 0.9388. Overall, the model's performance can be significantly improved by increasing the database's sample number and input distribution and by ML techniques.

Therefore, the aim of this investigation is to develop the best ML technique to predict the CS of SCC. The selection is evaluated by two criteria: the accuracy of the ML technique and the training process' reliability and speed. Determining a model that satisfies both of these criteria is crucial in practice in order to provide good prediction with minimum time and computation costs. Moreover, the model should allow engineers to predict problems without difficulty, even without an in-depth understanding of the algorithms. In this study, the collection of the highest number of data, including 1287 data samples from 44 published literature using 13 input variables, is firstly carried out. In the next step, nine ML techniques, namely Linear regression (LR), K-Nearest Neighbors (KNN), SVM, Decision Trees (DTR), RF, Gradient Boosting (GB), Extreme Gradient Boosting (XGB), ANN using two optimization algorithms LBFGS and SGD (denoted as ANN_LBFGS and ANN _SGD, respectively) are developed. In this step, to minimize the computation time, the default hyperparameters of nine ML techniques are taken from the Sklearn library [36] and XGBoost library [37] in Python. Hybrid ML techniques are also developed, in which the hyperparameters of the nine proposed ML techniques are tuned by the Particle Swarm Optimization (PSO) algorithm. The prediction accuracies of non-hybrid and hybrid models are compared to evaluate the robustness of different ML techniques in predicting the CS of SCC. In the last step, the feature importance is investigated to show the effect of input variables on the compressive strength prediction model.

2 Significance of the study

A number of ML techniques have been developed to predict the CS of SCC in the field of civil engineering, namely ANN [29], and a newly proposed normalization method [32], BPNN [38], FL [39], RF regression [40], SVM [41, 42], but the accuracy and reliability of the predictions could be better. As a result, the following main points may be used to highlight this investigation's contributions.

1) The highest number of samples, to date, are gathered to develop the database of SCC compressive strength.

2) Nine ML algorithms and their hybridized models using PSO are developed to predict the CS of SCC.

3) The XGB model, in both cases, gives the best performance in comparison with the other ML techniques.

4) The best performances of these XGB and XGB_PSO in this investigation are compared with seven previously published investigations and achieved a higher R^2 value. Thus, the simplicity, practical value, and effectiveness of the proposed approach using the XGB model are confirmed. Furthermore, the two criteria mentioned above for assessing the predictive ML technique are achieved using the XGB model.

3 Database description and analysis

Any model's predictive performance depends on many factors, such as the completeness of the training data, the amount of data, and the relationship between input and output variables. In this paper, the database with 1287 experimental data is constructed from 44 published literature [7,43–84]. The total collected database is randomly divided into 2 parts; 70% (901 samples) provide the training and testing data to build the models. 30% (386 samples) of data is used to test the model. The input space includes 13 parameters (denoted from X_1 to X_{13}), which are directly related to the the CS of SCC. A summary of the inputs, output with their range and statistical analysis (average, standard deviation, minimum, Q25, median, Q75, maximum, and skewness) is provided in Table 1.

As illustrated in Fig. 1, the cement content varies from about 83 to 670 (kg/m^3), mainly concentrated in the range from 200 to 450 (kg/m^3). The FA content is 0 to 500 (kg/m^3), with a few samples having about 500 kg/m^3 FA, and about 500 samples without FA. Water content is about 150 to 200 (kg/m^3). Sand or fine aggregate has a broad data spectrum, from 240 to 1200 (kg/m^3), mainly concentrated in the range from 650 to 900 kg/m^3. Similarly, coarse aggregate in the collected data set also has an extensive range of values—the minimum value is 500 kg/m^3, and the maximum value is 1600 kg/m^3. The superplasticizer content is concentrated in the 0 to 25 kg/m^3 range, but most samples range from 1 to 4 kg/m^3. The remaining input data have narrow data spectra (X_7 to X_{12}). GGBFS, silica fume, metakaolin, rice husk ash, viscosity modifying admixtures content are about 1000, 1200, 1100, 1200, 1200, 1100 samples, respectively. There are nine values of curing time, where the minimum value is 0.5 d, and the maximum value is 400 d. Besides, the output parameter (the CS of SCC) has a broad data spectrum, which is in the 1 to 115 (MPa) range. The distribution red lines in Fig. 1 represent the Gaussian Kernel Density Estimation, and each point contributes a Gaussian curve to the total. The outcome is a smooth density estimate obtained from the data that serves as a powerful non-parametric model of the point distribution. The compressive strength values have relatively normal distribution.

The correlation matrix of the inputs and output is plotted in Fig 2. The image is built based on Pearson rank correlation coefficient (r_s) between each pairwise variable. In this image, the correlations between all parameters are clearly and concisely drawn. Different colors show different correlation values. As observed in Fig. 2, some input variables are significantly correlated, like X_1 and X_6, X_4 and X_{12}. However, in this paper, all input variables are considered and cover as many SCC types as possible.

Table 1 Summary of the inputs and output

variable	notation	unit	count	average	Std[a)]	min	Q25%	median	Q75%	max	Skw[b)]
cement	X_1	kg/m^3	1287	353.44	114.69	83.00	250.00	350.00	449.00	670.00	0.11
FA	X_2	kg/m^3	1287	102.50	105.40	0.00	0.00	96.00	173.25	525.00	0.80
water	X_3	kg/m^3	1287	183.85	26.96	126.00	166.12	178.50	196.23	331.50	1.19
sand	X_4	kg/m^3	1287	812.18	153.12	240.00	742.00	820.00	891.00	1180.00	-0.81
coarse aggregate	X_5	kg/m^3	1287	821.30	162.16	500.00	746.00	837.00	900.00	1600.00	1.53
superplasticizers	X_6	kg/m^3	1287	5.70	4.37	0.00	2.41	4.31	8.80	22.50	1.09
limestone powder	X_7	kg/m^3	1287	28.42	64.53	0.00	0.00	0.00	0.00	376.00	2.61
GGBS (kg/m^3)	X_8	kg/m^3	1287	17.22	55.20	0.00	0.00	0.00	0.00	440.00	4.03
silica fume	X_9	kg/m^3	1287	4.06	13.98	0.00	0.00	0.00	0.00	82.50	3.83
metakaolin	X_{10}	kg/m^3	1287	1.41	8.73	0.00	0.00	0.00	0.00	82.50	7.18
rice husk ash	X_{11}	kg/m^3	1287	1.48	13.55	0.00	0.00	0.00	0.00	200.00	11.59
viscosity modifying admixtures	X_{12}	kg/m^3	1287	0.16	0.51	0.00	0.00	0.00	0.00	4.46	4.22
curing time	X_{13}	d	1287	40.40	63.93	0.50	7.00	28.00	28.00	400.00	3.98
compressive strength	CS	MPa	1287	48.62	23.95	1.20	30.00	47.40	66.90	113.10	0.17

Notes: a) Std: Standard deviation; b) Skw: Skewness.

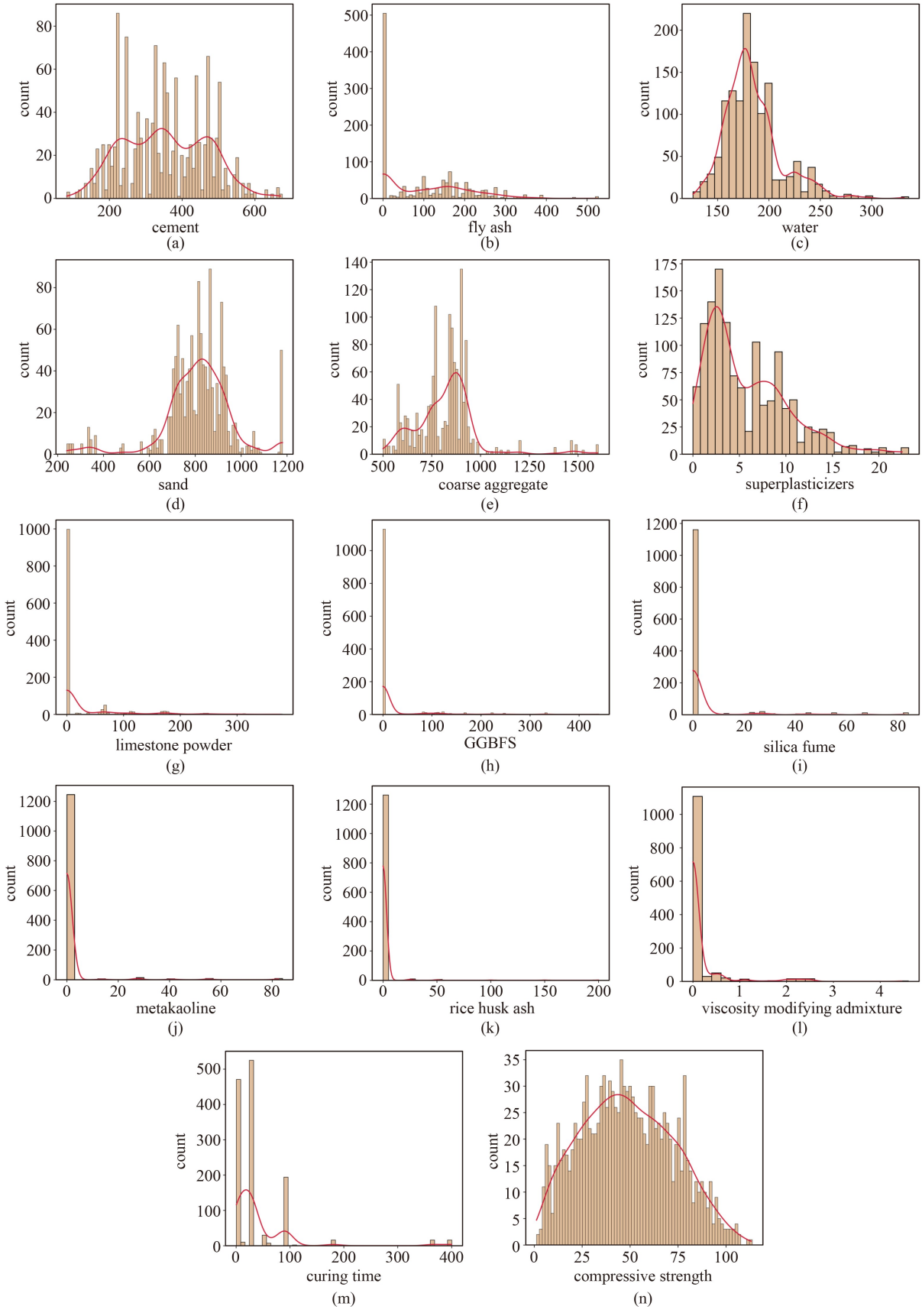

Fig. 1 Histograms of the considered variables of ML models. (a) X_1; (b) X_2; (c) X_3; (d) X_4; (e) X_5; (f) X_6; (g) X_7; (h) X_8; (i) X_9; (j) X_{10}; (k) X_{11}; (l) X_{12}; (m) X_{13}; (n) CS.

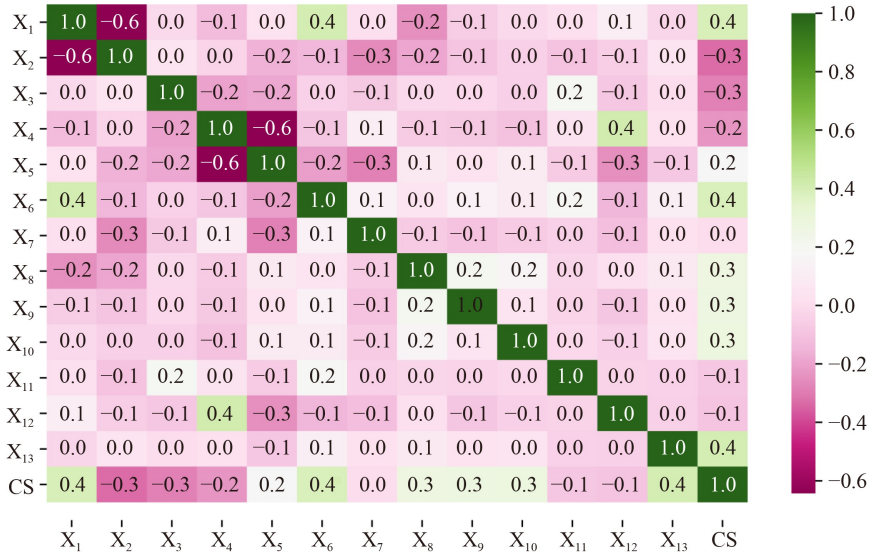

Fig. 2 Correlation matrix of the inputs and CS of SCC.

4 Machine learning methods

ML is a subfield of soft computing that is able to automatically learn knowledge from data without the need for explicit, detailed programming. The basis of ML algorithms is that they can access data, then use that data for self-learning. The main goal of ML is to allow computers to quickly learn about the process automatically in the absence of human knowledge to adjust actions efficiently. ML algorithm generates an inference function with the correct prediction of output values. To perform the desired conversion from input to output, different models can be used. Many ML algorithms have been proposed to find a relationship between inputs and the output. ML algorithms use different models/techniques to perform the learning process and represent knowledge. Nine algorithms have been applied in this work to estimate the CS of SCC.

4.1 Linear regression

LR is the most basic ML model used to develop the relationship between dependent and independent variables [85]. LR consists of finding the best fit representing the relationship of more than one variable in a straightforward term. This best fit line can be achieved by a minimization process of the squared distance between variables in the input space to that line.

4.2 K-Nearest Neighbors

KNN is a supervised-learning algorithm and one of the most straightforward models [85]. The principle of KNN for a regression problem is that the desired output is equal to that of the nearest sample in the data space; alternatively, the algorithm can calculate the average values of outputs corresponding to the nearest data points, or can deduce a relationship using the distance to the nearest data points. Basically, the KNN algorithm tries to find the desired output based on the information of K closest (by value) samples in the training dataset, regardless of whether some of these closest data points are noisy.

4.3 Support Vector Machine

SVM, which is also a supervised-learning algorithm, is one of the most commonly used algorithms nowadays for classification or regression problems. SVM was proposed by Vladimir N. Vapnik and his colleagues [86] in 1963 in Russia and then became popular in the 90s thanks to the application of solving nonlinear problems by the Kernel trick method. The idea of SVM is to find a hyperplane to separate the data points. This hyperplane will divide the space into different regions, and each domain will contain a data type. One of the fundamental notions behind SVMs is the optimum separating hyperplane. It gives birth to the so-called support vectors, which are the data points on the hyperplane's margin border. ML theory has shown that such a hyperplane minimizes the error limit.

4.4 Decision Trees

The decision tree algorithm, proposed by Quinlan [87], is a supervised ML model which can be applied to solve classification or regression problems. DTR builds a model that predicts the value of the target variable using data-driven decision rules. The root, internal, and leaf nodes, as well as branches, comprise the structure of a decision tree. The term "leaf nodes" refers to the leaves that hold the value of the previous category variable, and branch is the branching rule representing the relationship between the independent and target variables' values.

4.5 Random Forest

RF was firstly proposed by Breiman [88], using multiple classifications or regression trees in a group. RF is one of the algorithms built based on the decision tree model, and each tree acts as a vote as the basis of decision-making for the algorithm. Thus, group learning methods combined with the individual results of each tree often yield better results. RF is an extended algorithm based on bagging or bootstrap aggregation that uses random (repetitive) training data samples to generate multiple regression trees without pruning and is the sum of their averages.

4.6 Gradient Boosting

GB is a synthesis algorithm, taking advantage of boosting methods to construct an advanced prediction tool inspired by gradient descent [89]. Boosting begins with a tree's construction to find the relationship between input and output variables, and subsequent trees are next developed to reduce errors. GB considers the boosting problem an optimization one, in which a loss function is used to minimize error.

4.7 Extreme Gradient Boosting

XGB is an upgraded algorithm from the GB algorithm developed by Friedman [90] in 2000. The basic principle used in the GTB algorithm is to combine weak (i.e., high error) essential learning trees into a more substantial learning model tree in a sequential fashion. In order to improve the GTB model's performance, a component is added to the loss function in the XGB model, called regularization, to evaluate the model's complexity. By including a regularization component, it is possible to harmonize the learning model's parameters and prevent overfitting. The core problem of this algorithm is to optimize the value of the objective function. Furthermore, it implements ML algorithms in a gradient-enhanced framework. As a result, XGBoost can handle many data science problems quickly and accurately with parallel boost trees.

4.8 Artificial Neural Network (ANN_LBFGS and ANN_SGD)

ANN is the most popular ML model built on the basis of biological neural networks. It consists of input neurons multiplied by weights, which are the strengths of the corresponding signals, and then calculated by the neuron's activation function. Finally, one more function (possibly an identity function) will compute the desired output. By adjusting the artificial neuron's weight, it is possible to obtain the output corresponding to specific inputs. However, when there are too many neurons, it becomes difficult to calculate these weights. At this point, several approaches are developed to adjust the neurons' weights to achieve the output as quickly as possible. In this study, two different weight updating methods are evaluated, namely the Limited-Memory Broyden-Fletcher-Goldfarb-Shanno method (denoted as LBFGS) and the Stochastic Gradient Descent (SGD).

LBFGS is a training algorithm that belongs to the family of quasi-Newtonian methods, allowing an approximation of the standard BFGS algorithm using the computer's limited memory [91]. Thus, the LBFGS algorithm is advantageous with large datasets because it requires less computational memory, especially while estimating the inverse of the Hessian matrix to drive the transform space search.

SGD is a straightforward variation of the gradient descent technique that computes the slope and updates the weight matrix W in subsets of the training set, rather than in the complete training set [92]. Although this algorithm is highly stable during training, the results obtained are comparable to other algorithms. The idea of the algorithm is quite simple, which is to "calculate the gradient value of each parameter, and take a small step in the direction of the gradient". If this process is repeated many times and randomly selects (stochastically) a batch in the training set, then the model will lead to faster convergence with no adverse effect on loss and correct classification.

4.9 Particle Swarm Optimization (PSO)

PSO is a commonly used and robust algorithm, proposed by Eberhart and Kennedy, based on the behavior of swarms of animals [93]. PSO can handle multiple optimization problems, with each instance representing a possible solution. To find the optimal solution, PSO performs the following steps: 1) initialization; 2) update of velocity; 3) update of instance location; 4) update of the "global best" and "local best" positions if a new instance finds a better position; and 5) check of the termination condition. If the stopping criteria are satisfied (that is, if the error is minimal), the search stops and returns the "globally optimal" and "locally optimal" values corresponding to the position with the least error. If not, the algorithm goes back to step (ii) and continues to execute the loop until the stopping criterion is satisfied.

The details of PSO algorithm is presented in an investigation by Liu et al. [94]. Particle flying speed is dynamically modified based on individual and community flight experience. The formula for the particle swarm algorithm is as follows:

$$V_{\mathrm{id}}^{t+1} = wV_{\mathrm{id}}^{t} + c_1 r_1 \left(p_i^{\mathrm{d}} - x_i^{t}\right) + c_2 r_2 \left(p_{\mathrm{g}}^{\mathrm{d}} - x_i^{t}\right), \tag{1}$$

$$x_i^{t+1} = x_i^{t} + V_{\mathrm{id}}^{t+1}, \tag{2}$$

where x_i^t is the present location of particle i, p_i^d is the superior position of particle i which searches at current, and p_g^d is the superior position of the whole particle swarm. w is a positive constant known as the inertia factor, which decreases linearly with iteration; c_1 and c_2 are positive constants known as the cognitive weights and social weights. r_1 and r_2 are random values in the range [0,1]; $V_{id} \in [-V_{max}, V_{max}]$ with V_{max} is a positive constant. The chosen criterion for terminating iteration is the maximum number of iterations or the best position of particle found so far that fulfills the adaptive thresholds significance. The velocity change of PSO in Eq. (1) has three components: inertia factor, cognitive weight, and social weight. It decides the PSO's performance and how to balance it. According to Blanke [95], the values of w, c_1 and c_2 are set to be equal to 0.4, 0.7, and 0.7, respectively.

4.10 K-Fold Cross-Validation

K-Fold Cross-Validation (K-Fold CV) is a method of breaking down datasets to verify the performance of ML models [96]. It is often used to compare and select the best model for a problem. In this strategy, the crucial parameter is k, which specifies the number of groups into which the data will be divided. When the value of k is chosen, that value is used directly in the name of the evaluation method.

This technique usually includes the following steps.

Step (i): random shuffle of the dataset.

Step (ii): division of the dataset into k groups.

Step (iii): consideration of each group:

1) use of the current group to evaluate model performance;

2) use of the remaining groups to train the model;

3) model training;

4) evaluation of the model;

Step (iv): synthesis of the effectiveness of the model based on the evaluation data.

The aggregated results are usually the average of the evaluations. The addition of variance and standard deviation information to the aggregate results is also used in practice.

4.11 SHAP values

SHapley Additive Explanations (SHAP) are similar to many other sensitivity analysis techniques [97], introduced by Shapley based on cooperative game theory in 1953 [98]. Strumbelj and Kononenko [99] were the first to apply the approach to ML problems in the subsequent years [100]. Since the publication of Lundberg and Lee's study [101] and the associated Python package for SHAP, SHAP has received extensive usage.

The main idea of SHAP is to determine the importance of a given feature, for instance, X. In order to do this, first, all subsets of features in the input space that do not contain X are collected. In a second step, the effect on prediction results while adding X to all the earlier subsets are evaluated. In the final step, all the contribution are aggregated, and the contribution of feature X is computed.

4.12 Performance indices of models

The ML models' prediction performance is evaluated by several well-known and commonly used statistical measures such as root mean square error (*RMSE*), the coefficient of determination (R^2), and mean absolute error (*MAE*). In particular, R^2 is an essential criterion in regression analysis, computed by the square of the correlation (R^2) between the predicted and actual outcome, ranging from 0 to 1 (Eq. (3)). Thus, a high R^2 value shows a good correlation between the model's output and actual values. *RMSE* is an error measurement of the mean squared difference of a ML model's predicted and actual outputs (Eq. (4)), while *MAE* measures the average error (Eq. (5)) [102]. Lower values of these indicators show better prediction performance [103]. The values of these measures are written:

$$R^2 = 1 - \left[\frac{\sum_{j=1}^{N} (M_j - Q_j)^2}{\sum_{i=1}^{N} M_j^2} \right], \tag{3}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (M_j - Q_j)^2}, \tag{4}$$

$$MAE = \frac{1}{N} \sum_{j=1}^{N} \left| M_j - Q_j \right|, \tag{5}$$

with N the data point number in the database, M is the actual value of the output, and Q is the predicted value calculated by the model.

5 Methodology flowchart

This work proposes four main steps in the overall methodology flowchart to predict the CS of SCC as follows (Fig. 3).

Step (i): Data collection.

In the first step, the SCC database consisting of 1287 data points on the CS of SCC is collected from 44

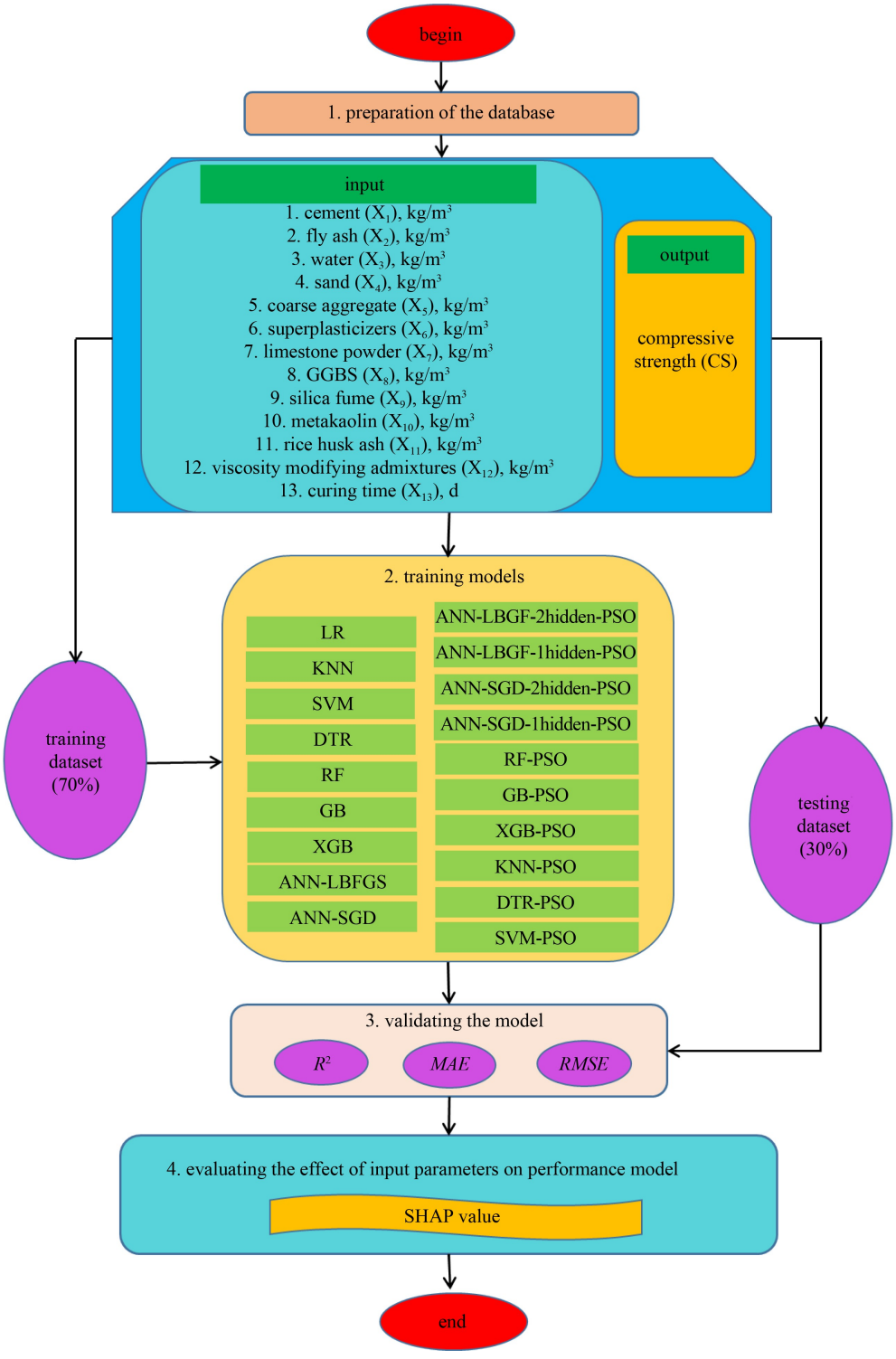

Fig. 3 The present study's methodology flowchart.

published literature. This dataset is divided into two parts: the training and testing datasets. Precisely, 70% of the samples are used to train the ML models, whereas the remaining 30% of data is used for the testing phase of the ML models.

Step (ii): Training models.

The training process with nine ML models is conducted in this second step, namely LR, KNN, SVM, DTR, RF, GB, XGB, ANN_LBFGS, and ANN_SGD. In addition, the PSO algorithm is also used to finely tune the hyperparameters of these models. For the sake of clarification, these ML models are divided into two groups: the first one consists of non-hybrid ML models, and the second one corresponds to hybrid ML models

(using the PSO optimization algorithm). This second step is performed until all the models are successfully trained with tolerance error criteria lower than a predefined threshold. 70% of dataset is used for tunning hyperparameters by the PSO algorithm combined with K-Fold CV technique.

In both steps (ii) and (iii), the prediction performance of the models is evaluated by the statistical measures which are previously presented (i.e., R^2, *RMSE*, and *MAE*).

Step (iii): Validating the model.

Next, the nine proposed ML algorithms are evaluated using Monte Carlo simulations (MCS). In this step, the split 70/30 of dataset is performed multiple times.

Step (iv): Sensitivity analysis.

In this step, the importance and influence of the input variables are evaluated and simulated by SHAP values. After analyzing and evaluating the influence of input variables on the CS of SCC and predictive performance models, less important input variables are removed. Finally, the best model is applied to predict the CS of SCC.

6 Results and discussion

6.1 Assessment of non-hybrid machine learning models

The prediction results obtained by non-hybrid ML models are presented in this section. To fully evaluate the performance of ML models, 500 simulations are conducted, taking 70% of data randomly to construct the training dataset. Figure 4 shows the prediction performance on the training and testing datasets of 9 non-hybrid ML algorithms based on R^2, *RMSE*, and *MAE* over 500 simulations. Typically, the predictive performance of ML models is often reflected by the statistical criteria applied to the testing dataset. Therefore, the R^2, *RMSE*, and *MAE* values that correspond to the testing dataset are used for the sake of comparison. For example, based on Fig. 4(a), the XGB model has the highest R^2 value (R^2 = 0.95), followed by RF and GB models (R^2 values are 0.92 and 0.91, respectively). The remaining models obtain R^2 values in descending order are DTR, KNN, LR, ANN_LBFGS, ANN_SGD, and SVM. According to *RMSE* and *MAE*, Figs. 4(b) and 4(c), the performance of the models in descending order can be seen to be XGB, FR, GB, DTR, KNN, LR, ANN_LBFGS, ANN_SGD, and SVM. Based on the evaluation of these statistical criteria, the XGB model is better than the remaining models for predicting the CS of SCC. Summary of performance values for XGB over 500 simulations are presented in Table 2 with min, max, average, and standard deviation of statistical measures values.

6.2 Assessment of hybrid machine learning models

This section presents the predictions of hybridized ML models using the particle swarm optimization algorithm. Specifically, 10 hybrid ML models are developed, namely KNN_PSO, SVM_PSO, RF_PSO, GB_PSO, XGB_PSO, Decision Tree_PSO, ANN_LBFGS_1hidden_PSO, ANN_LBFGS_2hiddens_PSO, ANN_GD_1hidden_PSO, and ANN_GD_2hiddens_PSO. In all cases, the ANN model is built with two structural possibilities using 1 and 2 hidden layers. The number of neurons in each hidden layer varies from 1 to 15. The number of training iterations to adjust the parameters and the "K" value in K-fold CV are essential parameters that control the performance of ML models. Several studies have shown that K = 5, or a maximum of 10, is sufficient to evaluate the performance of ML models [104,105]. Therefore, 5-Fold CV is selected in this study.

Using the computing hardware configuration as Intel(R) Core (TM) i7-6820HQ CPU 8M cache, up to 3.6 GHz, RAM 32 GB, the optimization process is performed by CPU. The computational time of each hybrid algorithm after 300 iterations is shown in Table 3. The computational time of the hybrid XGB_PSO, 266.40 s, is highest; the lowest computational time belongs to the hybrid KNN_PSO model with 1.14 second.

The models' performances are evaluated using the optimal number of iterations, and the coefficient of determination R^2. Figure 5 shows the mean of R^2 of 10 hybrid ML models with 5-fold cross-validation and 300 iterations. It's worth noting that mean values are a critical parameter for determining the correctness and stability of ML models. It is observed that after about 100 iterations, the mean values of R^2 of most of the proposed hybrid ML models achieve convergence. The Decision Tree_PSO model gives the convergence result of R^2 after about 250 iterations. This result shows that the selection of 300 iterations to evaluate the models' performance is reasonable. Figure 5 shows that GB-PSO and XGB-PSO models are the two models that performed the best among the proposed models. Next, the predictive performance of the hybrid ML models, listed in descending order, is ANN_LBFGS_2 hiddens_PSO > Decision Tree_PSO > ANN_SGD_2 hiddens_PSO > SVM_PSO > ANN_LBFGS_1 hidden_PSO > KNN_PSO > ANN_SGD_1 hidden_PSO > RF_PSO. Therefore, the following paragraphs and accompanying figures are dedicated to evaluating the reliability of the GB_PSO and XGB_PSO models.

Thanks to PSO, the hyperparameters of GB are n_estimators = 370, learning rate = 0.07368, max depth = 18, max features = 4, min samples leaf = 0.0722, min samples split = 0.0261 and the hyperparameters of XGB are n_estimators = 597, learning rate = 0.4587, max depth = 3 in this study. The performance of the GB_PSO and XGB_PSO models using these hyperparameters is

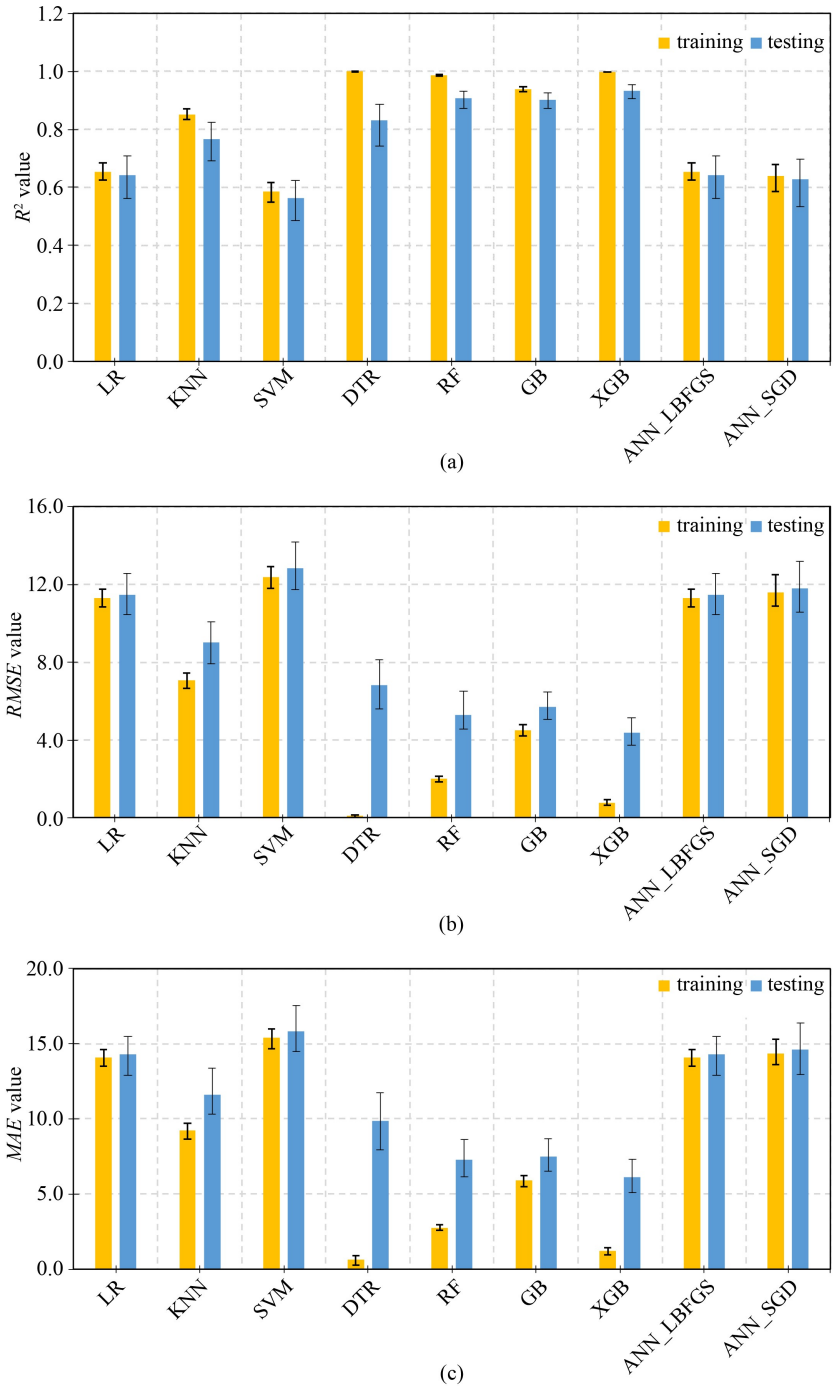

Fig. 4 Performance of different ML models over 500 simulations ML models. (a) R^2 value; (b) *RMSE* value (MPa); (c) *MAE* value (MPa).

Table 2 Summary of performance value for XGB on the training and testing datasets

statistical value	training			testing		
	R^2	*RMSE*	*MAE*	R^2	*RMSE*	*MAE*
min	0.9965	0.6511	0.9771	0.9048	3.7658	5.1078
average	0.9975	0.7811	1.1992	0.9339	4.3723	6.1432
max	0.9984	0.9388	1.4122	0.9536	5.1425	7.3213
Std	0.0003	0.0453	0.0777	0.0089	0.2421	0.3932

evaluated in detail using statistical criteria and MCS. Among the factors affecting ML model performance, the random sampling effect for both training and testing datasets has been shown to have a significant influence [106]. Furthermore, assessing the reliability of the proposed model should be performed with a sufficient number of simulations to make the obtained results more representative [107]. Therefore, 500 MCS for each model are performed, and the results are shown in Fig. 6. The R^2 values of the simulations for the two models are shown in

Table 3 Computational time in second of hybrid algorithms after 300 iterations

ANN (LBFGS) PSO	ANN (LBFGS2) PSO	ANN (SGD1) PSO	ANN (SGD2) PSO	RFPSO	GBPSO	XGBPSO	KNNPSO	DTPSO	SVMPSO
139.23	345.95	8.87	15.97	151.33	120.55	266.40	1.14	3.55	26.34

Note: In second (s).

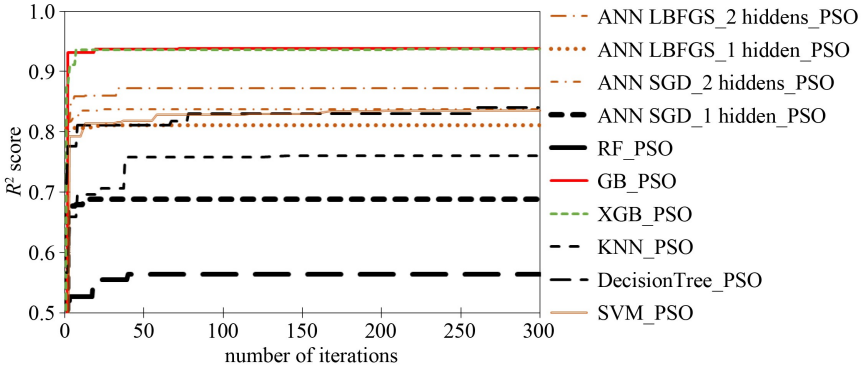

Fig. 5 Mean value of R^2 in function of iterations using hybrid models and 5-Fold CV.

Figs. 6(a) and 6(b) represent the training and testing datasets, respectively. It is observed that the GB_PSO model gives R^2 values in the 0.9526 to 0.9659 range for the training dataset and 0.8781 to 0.9432 for the testing dataset. The most frequented R^2 values are 0.9580 with 240 simulations (training set), and 0.918 with about 248 (testing set). The XGB_PSO model shows the higher accuracy when the R^2 value ranges from 0.9962 to 0.9985, with the highest frequency value R^2 = 0.997 (frequency about 235 times) corresponding to the training dataset. For the testing dataset, R^2 value varies from 0.8883 to 0.9644, and the highest R^2 is 0.946 with 230 simulations. A similar analysis is shown in Figs. 6(c), 6(d) (for *RMSE* values) and 6(e), 6(f) (for *MAE* values) for the two models corresponding to training and testing parts. The *RMSE* and *MAE* of XGB_PSO model are smaller than those of the GB_PSO model, showing that the XGB_PSO model has better performance. Statistical criteria values for the training and testing parts of the GB_PSO model and the XGB_PSO model are detailed in Table 4, which includes the maximum, minimum, mean, and standard deviation. The results show that both GB_PSO and XGB_PSO models are stable under random sampling effect (small standard deviation values) and have a good performance in predicting the CS of SCC. Precisely, the XGB_PSO model gives better results, with the mean value of R^2 = 0.9416, than the GB_PSO (R^2 = 0.9164). The mean values of *RMSE* and *MAE* for XGB_PSO are 4.1363 and 5.7651, respectively, which are lower than for the GB_PSO model (*RMSE* = 5.2119, *MAE* = 6.9033). Overall, the XGB_PSO model shows strong potential in predicting the CS of SCC.

Comparing the performance of the XGB model using default hyperparameters with the performance of the hybrid models XGB_PSO and GB_PSO (cf. Table 4), the predictive capability of three ML models could be ranked as XGB_PSO > XGB > GB_PSO. The performance of XGB_PSO is slightly superior to XGB and GB_PSO. Therefore, the XGB_PSO model using the default hyperparameters can satisfy both prediction accuracy criteria and time-consuming reduction.

6.3 Prediction results of typical ML algorithm

The most accurate hybrid ML model, namely XGB_PSO, is proposed to predict the CS of SCC. Correlation analysis between the actual and output values for the training, testing, and the whole dataset is presented in Figs. 7(a), 7(b), and 7(c), respectively. The correlation lines are found close to the perfect linear line in all cases. This means that there is a coherent correlation between the actual and predicted CS of SCC. Besides, the performance of this model is also evaluated by R^2, *RMSE*, *MAE*. The values of these criteria for the training dataset, the testing dataset, and all datasets are presented in Figs. 7(a), 7(b), and 7(c), corresponding to the XGB_PSO model structure with the best-predicted results. The R^2 value is 0.9970, 0.9644, 0.9861, respectively, for the training, testing, and all dataset. *RMSE* values in training, testing, and all dataset are 1.2472, 4.7801, and 2.8211 MPa, respectively. The *MAE* values are respectively 0.8214, 3.4832, and 1.6218 MPa. These values indicate that the XGB_PSO model can be applied to predict quite accurately the CS of SCC.

Next, the error diagrams of the XGB_PSO model for the training and testing datasets are computed and presented in Figs. 8(a) and 8(b). As can be seen, the error values are small, corresponding to the range of [−5 to 4] MPa for the training dataset and [−18 to 15] MPa for the testing one. The number of samples with prediction errors out of the [−5 to 5] kN range is small (2 samples) for the

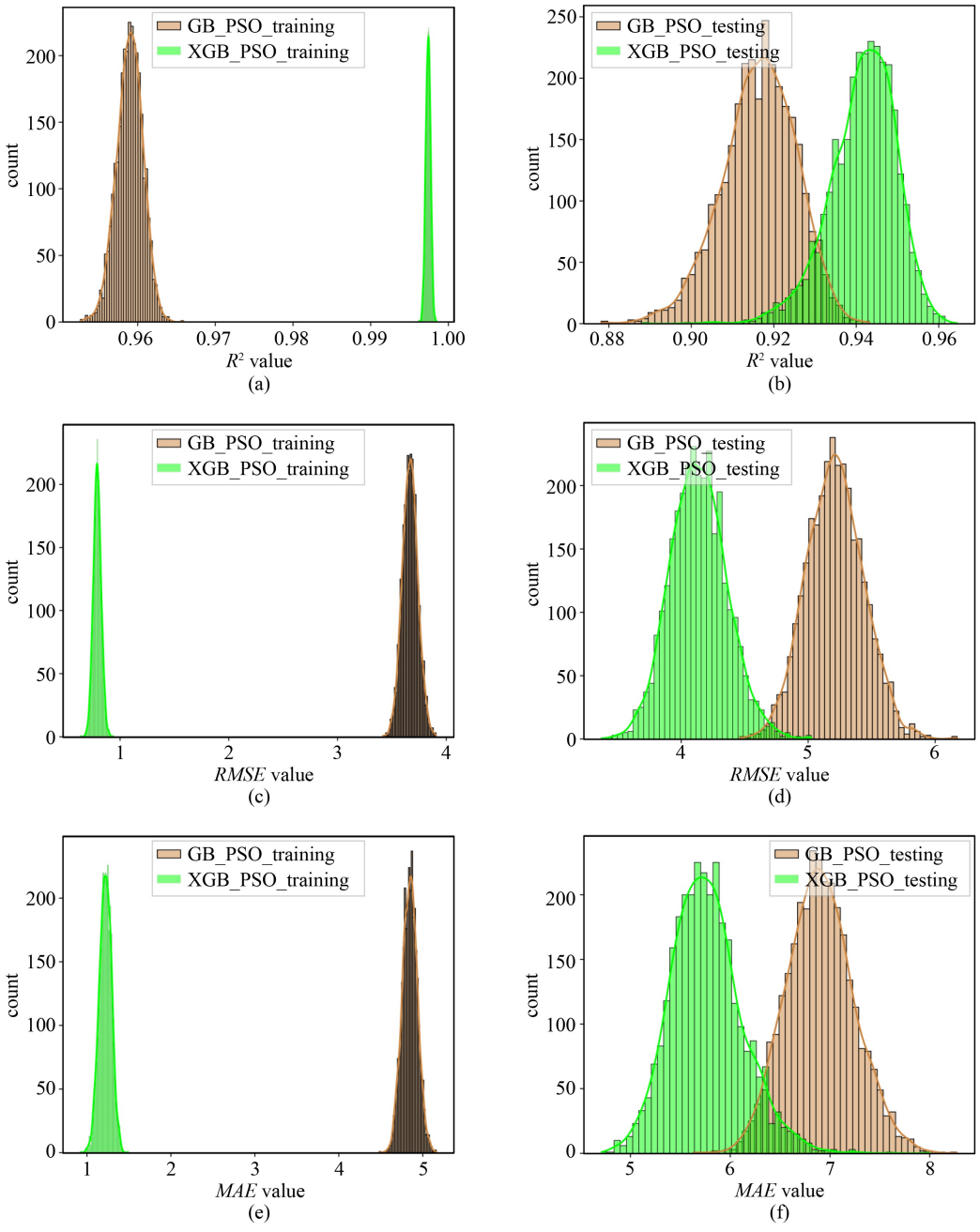

Fig. 6 Histograms showing the results of 500 MCS for GB-PSO and XGB-PSO. (a) R^2 value for training dataset; (b) R^2 value for testing dataset; (c) *RMSE* for training dataset; (d) *RMSE* for testing dataset; (e) *MAE* for training dataset; (f) *MAE* for testing dataset.

Table 4 Summary of performance value for two hybrid models GB-PSO and XGB-PSO

ML model	Statistical value	training part			testing part		
		R^2	*RMSE*	*MAE*	R^2	*RMSE*	*MAE*
GB_PSO	min	0.9526	3.4179	4.4817	0.8781	4.4611	5.6289
	average	0.9590	3.6674	4.8454	0.9164	5.2119	6.9033
	max	0.9659	3.9058	5.1605	0.9432	6.1726	8.2714
	Std	0.0017	0.0710	0.0901	0.0090	0.2250	0.3379
XGB_PSO	min	0.9962	0.6412	0.9269	0.8883	3.3736	4.7113
	average	0.9974	0.7904	1.2192	0.9416	4.1363	5.7651
	max	0.9985	0.9348	1.4880	0.9644	5.0243	7.9969
	Std	0.0003	0.0357	0.0714	0.0080	0.0714	0.3586

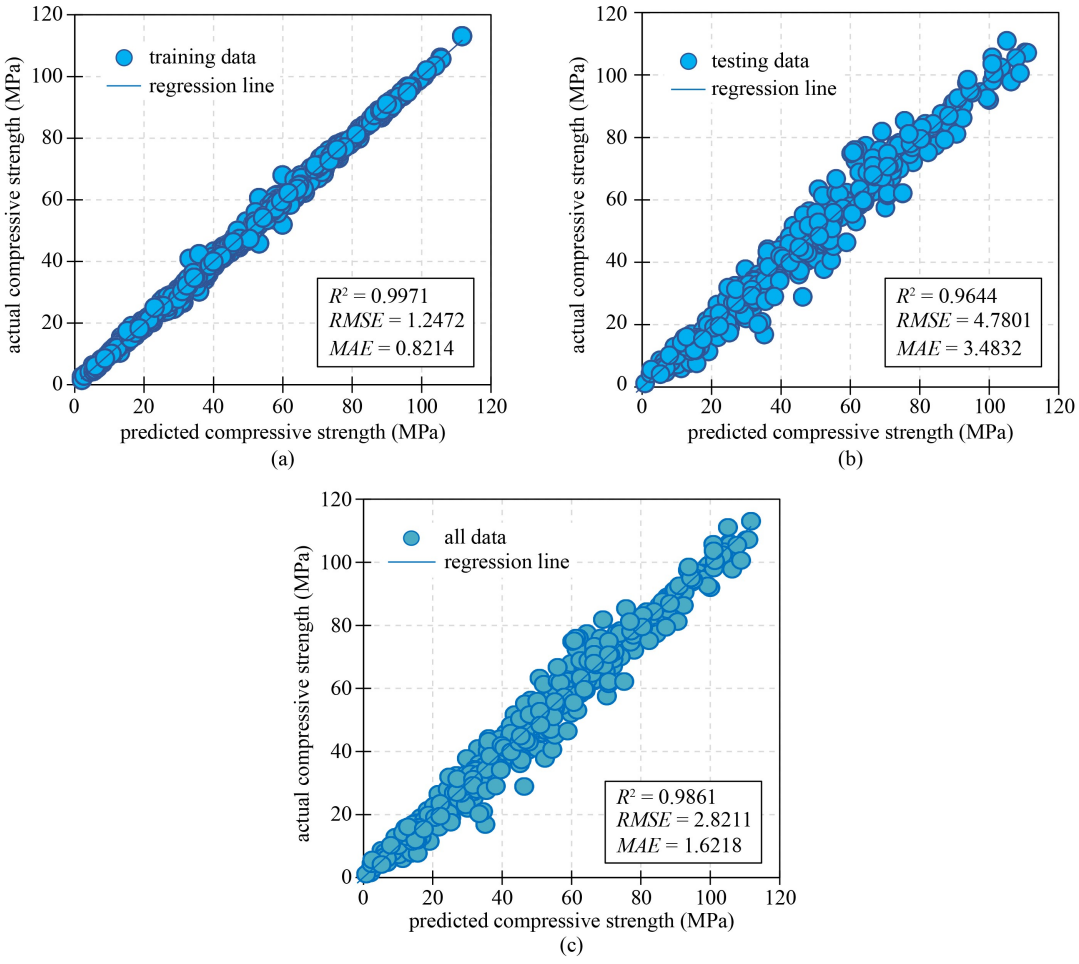

Fig. 7 Correlation graphs between experimental and predicted CS: (a) training dataset; (b) testing dataset; (c) all data.

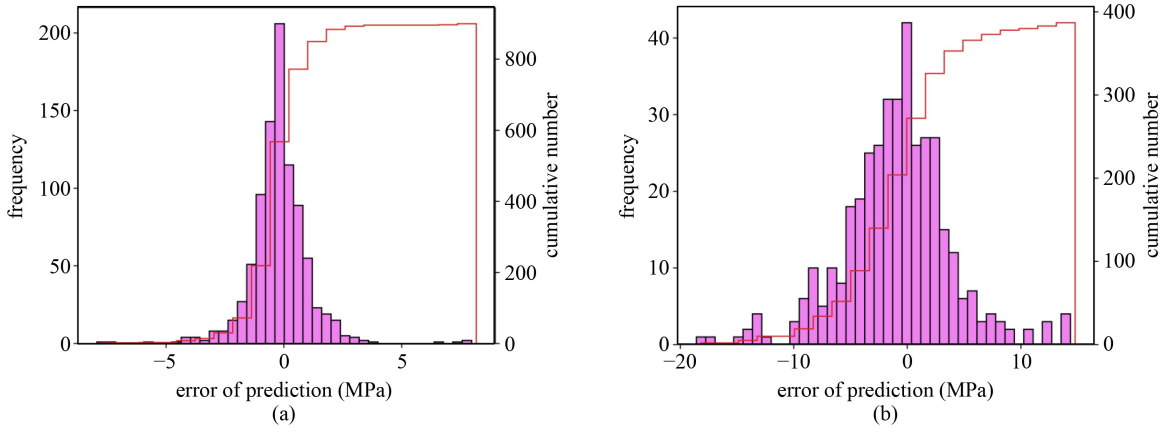

Fig. 8 Comparison between predicted and experimental CS for (a) training process and (b) testing process.

training process. Besides, the cumulative red lines show that about 800 samples with errors are in the range [−2 to 2] MPa for the training process, whereas about 300 samples with errors are within the [−5 to 5] MPa for the testing process.

In order to clarify the significance of this study, Table 5 compares different ML techniques used in previously published works with the best ML models XGB_PSO and XGB proposed herein. The comparisons are based on several criteria such as the number of inputs used, number of data, and performance measure evaluated by R^2 value for the testing process. The comparison shows that the XGB model of this investigation, using only the default hyperparameters, could give almost the highest performance with the highest number of samples in the database. The performance values in Asteris and Kolovos

Table 5 Performance comparison with published works in predicting the CS of SCC

reference	ML algorithm	input variable	sample size	statistical measures
Siddique et al. [29]	ANN	6 inputs: cement, FA, water to powder ratio, superplasticizer, sand, coarse aggregate	80	$R^2 = 0.9187$
Asteris and Kolovos [32]	ANN	11 inputs: cement, coarse aggregate, fine aggregate, water, limestone powder, FA, ground granulated BFS, silica fume, rice husk, new generation superplasticizers, viscosity modifying admixtures	205	$R^2 = 0.9658$
Asteris et al. [38]	ANNs, BPNN	11 inputs: cement, limestone powder, FA, ground granulated BFS GGBFS, silica fume, rice husk ash, coarse aggregate, fine aggregate, water, superplasticizer, viscosity modifying agent	169	$R^2 = 0.9655$
Akkurt et al. [39]	ANN, FL	11 inputs: cement, limestone powder, FA, ground granulated BFS, silica fume, rice husk ash, coarse aggregate, fine aggregate, water, superplasticizer, viscosity modifying agent	169	$R^2 = 0.9604$
Kovačević et al. [40]	RF regression	7 inputs: water to binder ratio, macro-synthetic polypropylene fibers, steel fiber, scoria, crumb rubber, natural fine aggregate, natural coarse aggregate	131	$R^2 = 0.9477$
Azimi-Pour et al. [41]	Linear and nonlinear SVM	10 inputs: cement, water to cement, water to powder, water to binder, fine aggregate to powder, coarse aggregate to powder, high water range reducer to powder, viscosity modify admixture to powder, micro silica to binder.	340	$R^2 = 0.9388$
Saha et al. [42]	Support vector regression approach	6 inputs: binder content, FA percentage, water to powder ratio, fine aggregate (f), coarse aggregate, superplasticizer dosage	115	SVR: $R^2 = 0.955$ ANN: $R^2 = 0.882$
this study	LR, KNN, SVM, DTR, RF, GB, XGB, ANN_LBFGS and ANN_SGD, and the Hyperparameters tuning with PSO	13 inputs: cement, FA, water, sand or fine, coarse aggregate, superplasticizer, limestone powder, ground granulated BFS, silica fume, metakaolin, rice husk ash, viscosity modifying admixtures, curing time	1280	XGB_PSO: $R^2 = 0.9644$ XGB: $R^2 = 0.9536$

[32], and Asteris et al. [38] are higher, but the number of samples in these works is much lower than in this investigation. However, the performance of XGB_PSO is relatively equivalent to that of these investigations. The evaluation of CS of SCC in this study is more accurate than the values obtained in these previous works. Based on the simulations performed, this work shows that the XGB model using default hyperparameters has a higher prediction accuracy. Overall, these results indicate XGB models, in both non-hybrid and hybrid variations, can be applied as a quick prediction tool for material engineers.

6.4 Effect of input number on performance typical model

In this section, the importance and influence of the input variables are evaluated by the SHAP value implanted in Python programming language (Fig. 9). Observing the SHAP value, the X_3 input corresponding to water content is the most important feature, and X_3 input negatively influences the CS of SCC. It shows that with a higher water content, the CS of SCC decreases. That is also confirmed by other investigations, such as Oner and Akyuz [108], Shen and Xu [109], Zhou et al. [110]. Similarly, the results show that the coarse aggregate, sand, and superplasticizers, denoted as X_5, X_4, and X_6, respectively, have the most crucial positive impact on the CS. The sand content also positively influences the CS of SCC. In fact, with a higher sand content, the CS of SCC is improved. Superplasticizer content also has a positive impact on the CS of SCC. However, a higher superplasticizer content seems to decrease the CS of SCC. Lastly, among 13 input variables, the variables that have a negligible impact on the CS according to the analysis are proportions of silica fume, limestone powder,

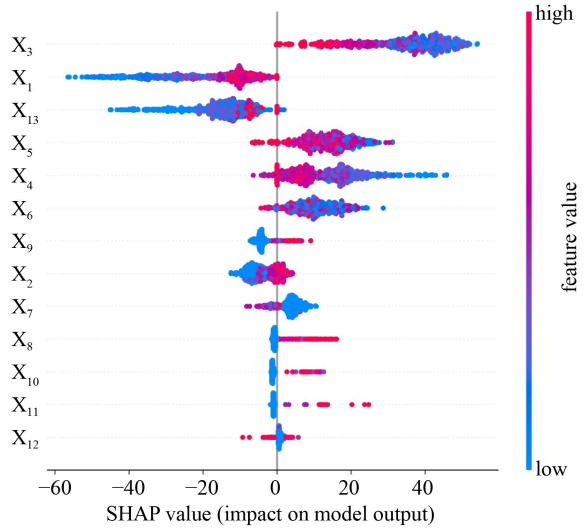

Fig. 9 Feature importance analysis presented by SHAP value.

abrasive slag, metakaolin, rice husk ash, viscosity modifying admixtures, denoted as X_9, X_7, X_8, X_{10}, X_{11}, X_{12}, respectively. After analyzing and evaluating the influence input variables on CS of SCC and predictive performance of the model, less important input variables are removed. Simulations with XGB_PSO are conducted without the above-mentioned input variables. The performance of the model for these cases is presented in Table 6. It is shown that XGB_PSO model using 6 input variables including the five most important variables, namely proportions of cement, FA, water, sand, coarse aggregate and curing time, could predict the CS of SCC with high accuracy, with $R^2 = 0.9515$. In another attempt, it is shown that if the database has only three importance input variables such as X_1, X_3, and X_{13} (cement, water

Table 6 Performance of the XGB model over the number of inputs

input	training part			testing part		
	R^2	*RMSE*	*MAE*	R^2	*RMSE*	*MAE*
X_1, X_3, X_4, X_5, X_6, X_{13}	0.9935	1.8764	1.0218	0.9450	5.9407	4.1603
X_1, X_3, X_4, X_5, X_{13}	0.9907	2.2474	1.2671	0.9434	6.0281	4.3440
X_1, X_3, X_5, X_{13}	0.9877	2.5847	1.5112	0.9429	6.0573	4.5451
X_1, X_3, X_{13}	0.9676	4.1970	2.4100	0.9253	6.9245	5.0681
X_1, X_2, X_3, X_4, X_5, X_{13}	0.9917	2.1191	1.1322	0.9515	5.5780	4.0643

and curing time, respectively), the XGB_PSO model can also provide good prediction of SCC compressive strength with R^2 = 0.9253 for the testing part. Several other scenarios are also shown in Table 6.

7 Conclusions

In this study, the XGB algorithm was investigated to determine its validity for prediction of the CS of SCC from mix design and curing age. A high-quality and reliable dataset with the highest number of data points and broadest data range from the literature was used for the evaluation of the XGB model. The comparison of performance with nine other ML methods showed the best performance of the XGB model in predicting SCC's CS. Furthermore, the XGB model using the default hyperparameters predicted the CS of SCC with high performance and reliability, only slightly lower than performance and reliability of the hybrid model XGB_PSO. Therefore, the XGB method, including a hybrid or non-hybrid model, is faster to train, more accurate, and more robust than other ML methods. Indeed, the XGB model using default parameters, implemented with the XGB package of Python, without any further optimization, could achieve better prediction results than LR, KNN, RF, GB, ANN_LFBGS, ANN_SGD, SVM models.

The feature importance analysis indicated that the most critical features for predicting the CS of SCC are cement, FA, water, sand, coarse aggregate, and curing time. Using the first five important input variables, the XGB_PSO model could predict the CS of SCC with the high accuracy R^2 = 0.9515 for the testing part and gain comparative advantage over methods used in previously published works. Moreover, the prediction of the compressive strength using the XGB_PSO model with just three important inputs, namely cement, water, and curing time, provides reasonably good results with R^2 = 0.9253 for the testing process. From the perspective of this work, finding appropriate processing techniques and using only relevant input variables could be investigated to improve prediction ability.

References

1. EFNARC. Specification and Guidelines for Self-Compacting Concrete. Farnham: European Federation of Specialist Construction Chemicals and Concrete System, 2002
2. Fernandez-Gomez J, Landsberger G A. Evaluation of shrinkage prediction models for self-consolidating concrete. ACI Materials Journal, 2007, 104(5): 464
3. Skarendahl Å, Petersson Ö. Self-Compacting Concrete—State-of-the-Art report of RILEM Technical Committee 174-SCC. Cachan: RILEM Publications, 2000
4. Ramadan K Z, Haddad R H. Self-healing of overloaded self-compacting concrete of rigid pavement. European Journal of Environmental and Civil Engineering, 2017, 21(1): 63−77
5. Busari A A, Akinmusuru J O, Dahunsi B I O, Ogbiye A S, Okeniyi J O. Self-compacting concrete in pavement construction: Strength grouping of some selected brands of cements. Energy Procedia, 2017, 119: 863−869
6. Pasko Jr T J. Concrete pavements—Past, present, and future. Public Roads, 1998, 62(1): 7−15
7. Bouzoubaâ N, Lachemi M. Self-compacting concrete incorporating high volumes of class F fly ash: Preliminary results. Cement and Concrete Research, 2001, 31(3): 413−420
8. Busari A, Akinmusuru J, Dahunsi B. Mechanical properties of dehydroxylated kaolinitic clay in self-compacting concrete for pavement construction. Silicon, 2019, 11(5): 2429−2437
9. Rajah Surya T, Prakash M, Satyanarayanan K S, Keneth Celestine A, Parthasarathi N. Compressive strength of self compacting concrete under elevated temperature. Materials Today: Proceedings, 2021, 40: S83−S87
10. Guo H, Zhuang X, Chen P, Alajlan N, Rabczuk T. Stochastic deep collocation method based on neural architecture search and transfer learning for heterogeneous porous media. Engineering with Computers, 2022, 1−26
11. Nguyen-Thanh V M, Anitescu C, Alajlan N, Rabczuk T, Zhuang X. Parametric deep energy approach for elasticity accounting for strain gradient effects. Computer Methods in Applied Mechanics and Engineering, 2021, 386: 114096
12. Ly H B, Pham B T, Le L M, Le T T, Le V M, Asteris P G. Estimation of axial load-carrying capacity of concrete-filled steel tubes using surrogate models. Neural Computing & Applications, 2021, 33(8): 3437−3458
13. Ly H B, Nguyen M H, Pham B T. Metaheuristic optimization of Levenberg–Marquardt-based artificial neural network using particle swarm optimization for prediction of foamed concrete compressive strength. Neural Computing & Applications, 2021,

33(24): 17331

14. Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh V M, Guo H, Hamdia K, Zhuang X, Rabczuk T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. Computer Methods in Applied Mechanics and Engineering, 2020, 362: 112790
15. Mortazavi B, Silani M, Podryabinkin E V, Rabczuk T, Zhuang X, Shapeev A V. First-principles multiscale modeling of mechanical properties in graphene/borophene heterostructures empowered by machine-learning interatomic potentials. Advanced Materials, 2021, 33(35): 2102807
16. Ly H B, Nguyen T A, Thi Mai H V, Tran V Q. Development of deep neural network model to predict the compressive strength of rubber concrete. Construction & Building Materials, 2021, 301: 124081
17. Goswami S, Anitescu C, Chakraborty S, Rabczuk T. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. Theoretical and Applied Fracture Mechanics, 2020, 106: 102447
18. Rabczuk T, Zi G, Bordas S, Nguyen-Xuan H. A simple and robust three-dimensional cracking-particle method without enrichment. Computer Methods in Applied Mechanics and Engineering, 2010, 199(37−40): 2437−2455
19. Rabczuk T, Belytschko T. Cracking particles: A simplified meshfree method for arbitrary evolving cracks. International Journal for Numerical Methods in Engineering, 2004, 61(13): 2316−2343
20. Rabczuk T, Belytschko T. A three-dimensional large deformation meshfree method for arbitrary evolving cracks. Computer Methods in Applied Mechanics and Engineering, 2007, 196(29−30): 2777−2799
21. Ren H L, Zhuang X Y, Anitescu C, Rabczuk T. An explicit phase field method for brittle dynamic fracture. Computers & Structures, 2019, 217: 45−56
22. Ly H B, Le T T, Vu H L T, Tran V Q, Le L M, Pham B T. Computational hybrid machine learning based prediction of shear capacity for steel fiber reinforced concrete beams. Sustainability (Basel), 2020, 12(7): 2709
23. Nguyen T A, Ly H B, Mai H V T, Tran V Q. Prediction of later-age concrete compressive strength using feedforward neural network. Advances in Materials Science and Engineering, 2020, 2020
24. Quan Tran V, Quoc Dang V, Si Ho L. Evaluating compressive strength of concrete made with recycled concrete aggregates using machine learning approach. Construction & Building Materials, 2022, 323: 126578
25. Topçu İ B, Sarıdemir M. Prediction of compressive strength of concrete containing fly ash using artificial neural networks and fuzzy logic. Computational Materials Science, 2008, 41(3): 305−311
26. Mai H V T, Nguyen T A, Ly H B, Tran V Q. Prediction compressive strength of concrete containing GGBFS using random forest model. Advances in Civil Engineering, 2021, 2021: e6671448
27. Mai H V T, Nguyen T A, Ly H B, Tran V Q. Investigation of ANN model containing one hidden layer for predicting compressive strength of concrete with blast-furnace slag and fly ash. Advances in Materials Science and Engineering, 2021, 2021: e5540853
28. Ly H B, Nguyen T A, Pham B T. Investigation on factors affecting early strength of high-performance concrete by Gaussian Process Regression. PLoS One, 2022, 17(1): e0262930
29. Siddique R, Aggarwal P, Aggarwal Y. Prediction of compressive strength of self-compacting concrete containing bottom ash using artificial neural networks. Advances in Engineering Software, 2011, 42(10): 780−786
30. Abu Yaman M, Abd Elaty M, Taman M. Predicting the ingredients of self compacting concrete using artificial neural network. Alexandria Engineering Journal, 2017, 56(4): 523−532
31. Asteris P G, Kolovos K G, Douvika M G, Roinos K. Prediction of self-compacting concrete strength using artificial neural networks. European Journal of Environmental and Civil Engineering, 2016, 20(sup1): s102−s122
32. Asteris P G, Kolovos K G. Self-compacting concrete strength prediction using surrogate models. Neural Computing & Applications, 2019, 31(S1): 409−424
33. Malagavell V, Manalel P A. Modeling of compressive strength of admixture-based self compacting concrete using fuzzy logic and artificial neural networks. Asian Journal of Applied Sciences, 2014, 7(7): 536−551
34. Zhang J, Ma G, Huang Y, sun J, Aslani F, Nener B. Modelling uniaxial compressive strength of lightweight self-compacting concrete using random forest regression. Construction & Building Materials, 2019, 210: 713−719
35. Azimi-Pour M, Eskandari-Naddaf H, Pakzad A. Linear and non-linear SVM prediction for fresh properties and compressive strength of high volume fly ash self-compacting concrete. Construction & Building Materials, 2020, 230: 117021
36. Pedregosa F, Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 2011, 12: 2825−2830
37. Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York: Association for Computing Machinery, 2016, 785−794
38. Asteris P, Kolovos K, Douvika M, Roinos K. Prediction of self-compacting concrete strength using artificial neural networks. European Journal of Environmental and Civil Engineering, 2016, 20(sup1): s102−s122
39. Akkurt S, Tayfur G, Can S. Fuzzy logic model for the prediction of cement compressive strength. Cement and Concrete Research, 2004, 34: 1429−1433
40. Kovačević M, Lozančić S, Nyarko E K, Hadzima-Nyarko, M. Application of artificial intelligence methods for predicting the compressive strength of self-compacting concrete with class F fly ash. Materials (Basel), 2022, 15: 4191
41. Azimi-Pour M, Eskandari-Naddaf H, Pakzad A. Linear and non-linear SVM prediction for fresh properties and compressive strength of high volume fly ash self-compacting concrete. Construction & Building Materials, 2020, 230: 117021

42. Saha P, Debnath P, Thomas P. Prediction of fresh and hardened properties of self-compacting concrete using support vector regression approach. Neural Computing & Applications, 2020, 32(12): 7995−8010
43. Siddique R. Properties of self-compacting concrete containing class F fly ash. Materials & Design, 2011, 32(3): 1501−1507
44. Sukumar B, Nagamani K, Srinivasa Raghavan R. Evaluation of strength at early ages of self-compacting concrete with high volume fly ash. Construction & Building Materials, 2008, 22(7): 1394−1401
45. Gesoğlu M, Güneyisi E, Özbay E. Properties of self-compacting concretes made with binary, ternary, and quaternary cementitious blends of fly ash, blast furnace slag, and silica fume. Construction & Building Materials, 2009, 23(5): 1847−1854
46. Güneyisi E, Gesoğlu M, Özbay E. Strength and drying shrinkage properties of self-compacting concretes incorporating multi-system blended mineral admixtures. Construction & Building Materials, 2010, 24(10): 1878−1887
47. Dinakar P. Design of self-compacting concrete with fly ash. Magazine of Concrete Research, 2012, 64(5): 401−409
48. Guru Jawahar J, Sashidhar C, Ramana Reddy I V, Annie Peter J. Micro and macrolevel properties of fly ash blended self compacting concrete. Materials & Design, 2013, 46: 696−705
49. Boel V, Audenaert K, De Schutter G, Heirman G, Vandewalle L, Desmet B, Vantomme J. Transport properties of self compacting concrete with limestone filler or fly ash. Materials and Structures, 2007, 40(5): 507−516
50. Jalal M, Mansouri E. Effects of fly ash and cement content on rheological, mechanical, and transport properties of high-performance self-compacting concrete. Science and Engineering of Composite Materials, 2012, 19(4): 393−405
51. Dinakar P, Babu K G, Santhanam M. Mechanical properties of high-volume fly ash self-compacting concrete mixtures. Structural Concrete, 2008, 9(2): 109−116
52. Nehdi M, Pardhan M, Koshowski S. Durability of self-consolidating concrete incorporating high-volume replacement composite cements. Cement and Concrete Research, 2004, 34(11): 2103−2112
53. Uysal M, Sumer M. Performance of self-compacting concrete containing different mineral admixtures. Construction & Building Materials, 2011, 25(11): 4112−4120
54. Venkatakrishnaiah R, Sakthivel G. Bulk utilization of flyash in self compacting concrete. KSCE Journal of Civil Engineering, 2015, 19(7): 2116−2120
55. Hemalatha T, Ramaswamy A, Chandra Kishen J M. Micromechanical analysis of self compacting concrete. Materials and Structures, 2015, 48(11): 3719−3734
56. Liu M. Self-compacting concrete with different levels of pulverized fuel ash. Construction & Building Materials, 2010, 24(7): 1245−1252
57. Bingöl A F, Tohumcu İ. Effects of different curing regimes on the compressive strength properties of self compacting concrete incorporating fly ash and silica fume. Materials & Design, 2013, 51: 12−18
58. Barbhuiya S. Effects of fly ash and dolomite powder on the properties of self-compacting concrete. Construction & Building Materials, 2011, 25(8): 3301−3305
59. Sun Z J, Duan W W, Tian M L, Fang Y F. Experimental research on self-compacting concrete with different mixture ratio of fly ash. Advanced Materials Research, 2011, 236–238: 490–495
60. Pathak N, Siddique R. Properties of self-compacting-concrete containing fly ash subjected to elevated temperatures. Construction & Building Materials, 2012, 30: 274−280
61. Patel R, Hossain K, Shehata M, Bouzoubaâ N, Lachemi M. Development of statistical models for mixture design of high-volume fly ash self-consolidating concrete. ACI Materials Journal, 2004, 101: 294−302
62. Sonebi M. Medium strength self-compacting concrete containing fly ash: Modelling using factorial experimental plans. Cement and Concrete Research, 2004, 34(7): 1199−1208
63. Bui V K, Akkaya Y, Shah S P. Rheological model for self-consolidating concrete. Materials Journal, 2002, 99(6): 549−559
64. Ghezal A, Khayat K. Optimizing self-consolidating concrete with limestone filler by using statistical factorial design methods. ACI Materials Journal, 2002, 99: 264−272
65. Dinakar P, Sethy K P, Sahoo U C. Design of self-compacting concrete with ground granulated blast furnace slag. Materials & Design, 2013, 43: 161−169
66. Felekoğlu B, Türkel S, Baradan B. Effect of water/cement ratio on the fresh and hardened properties of self-compacting concrete. Building and Environment, 2007, 42(4): 1795−1802
67. Gesoğlu M, Özbay E. Effects of mineral admixtures on fresh and hardened properties of self-compacting concretes: Binary, ternary and quaternary systems. Materials and Structures, 2007, 40(9): 923−937
68. Grdic Z, Despotovic I, Toplicic-Curcic G. Properties of self-compacting concrete with different types of additives. Facta Universitatis—Series: Architecture and Civil Engineering, 2008, 6(2): 173−177
69. Güneyisi E, Gesoglu M, Azez O A, Öz H Ö. Effect of nano silica on the workability of self-compacting concretes having untreated and surface treated lightweight aggregates. Construction & Building Materials, 2016, 115: 371−380
70. Memon S A, Shaikh M A, Akbar H. Utilization of rice husk ash as viscosity modifying agent in self compacting concrete. Construction and building materials, 2011, 25(2): 1044−1048
71. Rahman M E, Muntohar A S, Pakrashi V, Nagaratnam B H, Sujan D. Self compacting concrete from uncontrolled burning of rice husk and blended fine aggregate. Materials & Design, 2014, 55: 410−415
72. Sfikas I P, Trezos K G. Effect of composition variations on bond properties of self-compacting concrete specimens. Construction & Building Materials, 2013, 41: 252−262
73. Valcuende M, Marco E, Parra C, Serna P. Influence of limestone filler and viscosity-modifying admixture on the shrinkage of self-compacting concrete. Cement and Concrete Research, 2012, 42(4): 583−592
74. Şahmaran M, Yaman İ Ö, Tokyay M. Transport and mechanical properties of self consolidating concrete with high volume fly ash. Cement and Concrete Composites, 2009, 31(2): 99−106
75. Patel R. Development of statistical models to simulate and optimize self-consolidating concrete mixes incorporating high volumes of fly ash. Thesis for the Master's Degree. Toronto: Ryerson University, 2004

76. Nepomuceno M C S, Pereira-de-Oliveira L A, Lopes S M R. Methodology for the mix design of self-compacting concrete using different mineral additions in binary blends of powders. Construction & Building Materials, 2014, 64: 82−94
77. Krishnapal P, Yadav R K, Rajeev C. Strength characteristics of self compacting concrete containing fly ash. Research Journal of Engineering Sciences, 2013, 2278: 9472
78. Dhiyaneshwaran S, Ramanathan P, Bose B, Venkatasubramani R. Study on durability characteristics of self-compacting concrete with fly ash. Jordan Journal of Civil Engineering, 2013, 7: 342−353
79. Mahalingam B, Nagamani K. Effect of processed fly ash on fresh and hardened properties of self compacting concrete. International Journal of Earth Sciences, 2011, 4(5): 930−940
80. Mahesh S. Self compacting concrete and its properties. International Journal of Engineering Research and Applications, 2014, 4(8): 72−80
81. Al-Rubaye M M. Self-compacting concrete: Design, properties and simulation of the flow characteristics in the L-box. Dissertation for the Doctoral Degree. Cardiff: Cardiff University, 2016
82. Mahato A, Gambhir G, Kumar A, Dutta A, Kisley K. Self compacting concrete. Thesis for the Bachelor's Degree. Bhubaneswar: KIIT University, 2016
83. Seo J, Torres E, Schaffer W. Self-Consolidating Concrete for Prestressed Bridge Girders. WisDOT ID NO. 0092-15-03. 2017
84. Douglas R P, Bui V K, Akkaya Y, Shah S P. Properties of Self-consolidating concrete containing class F fly ash: With a Verification of the minimum paste volume method. Aci Material Journal, 2006, 233: 45−64
85. Mitchell T M. Machine Learning. New York: McGraw-Hill Education, 1997
86. Cortes C, Vapnik V. Support-vector networks. Machine Learning, 1995, 20(3): 273−297
87. Quinlan J R. Induction of decision trees. Machine Learning, 1986, 1(1): 81−106
88. Breiman L. Random forests. Machine Learning, 2001, 45(1): 5−32
89. Ayyadevara V K. Pro Machine Learning Algorithms. Berkeley: Apress, 2018, 117−134
90. Friedman J H. Greedy function approximation: A gradient boosting machine. Annals of Statistics, 2001, 29(5): 1189−1232
91. June L W, Hassan M A. Modifications of the limited memory BFGs algorithm for large-scale nonlinear optimization. Mathematical Journal of Okayama University, 2005, 47(1): 175−188
92. Bottou L. Neural Networks: Tricks of the Trade. Heidelberg: Springer, 2012, 421−436
93. Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Nagoya: IEEE, 1995, 39−43
94. Liu B, Vu-Bac N, Rabczuk T. A stochastic multiscale method for the prediction of the thermal conductivity of Polymer nanocomposites through hybrid machine learning algorithms. Composite Structures, 2021, 273: 114269
95. Blanke S. Hyperactive: An optimization and data collection toolbox for convenient and fast prototyping of computationally expensive models. 2019. Available at the website of GITHUB
96. Stone M. Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society. Series B. Methodological, 1974, 36(2): 111−133
97. Liu B, Vu-Bac N, Zhuang X, Rabczuk T. Stochastic multiscale modeling of heat conductivity of polymeric clay nanocomposites. Mechanics of Materials, 2020, 142: 103280
98. Shapley L S. Quota Solutions of n-Person Games. Belvoir: Defense Technical Information Center, 1953, 343
99. Strumbelj E, Kononenko I. An efficient explanation of individual classifications using game theory. Journal of Machine Learning Research, 2010, 11: 1−18
100. Štrumbelj E, Kononenko I. Explaining prediction models and individual predictions with feature contributions. Knowledge and Information Systems, 2014, 41(3): 647−665
101. Lundberg S M, Lee S I. A unified approach to interpreting model predictions. Advances in neural information processing systems. 2017, 30
102. Ly H B, Le L M, Duong H T, Nguyen T C, Pham T A, Le T T, Le V M, Nguyen-Ngoc L, Pham B T. Hybrid artificial intelligence approaches for predicting critical buckling load of structural members under compression considering the influence of initial geometric imperfections. Applied Sciences (Basel, Switzerland), 2019, 9(11): 2258
103. Dao D V, Trinh S H, Ly H B, Pham B T. Prediction of compressive strength of geopolymer concrete using entirely steel slag aggregates: Novel hybrid artificial intelligence approaches. Applied Sciences (Basel, Switzerland), 2019, 9(6): 1113
104. Jung Y, Hu J. AK-fold averaging cross-validation procedure. Journal of Nonparametric Statistics, 2015, 27(2): 167−179
105. Marcot B G, Hanea A M. What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis? Computational Statistics, 2021, 36(3): 2009–2031
106. Nguyen T A, Ly H B, Mai H V T, Tran V Q. On the training algorithms for artificial neural network in predicting the shear strength of deep beams. Complexity, 2021, 2021: e5548988
107. Pham B T, Nguyen M D, Dao D V, Prakash I, Ly H B, Le T T, Ho L S, Nguyen K T, Ngo T Q, Hoang V, Son L H, Ngo H T T, Tran H T, Do N M, Van Le H, Ho H L, Tien Bui D. Development of artificial intelligence models for the prediction of Compression Coefficient of soil: An application of Monte Carlo sensitivity analysis. Science of the Total Environment, 2019, 679: 172−184
108. Oner A, Akyuz S. An experimental study on optimum usage of GGBS for the compressive strength of concrete. Cement and Concrete Composites, 2007, 29(6): 505−514
109. Shen J, Xu Q. Effect of moisture content and porosity on compressive strength of concrete during drying at 105 °C. Construction & Building Materials, 2019, 195: 19−27
110. Zhou J, Chen X, Wu L, Kan X. Influence of free water content on the compressive mechanical behaviour of cement mortar under high strain rate. Sadhana, 2011, 36(3): 357−369