

## Research Article

# Application of Extreme Gradient Boosting Based on Grey Relation Analysis for Prediction of Compressive Strength of Concrete

Liyun Cui <sup>1</sup>, Peiyuan Chen <sup>1</sup>, Liang Wang,<sup>1</sup> Jin Li,<sup>1</sup> and Hao Ling<sup>2</sup>

<sup>1</sup>School of Civil Engineering and Architecture, Anhui University of Science and Technology, Huainan 232001, China

<sup>2</sup>School of Materials Science and Engineering, Southeast University, Nanjing 211189, China

Correspondence should be addressed to Peiyuan Chen; peiyuan29@126.com

Received 9 August 2020; Revised 3 October 2020; Accepted 5 February 2021; Published 18 March 2021

Academic Editor: Yuantian Sun

Copyright © 2021 Liyun Cui et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The prediction of concrete strength is an interesting point of investigation and could be realized well, especially for the concrete with the complex system, with the development of machine learning and artificial intelligence. Therefore, an excellent algorithm should put emphasis to receiving increased attention from researchers. This study presents a novel predictive system as follows: extreme gradient boosting (XGBoost) based on grey relation analysis (GRA) for predicting the compressive strength of concrete containing slag and metakaolin. One of its highlights is a feature selection methodology, i.e., GRA, which was used to determine the main input variables. Another highlight is that its performance was compared with the frequently used artificial neural network (ANN) and genetic algorithm-artificial neural network (GA-ANN) by using random dataset and the same testing datasets. For three same testing datasets, the average  $R^2$  values of ANN, GA-ANN, and XGBoost are 0.674, 0.829, and 0.880, respectively, indicating that XGBoost has the highest absolute fraction of variance ( $R^2$ ). XGBoost can provide best result by testing the root mean squared error (RMSE) and mean absolute percentage error (MAPE). The average RMSE values of ANN, GA-ANN, and XGBoost are 15.569 MPa, 10.530 MPa, and 9.532 MPa, respectively, and those of MAPE of ANN, GA-ANN, and XGBoost are 11.224%, 9.140%, and 8.718%, respectively. Thus, the XGBoost definitely performed better than the ANN and GA-ANN. Finally, a type of application software based on XGBoost was developed for practical applications. This vivid software interfaces could help users in prediction and easy and efficient analysis.

## 1. Introduction

A series of dilemmas including waste emission and over-consumption of energy and natural resources have been currently pressing worldwide concerns because of global population explosion and rapid urbanization. According to the International Energy Agency report, 4.8 billion tons of cement will be produced worldwide by 2050, resulting in the emission of 3.8 billion tons of CO<sub>2</sub> [1]. Therefore, it is beneficial to find and utilize active admixtures with high-quality and low-energy consumption as alternatives of cement, partly or totally [1, 2]. In addition, those active admixtures can even enhance the properties of concrete such as compressive strength, antipermeability, and corrosion resistance [3]. Therefore, mix proportion of concrete, especially the high-performance concrete, has been in

increasing demand. Mix design of concrete is difficult to achieve for researchers, as it must reach a specific strength level, which is the most important property of concrete.

Currently, combining with machine learning, some researchers utilize various basic prediction models, such as logistic regression (LR), random forest model (BRF), support vector machine (SVM), and artificial neural network (ANN) [4–6]. These models are commonly used for predicting compressive strength of concrete, irrespective of costly and time-consuming nature. The artificial neural network, inspired by biological systems of the human brain, can learn and generalize from experience without prior knowledge. It has powerful pattern classification and pattern recognition capabilities [7]. Bilim et al. [8] conducted various ANNs and indicated their strong potential for predicting the compressive strength of granulated blast furnace

slag concrete. Saridemir et al. [9] investigated ANNs for predicting the compressive strength of concrete including metakaolin and silica fume, exhibiting good performance. However, the determination of weights and biases of ANN is a complicated process, and the standard ANN algorithm has a slow convergence rate and is easy to fall into the local minimum. As a matter of course, there are many kinds of intelligent algorithms used to optimize the superparameter parameters of the basic model such as conjugate simulated annealing [10], gradient algorithm [11], and genetic algorithm (GA) [12]. Among those algorithms, the performance of GA is excellent, attributing to its advantages such as scalability, adaptation, speed, fault tolerance, modularity, autonomy, and parallelism. Firouzi et al. [13] proposed an integrated GA-ANN approach, appropriate for finding optimum reliability-based inspection plans based on minimum life cycle costs.

The ANN model needs a lot of parameters, and the learning process is like a black box. Hence, it is difficult to carry out the actual derivation step by step, affecting the credibility. In addition, in the learning process, it is easy to appear overfitting state; therefore, some researchers used GA to optimize the internal structure parameters of the neural network. Although GA is a solution for optimizing the neural network, its programming process is complex and the search speed is slow. Therefore, another new model, XGBoost, will be introduced later.

The extreme gradient boosting (XGBoost) is a novel machine learning algorithm mainly used for supporting classification, regression, and ranking. In recent years, it has gradually attracted attention owing to its excellent performance of strong learning ability and fast convergence during parallel learning computations shown in many AI competitions. In KDDCup competition including commercial sales forecast, the team of Top 10 used the XGBoost algorithm for web page text classification, customer behavior prediction, ad click rate prediction, and hazard risk prediction and other fields [14]. In consideration of those outstanding achievements, in this study, XGBoost was applied for predicting the compressive strength of concrete containing slag and metakaolin. In addition, the performance of the model was compared with the commonly used ANN and superior GA-ANN in order to reflect the advantages or disadvantages of XGBoost.

Before predicting the compressive strength of concrete based on ANN, GA-ANN, and XGBoost, the determination of main influencing factors of concrete strength (i.e., the input variables of prediction models) is a primary thing because it is beneficial to reduce the dimensions of the model and improve prediction accuracy [15]. In this study, there is a highlight that a feature selection methodology, grey relation analysis (GRA), was used to determine the main input variables. Figure 1 shows the overall process of predicting compressive strength of concrete containing slag and metakaolin. To train and test those prediction models, 600 groups of data selected from 18 research papers (Table 1) were utilized, after a necessary pretreatment process, i.e., unifying the compressive strength of concrete under different dimensions to avoid the influence of dimension effect. Those data were divided into training dataset and testing

dataset. The quantitative analysis of prediction performance of ANN, GA-ANN, and XGBoost was obtained by testing the value of absolute fraction of variance ( $R^2$ ), mean absolute percentage error (MAPE), and root mean squared error (RMSE).

## 2. Preprocessing and Preanalysis of Selected Data

### 2.1. Preprocessing of Data

**2.1.1. Unifying the Compressive Strength of Concrete.** Because of the dimension effect, the values of compressive strength will be valid just when they are compared under a uniform dimension. A cube with a length of 100 mm was frequently used in 600 groups of collected data. Therefore, it acts as the uniform dimension in this study. For other values of compressive strength of concrete under different dimensions, they would be first transformed by the formula of Neville [32], as shown in the following equation:

$$\frac{P}{P_0} = 0.56 + \frac{0.697}{(V/6h d) + (h/d)}, \quad (1)$$

where  $P$  is the compressive strength of concrete under other dimensions;  $P_0$  is the compressive strength of cube (length = 150 mm, approximately 5.9 inches);  $V$  is the specimen volume;  $h$  is the height; and  $d$  is the maximum transverse dimension of specimen. Notably, all the numerical values must be calculated with a unit of inch. By the formula of Neville and another simple conversion (cube with 150 mm length to cube with 100 mm length), the scaling factor (the value of  $P/P_0$ ) based on the uniform dimension of cube with a length of 100 mm was calculated, as listed in Table 2.

**2.1.2. Normalization of Data.** In the system of data-driven modelling, in order to eliminate the influences of outliers, missed values, and bad data, all raw data should be normalized to fit the range of [0, 1]. It ensures that the raw data collected by major journals are more suitable for modelling, and it also helps to achieve better results and significantly speeds up the calculation [32]. The linear mapping function is as follows:

$$X_m = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \quad (2)$$

where  $X$  represents the original value;  $X_{\min}$  and  $X_{\max}$  are the minimum value and maximum value of dataset, respectively; and  $X_m$  is the value after mapping, noting that the outputs will be remapped to their corresponding real values by the inverse mapping function before calculating any performance criterion [31].

## 3. Preanalysis of Data: Feature Selection Methodology

Generally speaking, the compressive strength of traditional concrete was determined by various factors, including cement type, water-binder ratio (W/B), sand ratio, dosage of

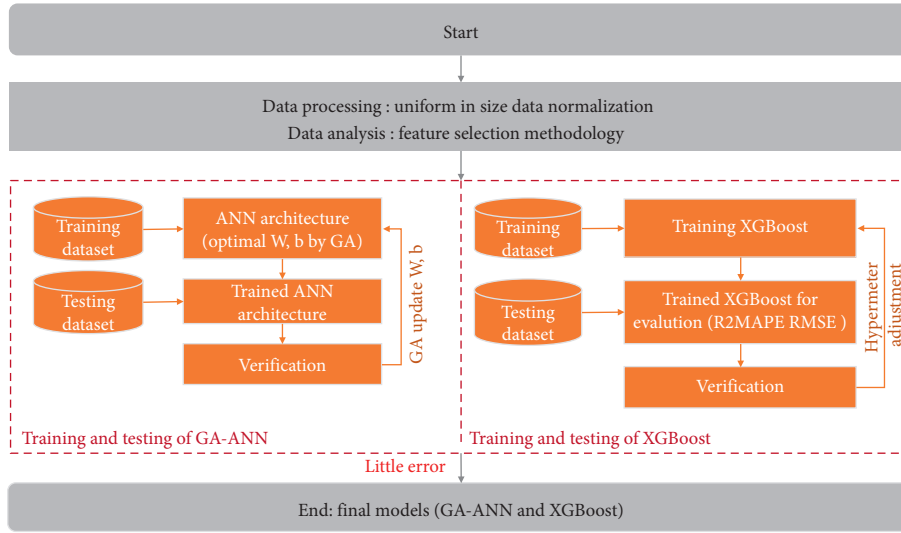


FIGURE 1: The overall process of GA-ANN and XGBoost for predicting the compressive strength of concrete.

TABLE 1: Data source and number of data.

Author of data source	Year	Number of data
Liu et al. [16]	2018	12
Duan et al. [17]	2013	16
Sujjavanich et al. [18]	2017	55
Mermerdaş et al. [19]	2012	84
Güneyisi et al. [20]	2008	36
Yunfeng et al. [21]	2017	90
Ramezani pour et al. [22]	2012	48
Shekarchi et al. [23]	2010	20
Shi et al. [24]	2015	18
Shannag et al. [25]	1995	32
Shafiq et al. [26]	2015	21
Khatib et al. [27]	2008	30
El-diadamony et al. [28]	2016	20
Güneyisi et al. [29]	2012	30
Perez-cortes et al. [30]	2019	14
Dinakar et al. [31]	2013	16
Poon et al. [1]	2001	28
Bilim et al. [8]	2009	30
Total		600

water-reducing agent, and curing age. For concrete containing slag and metakaolin, the content of slag, MK, and the Si/Al of metakaolin (Si/Al of MK) should also be included in consideration of some samples; therefore, the Fly ash content should be considered as an important factor. The selection of main influencing factors among various factors is significant in order to improve the efficiency and accuracy of the prediction models. In this study, GRA, a feature selection methodology, was used to address the abovementioned problem for obtaining the most influential and significant factors.

**3.1. Determination of the Analysis Sequence.** First, we construct the reference matrix (compressive strength of concrete)  $X_0(k)$  (where  $k=1, 2, \dots, 600$ , i.e., the number of sample datasets) and comparative matrix (cement type,

water-binder ratio (W/B), sand ratio, dosage of water-reducing agent, curing age, slag content, fly ash content, MK content, and Si/Al of MK)  $X_i(k)$  (where  $i=1, 2, \dots, 9$ , i.e., the number of factors and  $k=1, 2, \dots, 600$ ). The equations in Table 3 are the mathematical forms for the construction of the reference matrix and the comparative matrix [33].

**3.2. Dimensionless Treatment.** For the purpose of reducing the numerical fluctuation, the normalization process can be determined as follows:

$$x_i(k) = \frac{X_i(k)}{(1/n) \sum_{k=1}^n X_i(k)}, n = 600, \quad i = 1, 2, 3, \dots, 9. \quad (3)$$

**3.3. Calculation of Grey Relational Coefficient.** Subsequently, the grey relational coefficient  $\zeta_i$  can be calculated, according to the following equation [33]:

$$\zeta_i(k) = \frac{\min_i \min_k |x_0(k) - x_i(k)| + \rho \cdot \max_i \max_k |x_0(k) - x_i(k)|}{|x_0(k) - x_i(k)| + \rho \cdot \max_i \max_k |x_0(k) - x_i(k)|}, \quad (4)$$

where  $|x_0(k) - x_i(k)|$  represents the absolute difference between the two sequences;  $\rho$ , the distinguishing coefficient, is usually 0.5, and it represents the significance of  $\max_i \max_k |x_0(k) - x_i(k)|$ . The smaller the  $\rho$  is, the higher its distinguishability is. Therefore,  $\rho=0.5$  was considered, as it can offer more moderate distinguishing effects and better stability [34].

**3.4. Calculation of Grey Relational Grade ( $\gamma_i$ ) and Rank of Results.** As shown in equation (5), it is worth noting that the grey relational grade ( $\gamma_i$ ) can be used for measuring the degree of correlation between the comparative matrix and the reference matrix. The  $\gamma_i$  being close to 1 indicates the

TABLE 2: Scaling factor of compressive strength of concrete under different dimensions.

Shape	Size (mm)	Scaling factor ( $P/P_0$ )
Cylinder	$d = 100, h = 200$	0.72
Cube	1 = 40	1.16
Cube	1 = 50	1.13
Cube	1 = 70	1.07
Cube	1 = 75	1.06
Cube	1 = 100	1.00
Cube	1 = 150	0.91

strong correlation, whereas  $\gamma_i$  being close to 0 depicts the weak correlation.

$$\gamma_i = \frac{1}{n} \sum_{k=1}^n \zeta_i(k), \quad k = 1, 2, \dots, 600. \quad (5)$$

The size of the grey correlation grade  $\gamma_i$  objectively reflects the influencing degree of each factor on the compressive strength of concrete; therefore, choosing the influential and significant factors is important. In this study, MATLAB 7.0 was used to calculate  $\gamma_i$  of nine factors. According to the results of grey relational grade, the rank (cement type > sand ratio >  $W/B$  > MK content > slag content > curing age > dosage of water-reducing agent > Si/Al of MK > fly ash content) was obtained, as shown in Figure 2. However, too many input factors will decrease the calculation efficiency of the models and complicate the model architecture of the input layer and output layer pattern and thus will be unfavorable for the building models. Thus, the factors such as Si/Al of MK and fly ash content, considered as the weak factors, were not taken into account to optimize the model structure because of its low grey correlation grade ( $\gamma_i < 0.9$ ).

After the abovementioned three steps of preprocessing and preanalysis of data source, i.e., unifying the compressive strength of concrete, normalization of data, and selection of main influencing factors, the datasets are ready for training and testing the prediction models. The characteristics of the prepared dataset are listed in Table 4.

## 4. Main Theories and Parameters of ANN, GA-ANN, and XGBoost

### 4.1. Artificial Neural Network (ANN)

**4.1.1. Artificial Neuron.** ANN is inspired by understanding of the biological nervous system [35] and is a mathematical model that imitates the behavioral feature of the human neural network for distributed and parallel processing of information. These are massively parallel complex systems made up of many processing neurons connected by connection weights ( $w_i$ ), as shown in Figure 3.

A single neuron's structure is simple. However, a complete network system containing a mass of basic neurons can be skilled in processing nonlinear problems. Figure 3 shows the notational convention where the orange graphics represents a computational unit and  $X_i$  is an input variable and multiplied by the respective weight  $W_i$ . After that, the

results combined with bias term  $b$  will be transferred to every neuron in the next layer of the network. The summed values (one per neuron in the layer) are then forwarded to the next layer of the network usually *via* an activation function such as sigmoid function, as shown in equation (6) [36]. The output from the activation function  $Y$  can be determined by equation (7), and it will act as the final result or the input variable of the next layer.

$$f(x) = \frac{1}{1 + e^{-x}}, \quad (6)$$

$$Y = f\left(\sum_{i=1}^n W_i X_i + b\right). \quad (7)$$

**4.1.2. Feed-Forward Neural Network and Backpropagation Algorithm.** The feed-forward neural network is one of the simplest neural networks. Each neuron is only connected to the one in the previous layer. Take the output of the previous layer and output it to the next layer. There is no feedback between the layers. It is one of the most widely used and rapidly developed ANNs [37]. Commonly, the feed-forward neural network consists of one input layer, several hidden layers, and one output layer of neurons [38]. The original data are considered as the input information accepted by the first layer (input layer) and then is transmitted to the corresponding neurons of the second layer (hidden layer); at last, it will pass forward to the output layer. Besides, the number of input nodes can be regarded as the input parameters (main influencing factors mentioned previously), and the output node is the target prediction result. Figure 4 shows the framework of the ANN with 7 input nodes and 1 output node used in this study. Notably, in terms of empirical and experimental data types, the number of neurons of the hidden layer needs to be debugged multiple times for creating an optimal predictive neural network model.

There is no feedback process between the layers in the abovementioned ANN model. In order to get a better prediction model, some algorithms such as the frequently used backpropagation (BP) algorithm were added to the ANN for adjusting parameters. The essence of the BP algorithm is decreasing the difference between the actual result and the desirable result in each output layer by the fine adjustment of the weights in the back-propagated process. When the BP algorithm is injected into the forward-feeding neural network, the feed-forward neural network shows excellent performance such as strong generalization ability, fast convergence, and simple structure [39].

During the process of back-propagation, all of the weights should be updated by the BP algorithm in the neural network according to the following formula [40]:

$$W_{ji}(n+1) = W_{ji}(n) + \eta \cdot \delta_j \cdot X_i + \alpha \Delta W_{ji}(n). \quad (8)$$

In equation (8),  $n$  is the number of times,  $\eta$  is the learning rate,  $\delta_j$  means the error signal, and  $\alpha$  is the expression of

TABLE 3: Mathematical forms for the construction of the reference matrix and the comparative matrix.

Compressive strength: $X_0(k) = X_0(1), X_0(2), X_0(3), \dots, X_0(600)$	Reference matrix
Cement type: $X_1(k) = X_1(1), X_1(2), X_1(3), \dots, X_1(600)$	Comparative matrix
W/B: $X_2(k) = X_2(1), X_2(2), X_2(3), \dots, X_2(600)$	
Sand ratio: $X_3(k) = X_3(1), X_3(2), X_3(3), \dots, X_3(600)$	
Dosage of water-reducing agent: $X_4(k) = X_4(1), X_4(2), X_4(3), \dots, X_4(600)$	
Curing age: $X_5(k) = X_5(1), X_5(2), X_5(3), \dots, X_5(600)$	
Slag content: $X_6(k) = X_6(1), X_6(2), X_6(3), \dots, X_6(600)$	
Fly ash content: $X_7(k) = X_7(1), X_7(2), X_7(3), \dots, X_7(600)$	
MK content: $X_8(k) = X_8(1), X_8(2), X_8(3), \dots, X_8(600)$	
Si/Al of MK: $X_9(k) = X_9(1), X_9(2), X_9(3), \dots, X_9(600)$	

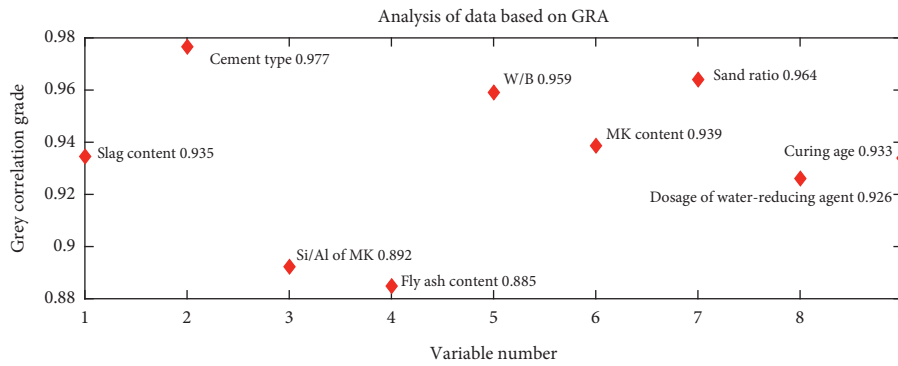


FIGURE 2: The grey correlation grade values of nine factors.

TABLE 4: The characteristics of dataset (600 groups of data selected from 18 research articles).

Factors	Minimum values	Maximum values	Average	Standard deviation
W/B	0.18	0.60	0.35	0.10
Sand ratio (%)	0.33	1	0.45	0.09
Cement type (four types)	1	4	—	—
Slag content (%)	0	20	0.83	0.83
MK content (%)	0	20	9.01	6.90
Dosage of water-reducing agent (%)	0	3.5	0.83	0.84
Curing age (days)	1	180	37.37	43.48
Compressive strength (MPa)	8.93	130.85	68.55	22.68

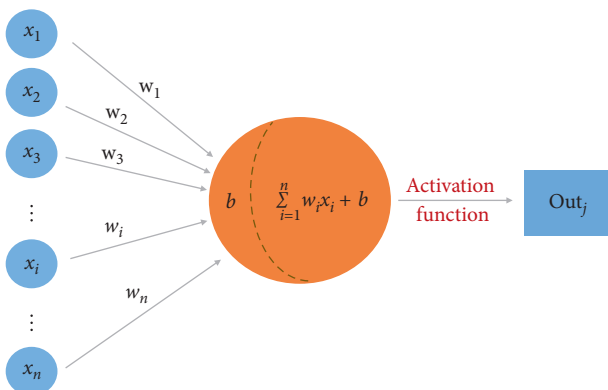


FIGURE 3: A simple artificial neuron model.

momentum factor. The difference between the target value and the true value can be evaluated by the error signal  $\delta_j$  for an output unit  $j$  [41]:

$$\delta_j = (t_j - X_j) \cdot X_j \cdot (1 - X_j). \quad (9)$$

The error signal  $\delta_j$  is a function of the error signals of those units in the next higher layer, connected to unit  $j$  and the weights of those connections [9]:

$$\delta_j = X_j \cdot (1 - X_j) \cdot \sum_k \delta_k W_{kj}. \quad (10)$$

As shown in equation (10), it is worth noting that the  $k_{th}$  layer means the layer in front of the  $j_{th}$  layer. The training procedure should be working until the iterative process converges. The updated connection weights are captured from the final trained network [42].

Although the BP algorithm has been widely used, it still has some shortcomings, such as long training time and easy to fall into local minimum. Therefore, some optimized algorithms were developed by researchers for combining with the ANN model to get a better prediction performance.



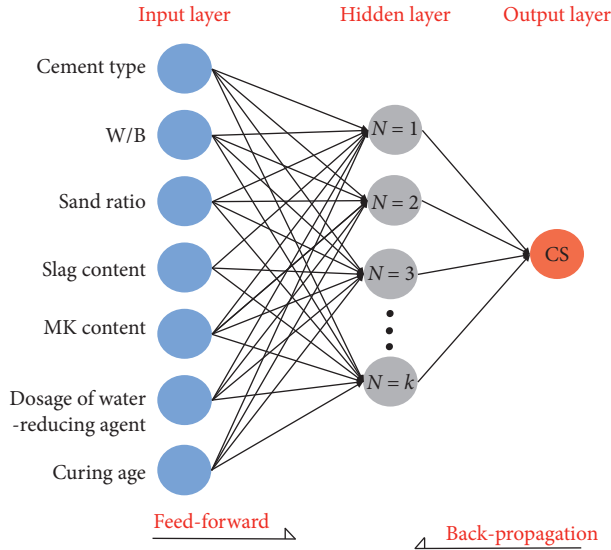


FIGURE 4: Feed-forward and backpropagation of ANN.

## 5. Parameters Optimized by Genetic Algorithm

**5.1. Genetic Algorithm.** The training process of ANN is a procedure of optimization of model parameters, and it starts from a random initial solution and iteratively finds out the optimal parameter values. During each iteration, the error function is first calculated at the current gradient, and then the search direction is determined based on the gradient. However, on this occasion, the optimization of parameters is likely to sink into local minimum (e.g., using the BP algorithm). At this point, the genetic algorithm (GA) was chosen for training the networks to better approximate the global minimum [43].

As one of the most commonly used intelligent optimization algorithms, GA is a mathematical computation model simulating the natural selection of Darwin's biological evolution theory and the process of biological evolution of genetic variation. Moreover, it is a characteristic methodology inspired by the physical phenomenon of biological evolution (selection, crossover, and mutation) to seek the global optimal solutions in space. In the field of artificial intelligence, it produces significant influence on optimization main parameters of the artificial neural network (ANN) [44]. Thus, for creating a powerful model with a higher prediction accuracy, it is necessary to utilize the genetic algorithm to optimize initial weights and thresholds of the ANN model.

**5.2. The Operating Procedure of GA.** Figure 5 shows the main operation procedure of GA used for the optimization of ANN. GA's key elements such as population size, numbers of generation, and fitness function and key operation procedure including selection, evolution, and mutation directly affect the final optimization result, making them definitely important in the operating procedure.

In this study, GA was combined with ANN to get its desired weights and thresholds. During the optimization

process, the initial weights and thresholds of the network could be construed as individuals with prediction errors of the initial ANN as its genetic information. Then, the genetic operation of selection, crossover, and mutation is repeated, to search the optimal individual, namely, the optimal weights and thresholds of ANN (GA-ANN).

## 6. The Extreme Gradient Boosting Methodology (XGBoost)

XGBoost is a machine learning algorithm based on decision tree with a process of gradient promotion [45]. It is one of the most successful machine learning algorithms at present and has been extensively used by researchers in some competitions of machine learning attributing to its excellent learning performance and fast calculation speed, even more used than the popular deep neural network method. In addition, the XGBoost algorithm is generally considered to be superior to ANNs in processing small- and medium-sized structured data although ANNs show excellent performance in analyzing unstructured model data such as pictures, text, audio, and video.

In a word, the main calculation process of XGBoost is an accumulation of iterative results after  $T$  times, as shown in equation (11) [46], where  $i$  is the number of samples;  $T$  is the number of decision trees, and  $\hat{y}_i^{(T)}$  is the final predicted value of the  $i_{th}$  sample in the Decision tree with number  $T$ . Function  $f_t(x_i)$  represents the calculation formula of the  $i_{th}$  sample in the  $T$  decision tree equation (12)),  $\omega$  is the weight vector corresponding to the leaf node, and  $q(x_i)$  is a function of the feature vector  $x_i$  mapped to the leaf node of the decision tree.

$$\hat{y}_i^{(T)} = \hat{y}_i^{(0)} + \sum_{t=1}^T f_t(x_i), \text{ where } f_0(x_i) = \hat{y}_i^{(0)} = 0, \quad (11)$$

$$f_t(x_i) = \omega_q(x_i). \quad (12)$$

To better understand the running process of the XGBoost model, the training process of a traditional decision tree was first introduced, as shown in Figure 6, indicating W/B, curing age, and sand ratio (not all the main factors for simple example), as the decision items for processing the data of four samples. The training process steps are as follows: (i) sample 1 is screened out when the limit is  $W/B > 0.4$ ; (ii) sample 3 is separated from other samples for curing age  $\leq 3$ ; (iii) samples 2 and 4 are divided by the sand ratio of 0.3; and (iv) a traditional decision tree completes an accurate prediction model with some potential problems.

Figures 7–9 show the training process of the XGBoost model. Figure 7 represents the Tree 1 with W/B as the discriminant condition, similar to the first branch of Figure 6. The difference is that the output results of the leaf nodes are averaged to represent the training output values of the four samples (i.e.,  $\omega_1(x_i)$ , where  $i = 1, 2, 3, 4$ ), and then the training error of each sample is obtained by calculating the deviation of the predicted value from the true value. The obtained errors in Tree 1 will be used as the input values of Tree 2 (Figure 8) and the curing age ( $\leq 3$  or  $> 3$ ) will act as the

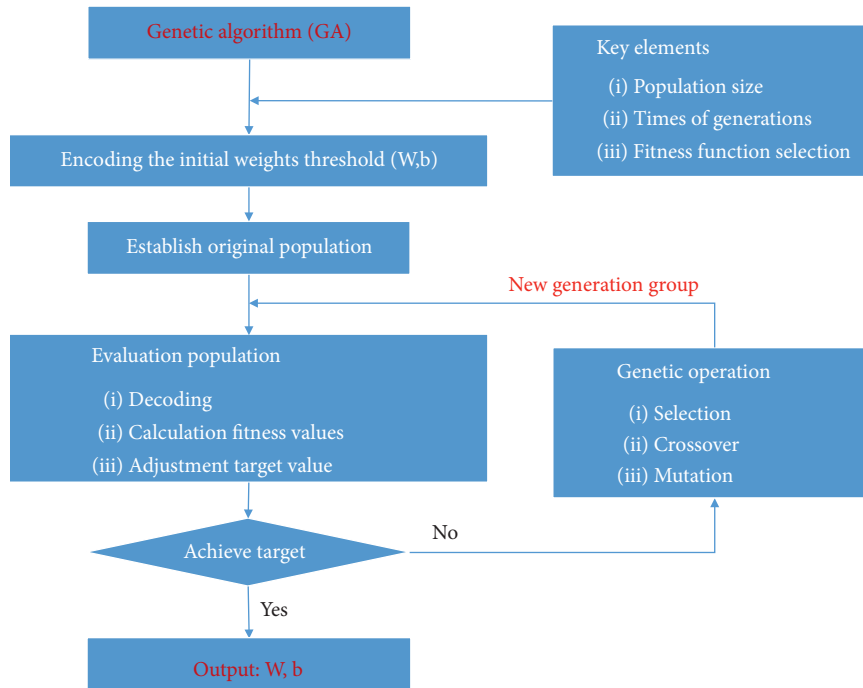
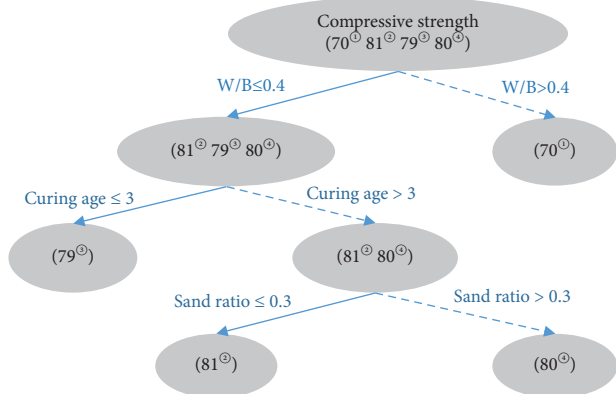


FIGURE 5: The working principle of GA in the optimal procedure.



Factors	Training set			Testing set	
	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
W/B	0.45	0.4	0.4	0.4	0.4
Sand ratio	0.4	0.21	0.21	0.35	0.21
Cement type (four types)	3	3	3	3	3
Slag content (%)	10	20	10	0	10
MK content (%)	0	0	0	0	0
Dosage of water-reducing agent (%)	2	2	2	1	2
Curing age (days)	3	7	3	28	7
Compressive strength (MPa)	70 <sup>⊙</sup>	81 <sup>⊙</sup>	79 <sup>⊙</sup>	80 <sup>⊙</sup>	Unknown

Note: 70<sup>1</sup> 81<sup>2</sup> 79<sup>3</sup> 80<sup>4</sup> in the decision tree on the left represents the compressive strength of the first, second, third and fourth samples respectively

FIGURE 6: The training process of a traditional decision tree.

discrimination standard. Then, the value of  $\omega_{2(x_i)}$  will be obtained based on the average value of decided errors. The predictive values of samples were obtained by the summation of  $\omega_1$  and  $\omega_2$ . The calculated training errors in Tree 2 will be used as input into Tree 3. Similarly, a decision tree based on sand ratio was established, as shown in Figure 9, and the average values are all 0, i.e.,  $\omega_{3(x_i)} = 0$ , where  $i = 1, 2, 3, 4$ . Then, the final predicted values of samples 1, 2, 3, and 4 were calculated by equation  $\omega_{1(x_i)} + \omega_{2(x_i)} + \omega_{3(x_i)}$  and found as 69.5, 80.5, 79.5, and 80.5 MPa, respectively. At this point, the model training is finished. You can find that the output values in XGBoost are different than those in the traditional decision tree. To be more specific, the predictive values of four samples are not equal to their true values in XGBoost

and thus seem to be worse than the traditional decision tree; however, it can solve a big problem of traditional decision tree, i.e., overfitting.

For instance, the traditional decision tree and the XGBoost were used, respectively, to predict the strength value of sample 5, indicating that the traditional decision tree will output the strength value of 81 MPa and that by the XGBoost model is 80.5 MPa. The following analysis can prove the competitive prediction ability of XGBoost. Compared with sample 2, except that the slag content of sample 5 is less than sample 2, other variables have exactly the same values. In combination with the general conclusion of the influence of slag content on the concrete strength, the compressive strength of sample 5 was

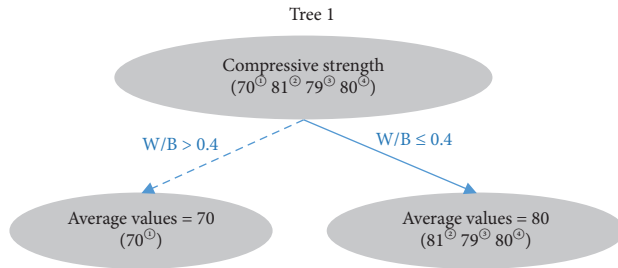


FIGURE 7: The training process of an XGBoost model (Tree 1).

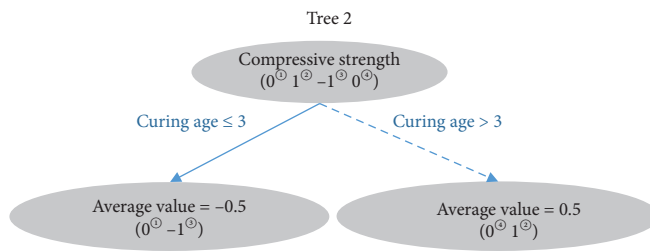


FIGURE 8: The training process of an XGBoost model (Tree 2).

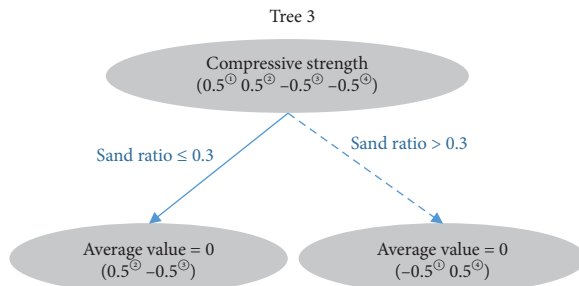


FIGURE 9: The training process of an XGBoost model (Tree 3).

obtained as predicted to be less than 81 MPa. In the same way, it was easy to know that the strength value of sample 5 will be more than that of sample 3 (i.e., 79 MPa) on account of the increasing curing age of samples under the same mix proportion. Hence, the approximate range of the strength of sample 5 will be in the range 79–81 MPa, indicating that the predictive result of XGBoost (80.5 MPa) is more reasonable and the predictive performance of XGBoost is superior than that of the traditional decision tree. This phenomenon is caused by the overfitting training process of the traditional decision tree. It is sometimes inclined to focus on the characteristics of the training data and summarizes the internal rules of the training set (four samples). However, for the new test dataset (sample 5), it cannot reveal its internal change rule. Therefore, a single decision tree model is not generalizable.

Serialnumber	1	2	3	4
W/B	0.45	0.4	0.4	0.4
$f_1(x_i) = \omega_1$	70 <sup>⊖</sup>	80 <sup>⊖</sup>	80 <sup>⊖</sup>	80 <sup>⊖</sup>
$\hat{y}_{\text{predict}}^{(1)}(\omega_1)$	70 <sup>⊖</sup>	80 <sup>⊖</sup>	80 <sup>⊖</sup>	80 <sup>⊖</sup>
$y_{\text{true}}$	70 <sup>⊖</sup>	81 <sup>⊖</sup>	79 <sup>⊖</sup>	80 <sup>⊖</sup>
Training error (Tree 1)	0 <sup>⊖</sup>	1 <sup>⊖</sup>	-1 <sup>⊖</sup>	0 <sup>⊖</sup>

$$f_t(x) = \omega_{q(x)}, t = 3 \quad x\text{-sample}$$

$$\text{Training error} = y_{\text{true}} - \hat{y}_{\text{predict}}^{(t)}$$

Serial number	1	2	3	4
Curing age	3	7	3	28
$f_1(x_i) = \omega_1$	70 <sup>⊖</sup>	80 <sup>⊖</sup>	80 <sup>⊖</sup>	80 <sup>⊖</sup>
$f_2(x_i) = \omega_2$	-0.5 <sup>⊖</sup>	0.5 <sup>⊖</sup>	-0.5 <sup>⊖</sup>	0.5 <sup>⊖</sup>
$\hat{y}_{\text{predict}}^{(2)}(\omega_1 + \omega_2)$	69.5 <sup>⊖</sup>	80.5 <sup>⊖</sup>	79.5 <sup>⊖</sup>	80.5 <sup>⊖</sup>
$y_{\text{true}}$	70 <sup>⊖</sup>	81 <sup>⊖</sup>	79 <sup>⊖</sup>	80 <sup>⊖</sup>
Training error (Tree 2)	0.5	0.5	-0.5	-0.5

$$f_t(x) = \omega_{q(x)}, t = 3 \quad x\text{-sample}$$

Serial number	1	2	3	4
Sand ratio	0.4	0.21	0.21	0.35
$f_1(x_i) = \omega_1$	70 <sup>⊖</sup>	80 <sup>⊖</sup>	80 <sup>⊖</sup>	80 <sup>⊖</sup>
$f_2(x_i) = \omega_2$	-0.5 <sup>⊖</sup>	0.5 <sup>⊖</sup>	-0.5 <sup>⊖</sup>	0.5 <sup>⊖</sup>
$f_3(x_i) = \omega_3$	0 <sup>⊖</sup>	0 <sup>⊖</sup>	0 <sup>⊖</sup>	0 <sup>⊖</sup>
$\hat{y}_{\text{predict}}^{(3)}(\omega_1 + \omega_2 + \omega_3)$	69.5 <sup>⊖</sup>	80.5 <sup>⊖</sup>	79.5 <sup>⊖</sup>	80.5 <sup>⊖</sup>
$y_{\text{true}}$	70 <sup>⊖</sup>	81 <sup>⊖</sup>	79 <sup>⊖</sup>	80 <sup>⊖</sup>

$$f_t(x) = \omega_{q(x)}, t = 3 \quad x\text{-sample}$$

## 7. Setting Structural Parameters and Learning Rate of ANN, GA-ANN, and XGBoost

In this study, the prediction performances of ANN, GA-ANN, and XGBoost were tested and compared. In order to avoid the influence of network structures on the ability of prediction models, the structural parameters of ANN and GA-ANNs were fixed, namely, with the same hidden layers and nodes, as listed in Table 5. Because XGBoost is a definitely different model with a tree structure compared with ANN and GA-ANN, it has different parameters, but some parameters similar to those in ANN and GA-ANN were set.

The programming of ANN, GA-ANN, and XGBoost was performed on the MATLAB 7.0 platform, combined with the dataset (600 groups of data selected from 18 research papers) prepared for the training and testing prediction models.



TABLE 5: Setting parameters of ANN, GA-ANN, and XGBoost.

Structural parameters	ANNs	GA-ANNs	Hyperparameters	XGBoost
Input layer nodes	6	6	Maximum depth	6
Hidden layers	2	2	Minimum child weight	1
Hidden layer nodes	80 (first) 60 (second)	80 (first) 60 (second)	Gamma	0
Output layer nodes	1	1	Subsample	1
Training parameters			Colsample by tree	1
Epoch times	10	10	Regularization alpha	0
Initial learning rate	0.01	0.01	Initial learning rate	0.01
Maximum iterations	100	100	Number of iterations	30

The prediction ability of those models was tested by comparing the value of absolute fraction of variance ( $R^2$ ), root mean squared error (RMSE), and mean absolute percentage error (MAPE). The detailed results are as follows.

## 8. Results and Discussion

*8.1. Comparison of  $R^2$ , RMSE, and MAPE of ANN, GA-ANN, and XGBoost with Random Testing Dataset.* To train and test those prediction models, 600 groups of data after pre-processing were randomly divided into a training dataset (containing 480 groups of data, i.e., 80% of the total dataset) and a testing dataset (containing 120 groups of data, i.e., 20% of total dataset). The values of the absolute fraction of variance ( $R^2$ ), root mean squared error (RMSE), and mean absolute percentage error (MAPE) were adopted to comprehensively evaluate the prediction performances of these models. The detailed equations are as follows:

$$R^2 = \frac{(\mathbf{n} \sum_{i=1}^n \mathbf{t}_i \mathbf{O}_i - \sum_{i=1}^n \mathbf{t}_i \sum_{i=1}^n \mathbf{O}_i)^2}{(\mathbf{n} \sum_{i=1}^n \mathbf{t}_i^2 - (\sum_{i=1}^n \mathbf{t}_i)^2)(\mathbf{n} \sum_{i=1}^n \mathbf{O}_i^2 - (\sum_{i=1}^n \mathbf{O}_i)^2)},$$

$$\text{RMSE} = \sqrt{\frac{1}{\mathbf{n}} \sum_{i=1}^n (\mathbf{t}_i - \mathbf{O}_i)^2},$$

$$\text{MAPE} = \frac{1}{\mathbf{n}} \left[ \frac{\sum_{i=1}^n |\mathbf{t}_i - \mathbf{O}_i|}{\sum_{i=1}^n \mathbf{t}_i} \times 100 \right], \quad (13)$$

where  $\mathbf{t}_i$  is the true value,  $\mathbf{O}_i$  is the predictive value, and  $\mathbf{n}$  is the total number [41, 47].

Figure 10 shows the values of  $R^2$ , RMSE, and MAPE of the three prediction models by using the test dataset. It is easy to find that the values of  $R^2$ , RMSE, and MAPE of ANN are 0.708, 13.38 MPa, and 9.72%, respectively. With the parameters optimized by the genetic algorithm (GA), the prediction performance is better attributing to the better values of  $R^2$  (0.837), RMSE (11.74 MPa), and MAPE (8.26%) of GA-ANN. The third image shows the prediction results of XGBoost. As we know, XGBoost is often known to deal with the classifying problems owing to its excellent performance, but surprisingly, it has also achieved good performance for dealing with regression prediction problems. As shown in Figure 10 (the third), the data points are more concentrated on the line  $y=x$  and nearly without the deviation beyond  $\pm 25$  MPa (the red dotted lines) of predictive values

compared with target values. The results of  $R^2$  (0.872), RMSE (8.62 MPa), and MAPE (8.25%) are also good, better than those obtained by GA-ANN.

The above analysis and discussion indicate superior prediction performance of XGBoost than those by ANN and GA-ANN; however, the results were obtained based on the random testing dataset, indicating that the dataset applied to XGBoost is different with those of ANN and GA-ANN. A special case where the dataset applied to XGBoost is easy to predict was tested, exhibiting good result. In order to eliminate the possibility of that special case and get a more persuasive conclusion, the  $R^2$ , RMSE, and MAPE were tested by using the same testing dataset as described in the following section.

*8.2. Comparison of  $R^2$ , RMSE, and MAPE of ANN, GA-ANN, and XGBoost with the Same Testing Dataset.* The  $R^2$ , RMSE, and MAPE values were calculated thrice by using the three datasets, i.e., dataset 1, 2, and 3 with 120 data in each dataset generated by random selection. In order to easily compare the deviation degree of predictive values (red hollow points) and true values (black solid points) of ANN, GA-ANN, and XGBoost, the diagrams of the results are presented side-by-side (as shown in Figure 11). It is obvious that the predictive values of XGBoost have the maximal degree of coincidence with true values shown in Figure 11, and the degree of coincidence of those points in the ANN is the lowest. It is a subjective conclusion; therefore, the objective results of ANN, GA-ANN, and XGBoost were compared. The specific  $R^2$  values obtained by the ANN, GA-ANN, and XGBoost are 0.708, 0.833, and 0.874, respectively. Second, for the same 120 groups of testing data, the RMSE values obtained by the ANN, GA-ANN, and XGBoost are 13.377, 9.413, and 10.982 MPa, respectively. Lastly, the MAPE values obtained by the ANN, GA-ANN, and XGBoost are 9.733%, 9.289%, and 9.544%, respectively.

The values in Figure 11 indicate that only the  $R^2$  value obtained by XGBoost is the highest, whereas the RMSE and MAPE obtained by the GA-ANN are the lowest than those obtained by the ANN and XGBoost, indicating better prediction performance of GA-ANN than that of XGBoost. However, the results obtained from dataset 2 and dataset 3, as shown in Figures 12 and 13, respectively, show different conclusions. For getting the comprehensive results of comparison, the average  $R^2$ , RMSE, and MAPE values were calculated by the ANN, GA-ANN, and XGBoost. Table 6

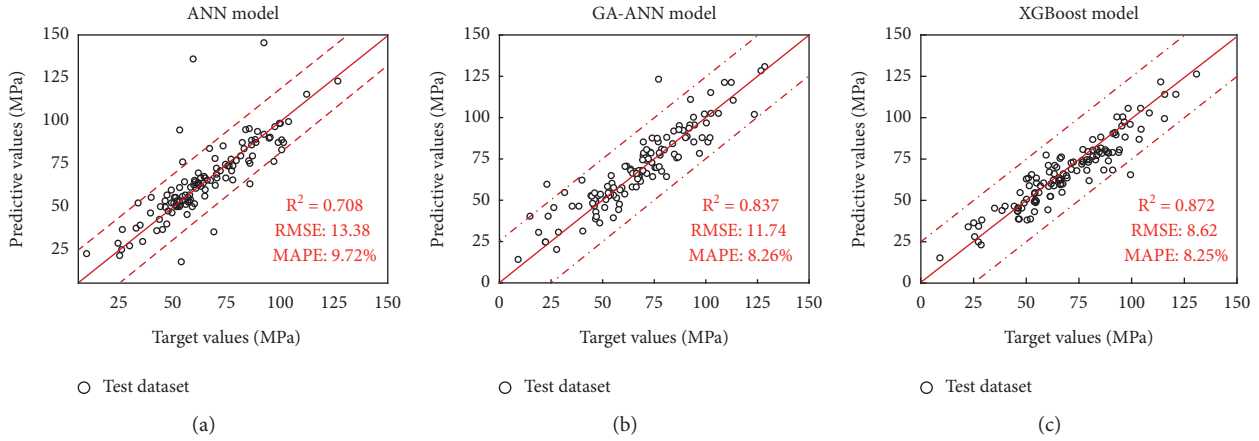


FIGURE 10: Prediction precision of three models for test datasets.

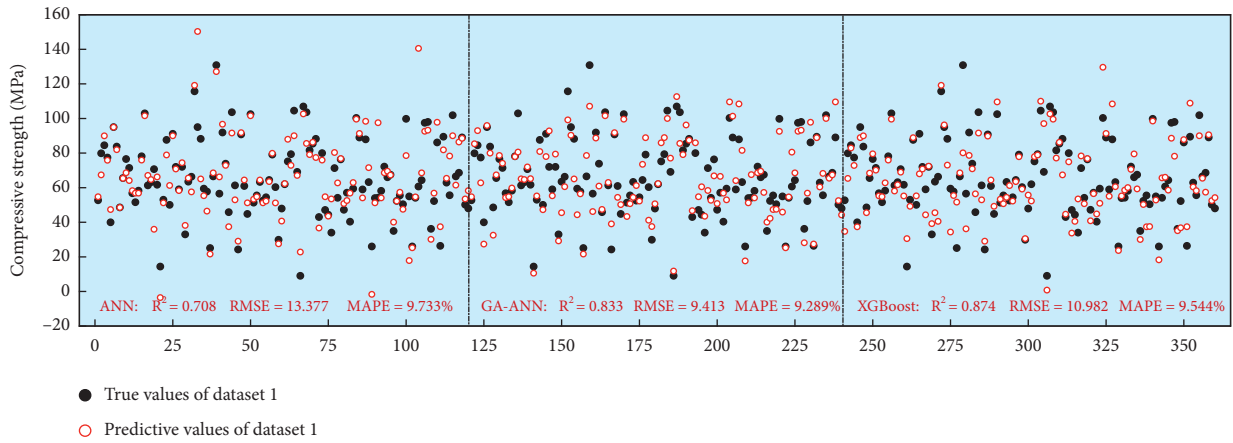


FIGURE 11: Predictive and true values of ANN, GA-ANN, and XGBoost for dataset 1.

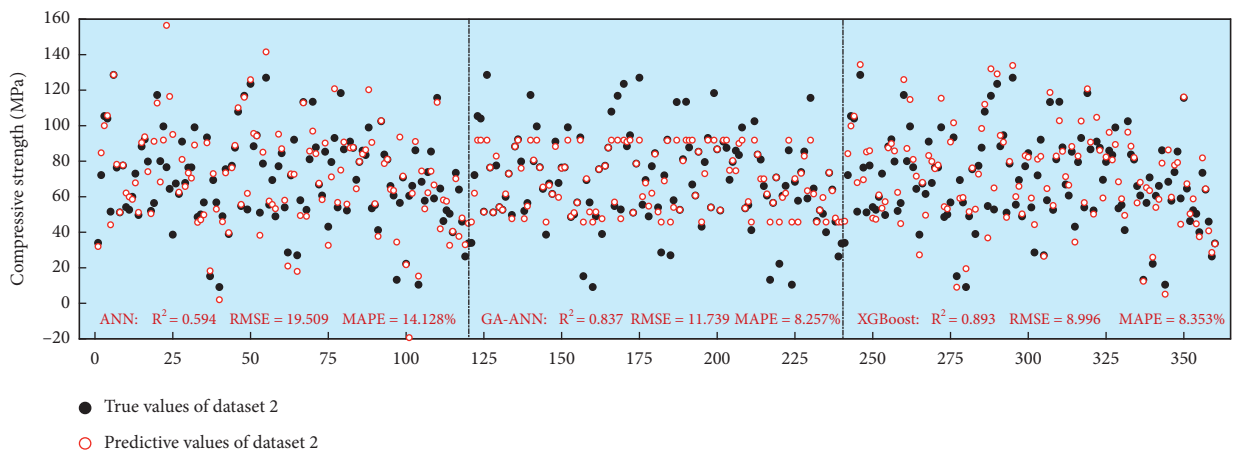


FIGURE 12: Predictive and true values of ANN, GA-ANN, and XGBoost for dataset 2.

shows the average  $R^2$  values (i.e.,  $\bar{R}$ ) of the ANN, GA-ANN, and XGBoost as 0.674, 0.829, and 0.880, respectively, indicating that the highest absolute fraction of variance ( $R^2$ ) by the XGBoost. The RMSE and MAPE values obtained by XGBoost were the best. The average RMSE value of the

ANN, GA-ANN, and XGBoost was 15.569, 10.530, and 9.532 MPa, respectively, and the average MAPE value of the ANN, GA-ANN, and XGBoost was 11.224%, 9.140%, and 8.718%, respectively, certainly verifying the superior prediction performance of XGBoost.

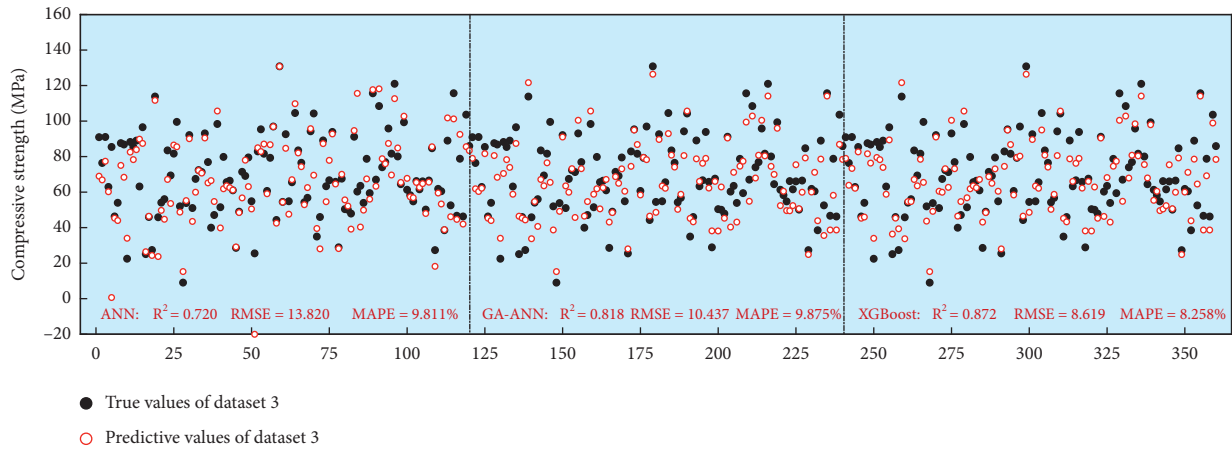


FIGURE 13: Predictive and true values of ANN, GA-ANN, and XGBoost for dataset 3.

TABLE 6: Individual and average values of the  $R^2$ , RMSE, and MAPE of ANN, GA-ANN, and XGBoost.

	$R^2$		RMSE (MPa)		MAPE (%)	
	$R_i (i=1, 2, 3)$	$\bar{R}$	$M_i (i=1, 2, 3)$	$\bar{M}$	$A_i (i=1, 2, 3)$	$\bar{A}$
ANN	0.708	0.674	13.377	15.569	9.733	11.224
	0.594		19.509		14.128	
	0.720		13.820		9.811	
GA-ANN	0.833	0.829	9.413	10.530	9.289	9.140
	0.837		11.739		8.257	
	0.818		10.437		9.875	
XGBoost	0.874	0.880	10.982	9.532	9.544	8.718
	0.893		8.996		8.353	
	0.872		8.619		8.258	

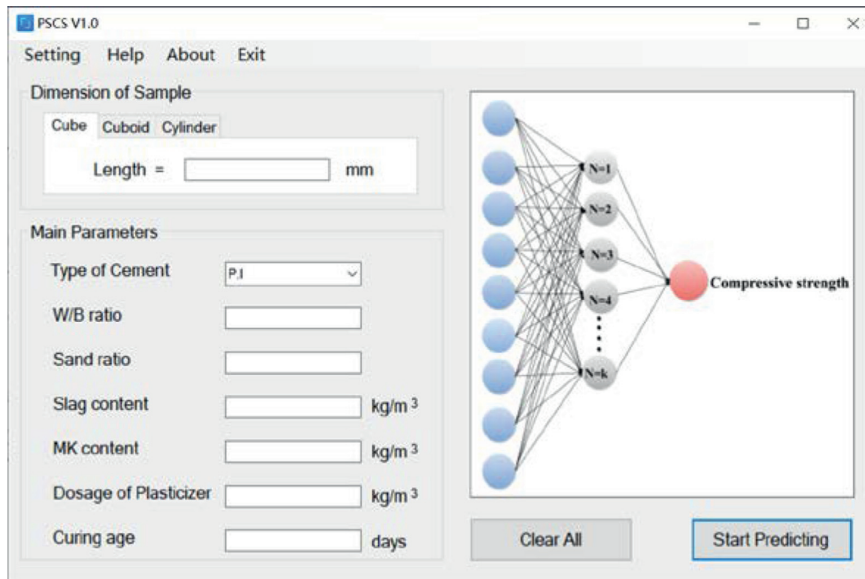


FIGURE 14: Interface for inputting parameters.

The above analyses and discussions of the results obtained by using the random testing dataset and three same testing datasets clearly indicate more stable prediction performance of XGBoost with the highest

prediction accuracy. Therefore, XGBoost is a recommended prediction model and could be used for predicting the compressive strength of concrete containing slag and metakaolin.



FIGURE 15: The procedure of prediction.

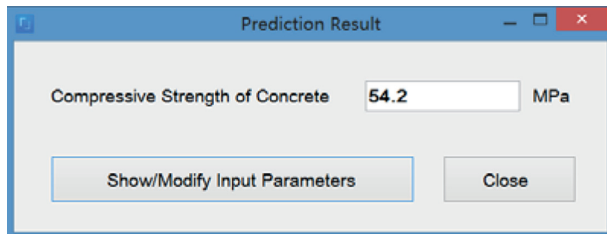


FIGURE 16: The results of compressive strength prediction.

## 9. Application Software for Easy Prediction Based on XGBoost and GRA

At present, the design and development of application software based on various prediction models is receiving increasing attention. This behavior can encourage to develop software from theory to practical application. Therefore, in this study, C# programming was utilized for integrating the involved research results and methods into an application software (named PSCS V1.0) with straightforward interactive interfaces to help users without the knowledge of MATLAB to predict and analyze easily and efficiently. With the help of PSCS, users can just input the dimension of sample (unifying the compressive strength of concrete under different dimensions automatically) and main parameters (the parameters obtained by GRA, i.e., type of cement, W/B ratio, sand ratio, slag content, MK content, dosage of water-reducing agent, and curing age), as shown in Figure 14. After a short wait time for computing of XGBoost (Figure 15), the final prediction results obtained are shown in Figure 16.

PSCS is a short but strong system and could be used for predicting the compressive strength of concrete containing slag and metakaolin. As the dataset used for the training XGBoost model was collected from various journal articles, it can ensure the applicability for different cases. In the following study, relevant literature data were collected continually for building up a complete and more comprehensive database.

## 10. Conclusions

In conclusion, the proposed novel predictive system XGBoost was successfully applied to predict the compressive strength of concrete containing slag and metakaolin, and its performance was compared with the commonly used ANN and optimized GA-ANN models. The datasets used for the training and testing three models were selected from 18

research articles (containing 600 groups of data), with the treatments of unifying the compressive strength of concrete, normalization of data, and selection of main influencing factors. With the help of GRA, the main input parameters such as the type of cement, W/B ratio, sand ratio, slag content, MK content, dosage of water-reducing agent, and curing age were determined scientifically and reasonably. After that, the prediction models were trained and tested by calculating the values of  $R^2$ , RMSE, and MAPE. The main results and conclusions can be drawn as follows:

- (1) For the random testing dataset, the  $R^2$ , RMSE, and MAPE obtained by the ANN are 0.708, 13.38 MPa, and 9.72%, respectively, and those obtained by the GA-ANN are 0.837, 11.74 MPa, and 8.26%, respectively, and the results obtained by XGBoost are 0.872, 8.62 MPa, and 8.25%, respectively, proving a superior prediction performance of XGBoost than those of ANN and GA-ANN.
- (2) XGBoost also exhibited superior prediction performance for prediction of the models based on the same testing dataset. The average  $R^2$  values of ANN, GA-ANN, and XGBoost are 0.674, 0.829, and 0.880, respectively. For the RMSE and MAPE, the average RMSE values of the ANN, GA-ANN, and XGBoost are 15.569, 10.530, and 9.532 MPa, respectively, and the average MAPE values of the ANN, GA-ANN, and XGBoost are 11.224%, 9.140%, and 8.718%, respectively. Therefore, XGBoost was the best prediction model for testing the same datasets.
- (3) Based on the analyses and discussions obtained by using the random testing dataset and the same testing datasets, XGBoost exhibited better performance than the ANN and GA-ANN for prediction capability. Therefore, XGBoost is the best method for predicting the compressive strength of concrete containing slag and metakaolin in this study. Lastly, in order to make XGBoost user friendly and easy to operate, the application software (PSCS V1.0) was encoded by C# in this study, making it a short but strong system that could predict the compressive strength of concrete containing slag and metakaolin efficiently.

## Data Availability

The data used in this paper are from references, and the corresponding literature sources have been marked, so the data have reliability and authenticity.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors would like to express gratitude to all those who helped them during the data analysis and writing of this thesis. This work was supported by the National Natural



Science Foundation of China (no. 51828201), China Postdoctoral Science Foundation (no. 2018M632518), Anhui Postdoctoral Science Foundation (nos. 2017B150 and 2018B248), Natural Science Foundation of Anhui University (no. KJ2018A0074), and Key Research and Development Program Project of Anhui Province (no. 201904a07020081).

## References

- [1] C.-S. Poon, L. Lam, S. C. Kou, Y.-L. Wong, and R. Wong, "Rate of pozzolanic reaction of metakaolin in high-performance cement pastes," *Cement and Concrete Research*, vol. 31, no. 9, pp. 1301–1306, 2001.
- [2] D. M. Roy, P. Arjunan, and M. R. Silsbee, "Effect of silica fume, metakaolin, and low-calcium fly ash on chemical resistance of concrete," *Cement and Concrete Research*, vol. 31, no. 12, pp. 1809–1813, 2001.
- [3] L. Kanamarlapudi, K. B. Jonalagadda, D. C. K. Jagarapu, and A. Eluru, "Different mineral admixtures in concrete: a review," *SN Applied Sciences*, vol. 2, no. 4, 2020.
- [4] F. Özcan, C. D. Atiş, O. Karahan, E. Uncuoğlu, and H. Tanyildizi, "Comparison of artificial neural network and fuzzy logic models for prediction of long-term compressive strength of silica fume concrete," *Advances in Engineering Software*, vol. 40, no. 9, pp. 856–863, 2009.
- [5] S. Safarzadegan Gilan, H. Bahrami Jovein, and A. A. Ramezani pour, "Hybrid support vector regression-particle swarm optimization for prediction of compressive strength and RCPT of concretes containing metakaolin," *Construction and Building Materials*, vol. 34, pp. 321–329, 2012.
- [6] H. Huang, X. Wei, and Y. Zhou, "Twin support vector machines: a survey," *Neurocomputing*, vol. 300, pp. 34–43, 2018.
- [7] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.
- [8] C. Bilim, C. D. Atiş, H. Tanyildizi, and O. Karahan, "Predicting the compressive strength of ground granulated blast furnace slag concrete using artificial neural network," *Advances in Engineering Software*, vol. 40, no. 5, pp. 334–340, 2009.
- [9] M. Saridemir, "Prediction of compressive strength of concretes containing metakaolin and silica fume by artificial neural networks," *Advances in Engineering Software*, vol. 40, no. 5, pp. 350–355, 2009.
- [10] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons, New York, NY, USA, 1988.
- [11] C. Charalambous, "Conjugate gradient algorithm for efficient training of artificial neural networks," *Institute of Electrical and Electronics Engineers Proceedings G Circuits, Devices and Systems*, vol. 139, no. 3, pp. 301–310, 1992.
- [12] J. H. Holland, *Genetic Algorithms and Adaptation*, Springer US, New York, NY, USA, 1984.
- [13] A. Firouzi and A. Rahai, "An integrated ANN-GA for reliability based inspection of concrete bridge decks considering extent of corrosion-induced cracks and life cycle costs," *Scientia Iranica*, vol. 19, no. 4, pp. 974–981, 2012.
- [14] G. Dror, M. Boulle, and I. Guyon, *The 2009 Knowledge Discovery and Data Mining Competition (KDD Cup 2009): Challenges in Machine Learning*, Microtome Publishing, Brookline, MA 02446, USA, 2011.
- [15] R. L. Wilson and R. Sharda, "Bankruptcy prediction using neural networks," *Decision Support Systems*, vol. 11, no. 5, pp. 545–557, 1994.
- [16] F. Liu, Z. You, X. Yang, and H. Wang, "Macro-micro degradation process of fly ash concrete under alternation of freeze-thaw cycles subjected to sulfate and carbonation," *Construction and Building Materials*, vol. 181, pp. 369–380, 2018.
- [17] P. Duan, Z. Shui, W. Chen, and C. Shen, "Effects of metakaolin, silica fume and slag on pore structure, interfacial transition zone and compressive strength of concrete," *Construction and Building Materials*, vol. 44, pp. 1–6, 2013.
- [18] S. Sujjavanich, P. Suwanvitaya, D. Chaysuwan, and G. Heness, "Synergistic effect of metakaolin and fly ash on properties of concrete," *Construction and Building Materials*, vol. 155, pp. 830–837, 2017.
- [19] E. Güneyisi, M. Gesoğlu, S. Karaoğlu, and K. Mermerdaş, "Strength, permeability and shrinkage cracking of silica fume and metakaolin concretes," *Construction and Building Materials*, vol. 34, no. 34, pp. 120–130, 2012.
- [20] E. Güneyisi, M. Gesoğlu, and K. Mermerdaş, "Improving strength, drying shrinkage, and pore structure of concrete using metakaolin," *Materials and Structures*, vol. 41, pp. 937–949, 2008.
- [21] L. I. Yunfeng, M. A. Jiandong, S. Zhang, L. I. Qianglong, and D. Wang, *Effects of Metakaolin Admixture on the Compressive Strength of High-Strength Concrete*, China Sciencepaper, Beijing, China, 2017.
- [22] A. A. Ramezani pour and H. Bahrami Jovein, "Influence of metakaolin as supplementary cementing material on strength and durability of concretes," *Construction and Building Materials*, vol. 30, pp. 470–479, 2012.
- [23] M. Shekarchi, A. Bonakdar, M. Bakhshi, A. Mirdamadi, and B. Mobasher, "Transport properties in metakaolin blended concrete," *Construction and Building Materials*, vol. 24, no. 11, pp. 2217–2223, 2010.
- [24] Z. Shi, Z. Shui, Q. Li, and H. Geng, "Combined effect of metakaolin and sea water on performance and microstructures of concrete," *Construction and Building Materials*, vol. 74, pp. 57–64, 2015.
- [25] M. J. Shannag and A. Yeginobali, "Properties of pastes, mortars and concretes containing natural pozzolan," *Cement and Concrete Research*, vol. 25, no. 3, pp. 647–657, 1995.
- [26] N. Shafiq, M. F. Nuruddin, S. U. Khan, and T. Ayub, "Calcined kaolin as cement replacing material and its use in high strength concrete," *Construction and Building Materials*, vol. 81, pp. 313–323, 2015.
- [27] J. M. Khatib, "Metakaolin concrete at a low water to binder ratio," *Construction and Building Materials*, vol. 22, no. 8, pp. 1691–1700, 2008.
- [28] H. El-Diadamony, A. A. Amer, T. M. Sökkary, and S. El-Hoseny, "Hydration and characteristics of metakaolin pozzolanic cement pastes," *HBRC Journal*, vol. 14, no. 2, pp. 150–158, 2018.
- [29] K. Mermerdaş, M. Gesoğlu, E. Güneyisi, and T. Özturan, "Strength development of concretes incorporated with metakaolin and different types of calcined kaolins," *Construction and Building Materials*, vol. 37, pp. 766–774, 2012.
- [30] P. Perez-Cortes and J. I. Escalante-Garcia, "Metakaolin based geopolymers with high limestone contents—statistical modeling of strength and environmental and cost analyses," *Cement and Concrete Composites*, vol. 106, Article ID 103450, 2019.
- [31] P. Dinakar, P. K. Sahoo, and G. Sriram, "Effect of metakaolin content on the properties of high strength concrete," *International Journal of Concrete Structures and Materials*, vol. 7, no. 3, pp. 215, 2013.



- [32] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *Institute of Electrical and Electronics Engineers Transactions on Nuclear Science*, vol. 44, no. 3, pp. 1464–1468, 1997.
- [33] J. Xu, X. Zhao, Y. Yu, T. Xie, G. Yang, and J. Xue, "Parametric sensitivity analysis and modelling of mechanical properties of normal- and high-strength recycled aggregate concrete using grey theory, multiple nonlinear regression and artificial neural networks," *Construction and Building Materials*, vol. 211, pp. 479–491, 2019.
- [34] T. C. Chang and S. J. Lin, "Grey relation analysis of carbon dioxide emissions from industrial production and energy uses in Taiwan," *Journal of Environmental Management*, vol. 56, no. 4, pp. 247–257, 1999.
- [35] F. Altun, Ö. Kişi, and K. Aydin, "Predicting the compressive strength of steel fiber added lightweight concrete using neural network," *Computational Materials Science*, vol. 42, no. 2, pp. 259–265, 2008.
- [36] J.-R. Zhang, J. Zhang, T.-M. Lok, and M. R. Lyu, "A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1026–1037, 2007.
- [37] B. B. Adhikary and H. Mutsuyoshi, "Prediction of shear strength of steel fiber RC beams using neural networks," *Construction and Building Materials*, vol. 20, no. 9, pp. 801–811, 2006.
- [38] A. Mukherjee and S. Nag Biswas, "Artificial neural networks in prediction of mechanical behavior of concrete at high temperature," *Nuclear Engineering and Design*, vol. 178, no. 1, pp. 1–11, 1997.
- [39] J. Bai, S. Wild, J. A. Ware, and B. B. Sabir, "Using neural networks to predict workability of concrete incorporating metakaolin and fly ash," *Advances in Engineering Software*, vol. 34, no. 11-12, pp. 663–669, 2003.
- [40] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon, "Accelerating the convergence of the back-propagation method," *Biological Cybernetics*, vol. 59, no. 4-5, pp. 257–263, 1988.
- [41] M. Saridemir, "Predicting the compressive strength of mortars containing metakaolin by artificial neural networks and fuzzy logic," *Advances in Engineering Software*, vol. 40, no. 9, pp. 920–927, 2009.
- [42] R. Ince, "Prediction of fracture parameters of concrete by artificial neural networks," *Engineering Fracture Mechanics*, vol. 71, no. 15, pp. 2143–2159, 2004.
- [43] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2/3, pp. 95–99, 1988.
- [44] R. Rebouh, B. Boukhatem, M. Ghrici, and A. Tagnit-Hamou, "A practical hybrid NNGA system for predicting the compressive strength of concrete containing natural pozzolan using an evolutionary structure," *Construction and Building Materials*, vol. 149, pp. 778–789, 2017.
- [45] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, New York, NY, USA, August 2016.
- [46] S. Lim and S. Chi, "Xgboost application on bridge management systems for proactive damage estimation," *Advanced Engineering Informatics*, vol. 41, Article ID 100922, 2019.
- [47] R. Siddique, P. Aggarwal, and Y. Aggarwal, "Influence of water/powder ratio on strength properties of self-compacting concrete containing coal fly ash and bottom ash," *Construction and Building Materials*, vol. 29, pp. 73–81, 2012.