Research Article

Grey Wolf Optimizer-Based ANNs to Predict the Compressive Strength of Self-Compacting Concrete

Amir Andalib ¹, Babak Aminnejad ² and Alireza Lork ³

¹Department of Civil Engineering, Kish International Branch, Islamic Azad University, Kish Island, Iran

²Department of Civil Engineering, Roudehen Branch, Islamic Azad University, Roudehen, Iran

³Department of Civil Engineering, Safadasht Branch, Islamic Azad University, Tehran, Iran

Correspondence should be addressed to Babak Aminnejad; aminnejad@riau.ac.ir

Received 10 September 2021; Revised 3 January 2022; Accepted 18 January 2022; Published 17 February 2022

Academic Editor: Cheng-Jian Lin

Copyright © 2022 Amir Andalib 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.

Ever since their presentation in the late 80s, self-compacting concrete (SCC) has been well received by researchers. SCC can flow under their weight and exhibit high workability. Nonetheless, their nonlinear behavior has made the prediction of their mix properties more demanding. Furthermore, the complex relationship between mixed proportions and rheological and mechanical properties of SCC renders their behavior prediction challenging. Soft computing approaches have been shown to optimize and reduce uncertainties, and therefore in this paper, we aim to address these challenges by employing artificial neural network (ANN) models optimized using the grey wolf optimizer (GWO) algorithm. The optimized model proved to be more accurate than genetic algorithms and multiple linear regression models. The results indicate that the four most influential parameters on the compressive strength of SCC are the cement content, ground granulated blast furnace slag, rice husk ash, and fly ash.

1. Introduction

A proper self-compacting concrete mix design requires balancing two conflicting objectives: deformability and stability, that is, acceptable rheological behavior and appropriate mechanical characteristics. So, the proportions of available materials, minerals, and admixtures must be considered. The optimum balance of coarse and fine aggregates and chemical admixtures ensures the greater cohesiveness of self-compacting concrete. External variations such as changes in the production process of cement and mineral additives and the type of aggregates can trigger significant variations in the properties of fresh self-compacting concrete. To minimize such external variants, the use of industrial derivatives and mineral additives in the manufacturing of lightweight self-compacting concrete has been the focus of many scientists [1].

Due to the complex relationship between mixed proportions and rheological and mechanical properties of SCC, researchers have proposed numerous treatments in the literature. Some researchers have used statistical models

such as linear regression to predict the compressive strength of SCC [2], while others used numerical methods to this end [3]. Among the different approaches suggested, soft computing-based methods have reported promising results [4]. They include methods such as random kitchen sink algorithm [5], swarm optimization algorithm [6], and back-propagation algorithm [7].

Zhang et al. developed a random forest model based on the beetle antenna search algorithm to predict the compressive strength of SCC [8], while Nehdi et al. utilized a neural network approach to this end [2]. ANNs have shown promising results in the engineering domain [9, 10] and are widely employed by researchers to predict concrete's compressive strength [11, 12]. Prasad et al. employed ANNs to predict the compressive strength of high-performance SCC [13], and Siddique et al. used ANNs for SCC containing fly ash [14]. Uysal and Tanyildizi used ANNs to predict the compressive strength of SCC mixtures with mineral additives [15]. A novel approach based on the proposed normalization method for artificial neural networks was employed by Asteris et al. to determine the compressive

strength of self-compacting concrete [1, 16]. Support vector machine and relevance vector machine methodologies were proposed by Aiyer et al. to predict the compressive strength of SCC [17], and Ghanizadeh et al. investigated ANNs and support vector machines in their research [18]. Asteris et al. proposed predictive models for the compressive strength of self-compacting concrete using surrogate models such as multivariate adaptive regression splines and M5P model tree [19].

There are several opportunities in soft computing to develop new models aiming to optimize and reduce uncertainties. Mirjalili introduced the grey wolf optimizer metaheuristic algorithm, which is inspired by grey wolves' social hierarchy and has successfully outperformed other metaheuristic algorithms [20, 21].

This study emphasizes enhancing the prediction model by training an ANN using the GWO algorithm. The proposed ANN-GWO model addresses all significant mix design parameters that influence SCC's compressive strength. The accuracy of the model is confirmed against 205 samples of a database from the literature. Section 2 offers the required literature review on ANNs and GWO. Section 3 addresses the development and training perspectives of the empirical model. Finally, the results and conclusion of the study are presented in Sections 4 and 5. An overview of the research study is illustrated in Figure 1.

2. Background

2.1. Artificial Neural Networks. Artificial Neural Networks (ANNs) are among the most dynamic areas of current research [22, 23]. The features and capabilities of ANNs include learning and adapting to existing knowledge, generalizability, parallel processing, and thus, higher processing speed and a high tolerance for errors [23, 24]. Figure 2 shows an ANN with 11 neurons in the input layer, two hidden layers comprising 7 and 13 neurons, and a single output layer.

The basis of ANNs is as follows [25]:

- (1) Data are processed in units called neurons (or nodes).
- (2) The signals between nodes are transmitted over the connection lines.
- (3) The weight associated with each connection indicates its power.
- (4) An activation function is applied to the weighted input (plus a bias value) of each neuron to determine its output [25].

A feed-forward network is a type of ANN in which the connection between its constituent units does not form a cycle, and information flows from the input nodes through the hidden layers to the output nodes [22].

The weights of an ANN are generally set randomly, thus, resulting in different output values in the network. The network's weights and biases are adjusted and optimized in a training process to reduce the model error to minimize the model error [26, 27]. The methods for solving this optimization problem can generally be divided into metaheuristic and

gradient-based methods. Gradient-based methods are fast but can get stuck in the local minima. Contrarily, metaheuristic methods do not get trapped in the local minima. However, there is a drawback to using metaheuristic methods (<https://www.sciencedirect.com/science/article/abs/pii/S235201242100182X?via%3DihubThanks>). Their result might not provide the global minimum. Nonetheless, considering their speed and capabilities, they are designed to explore and exploit large parts of the solution space to return precise outcomes [26–28].

2.2. Grey Wolf Optimizer. Native to the remote areas of Eurasia and North America, grey wolves are members of the Canidae family and are at the top of the food chain. Proposed by Mirjalili et al. in 2014, the grey wolf optimizer (GWO) algorithm is inspired by the strict sovereignty hierarchy of these animals [20].

The grey wolves typically live in groups of 5 to 12, and their social dominance hierarchy consists of four groups: alphas, betas, deltas, and omegas. Alpha grey wolves are the pack's leaders and dictate things like the decision for hunting, place to sleep, time to wake up, and so on [20, 21, 29]. The alphas are the best managers of the pack and not necessarily the strongest. The betas are the second in command. They assist alpha wolves in decision-making. Deltas (or, in some references, subordinates) are responsible for tasks such as scouting, guarding, hunting, and caregiving. Deltas are the third in command after alphas and betas. The last rank in the dominance hierarchy is the omegas. They follow their ranking wolves and are the last ones allowed to eat [20].

Grey wolves also have strict hunting rituals. The main stages of their hunting are as follows [30]:

- (1) Pursuing and advancing toward the prey.
- (2) Circling and harassing the target to make it stop moving.
- (3) Attacking the prey [20].

If the position of the prey is assumed to be $\mathbf{X}_p$, then the hunting process can be described using the following equations:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \quad (1)$$

$$\vec{C} = 2 \cdot \vec{r}_2, \quad (2)$$

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right|, \quad (3)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}, \quad (4)$$

where $\mathbf{A}$ and $\mathbf{C}$ are coefficient vectors computed using the $\mathbf{r}_1$ and $\mathbf{r}_2$ random vectors that are in the $[0, 1]$ range, $\mathbf{a}$ is a vector with components decreasing from 2 to 0 during the algorithm's iterations, t is the current iteration, $\mathbf{X}(t)$ is the position vector of the hunting wolf, and $\mathbf{X}(t+1)$ is its updated position in the $t+1$ iteration [20].

The grey wolf optimizer starts by randomly generating a wolf population and then ranking them to increase values

FIGURE 1: Overview of the research.

FIGURE 2: The present ANN architecture comprising 11 input neurons, 7 neurons in the first hidden layer, 13 neurons in the second hidden layer, and a single neuron in the output layer.

of the cost function used to evaluate solutions. Then, GWO labels the top, second, and third solutions as alpha, beta, and delta, respectively, to simulate the social hierarchy of the grey wolves. The algorithm designates the remaining population of wolves as omega that follows the alphas, betas, and deltas. Considering that we are unaware of the exact location of the prey in the search space, we assume

that the top three wolves have better knowledge of the location of the potential prey. Accordingly, in GWO, the hunting process is driven by alpha, beta, and delta. So GWO modifies equations (1) to (4) by substituting the location of alpha, beta, and delta in place of the location of the prey and updating the location of the remaining wolves. The update rules are as follows [20]:

$$\begin{aligned}\vec{D}_\alpha &= |\vec{C}_1 \cdot \vec{X}_\alpha|, \\ \vec{D}_\beta &= |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \\ \vec{D}_\delta &= |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|,\end{aligned}\quad (5)$$

$$\begin{aligned}\vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta), \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta),\end{aligned}\quad (6)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}, \quad (7)$$

where the A_1 , A_2 , and A_3 values are calculated using equation (1), and C_1 , C_2 , and C_3 are calculated using equation (2). $\vec{X}(t+1)$ is the location of the omega at the end of the iteration. The GWO updates the locations of all omega wolves using equations (5) to (7) and at the end of each iteration calculates the cost of all solutions and designates the top three solutions as alpha, beta, and delta. In a typical implementation, the algorithm continues until a predefined number of iterations has been completed. Figure 3 displays how a search agent (i.e., omega) updates its position consistent with the location of alpha, beta, and delta in a two-dimensional search space. To explain it more clearly, the alpha, beta, and delta wolves predict the prey's position while the rest of the pack randomly (close in) update their position around the prey [20].

When using the GWO algorithm for ANN, each wolf's position vector represents the weights and biases of the neural network. Therefore, in terms of dimension, it equals the weights and biases of the network. The prediction error of the network is defined as the cost function. After the GWO has completed its set number of iterations, the position vector of the wolf with the least cost (the alpha) is selected as the trained network's final weights and biases.

3. Methods and Materials

3.1. Dataset. The database in this paper uses the dataset obtained by Asteris and Kolovos [1]. The dataset includes 205 experimental self-compacting concrete (SCC) compressive strength results collected from several articles by Asteris and Kolovos [31–38]. As suggested by Asteris and Kolovos [1], the influencing parameters on compressive strength of SCC are the 11 parameters of cement (C), silica fume (SF), rice husk ash (RHA), limestone powder (LP), ground granulated blast furnace slag (GGBFS), coarse aggregate (CA), fine aggregate (FA), fly ash (F), water (W), viscosity modifying admixtures (VMA), and new generation superplasticizers (SP) as chemical admixtures. The SCC specimens' 28-day compressive strength is taken as the target value of [1]. The inputs are in the units of kg/m^3 , and the compressive strength is in MPa. The maximum,

FIGURE 3: Update rule of grey wolf optimizer [20].

minimum, mean, median, standard deviation, range, standard error, and average deviation of the data are provided in Table 1.

The histogram of the 28-day compressive strength of SCC specimens is depicted in Figure 4. As can be seen, most of the samples (159 cases) have compressive strength values that range from 30 to 80 MPa.

During ANN training, different input variable ranges can negatively affect the resulting model, including the divergence of the optimization algorithm and increased training time [26]. The input and output variables were normalized to the $[-1, 1]$ range using the following equation:

$$X_n = \frac{2(X - X_{\min})}{X_{\max} - X_{\min}} - 1, \quad (8)$$

where X_n is the normalized value of the variable, $X_{\max}$ is the maximum value, $X_{\min}$ is the minimum value, X is the original (nontransformed) value of the variable.

The minimum and maximum values used for each of the 11 input parameters and the target value of compressive strength are provided in Table 1. It is noteworthy considering that ANN is trained using normalized data. It is necessary to feed the network with normalized inputs when training the ANNs and using the original values (unnormalizing the data values) for the network output [9].

When evaluating the model results, it is essential to define the measures by which the model's performance and accuracy will be evaluated. The objective used will be the best fitness value (or lowest cost) on test data. The reason for gauging model performance based on test data is to choose the model with the highest generalization capability.

The statistical indices, including Mean Error (ME), Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Average Absolute Error (AAE), Model Efficiency (EF), and Variance Account Factor (VAF), are utilized to assess the performance of different topologies. They are defined as follows [39]:

TABLE 1: Descriptive statistics of the experimental data.

Input parameter	Min	Max	Mean	Standard deviation	Median	Range	Standard error	Average deviation
Cement	110	600	349.22	93.43	337.5	490	6.53	72.48
Limestone powder	0	272	25.67	60.78	0	272	4.25	41.82
Fly ash	0	440	106.36	94.01	110	440	6.57	78.60
GGBS	0	330	17.39	52.01	0	330	3.63	30.19
Silica fume	0	250	14.91	33.45	0	250	2.34	22.16
RHA	0	200	6.55	24.29	0	200	1.70	11.88
Coarse aggregate	500	1600	772.35	175.36	768.88	1100	12.25	124.16
Fine aggregate	336	1135	827.93	144.33	836	799	10.08	105.95
Water	94.5	250	179.27	27.65	180	155.5	1.93	20.49
SP	0	22.5	5.96	4.35	5.97	22.5	0.30	3.33
VMA	0	1.23	0.14	0.31	0	1.23	0.02	0.22

FIGURE 4: Distribution of compressive strength in the experimental data.

$$\begin{aligned}
ME &= \frac{1}{n} \sum_{i=1}^n (P_i - O_i), \\
MAE &= \frac{1}{n} \sum_{i=1}^n |P_i - O_i|, \\
MSE &= \frac{1}{n} \sum_{i=1}^n (P_i - O_i)^2, \\
RMSE &= \left[\frac{1}{n} \sum_{i=1}^n (P_i - O_i)^2 \right]^{1/2}, \\
AAE &= \frac{|\sum_{i=1}^n (O_i - P_i)/O_i|}{n} \times 100, \\
EF &= 1 - \frac{\sum_{i=1}^n (P_i - O_i)^2}{\sum_{i=1}^n (O_i - O_i)^2}, \\
VAF &= \left[1 - \frac{\text{var}(O_i - P_i)}{\text{var}(O_i)} \right] \times 100.
\end{aligned} \tag{9}$$

3.2. Developing the Empirical Model Using ANNs and GWO. According to Section 3.1, a total of 11 parameters affect the SCC's compressive strength. Consequently, as illustrated in Figure 2, the trained ANNs comprise 11 neurons in the input layer and a single neuron in the output layer.

ANNs are inclined toward the overfitting phenomenon, meaning that the trained ANN is a high performer (i.e., with minimum error on training data) but fails to perform well on the test data, which are unavailable during the training process. As recommended in literature [9, 26], data were randomly divided into two sets to alleviate the overfitting effect. Seventy percent (144 test cases) were employed for ANN training, and the remainder (61 test cases) were used as test data.

In ANNs, the number of hidden layers and their neurons varies based on the problem [40]. Consequently, a trial-and-error approach must be taken to identify a suitable architecture that best represents the existing data. Equation (10) offers a frequently used heuristic that calculates the total number of neurons in an ANN [41]:

$$N_H \leq 2N_I + 1, \tag{10}$$

where N_H is the number of hidden layer nodes and N_I is the number of inputs.

Considering that the number of effective parameters equals 11, equation (10) shows that the number of hidden layers could be less than 23. Therefore, different architectures with two hidden layers and 26 neurons (three more than the suggested value) were trained. Each hidden layer possessed 1 to 13 neurons; thus, 13^2 equals 169 different architectures (see Table 2) undertook training. For all the ANNs, we selected the hyperbolic tangent function as the activation function of the hidden layers; also, the identity function was selected as the activation function of the output layer.

The ANN training process (i.e., adjusting the weights and biases) is a minimization problem. The optimal solution to the problem is choosing the weights and biases that minimize the network error (cost function). The grey wolf optimizer (GWO) algorithm (see Section 2.2) was used for this purpose. In GWO, the weights and biases of the ANN represented each wolf's position vector, and the optimal solution, that is, the alpha grey wolf, found during the predetermined number of iterations of the algorithm

TABLE 2: Artificial neural network architectures trained.

Num	Topology	Num	Topology	...	Num	Topology	Num	Topology
1	1-1	14	2-1	...	144	12-1	157	13-1
2	1-2	15	2-2	...	145	12-2	158	13-2
3	1-3	16	2-3	...	146	12-3	159	13-3
4	1-4	17	2-4	...	147	12-4	160	13-4
...	...	...	...	...	...	...	...	...
12	1-12	25	2-12	...	155	12-12	168	13-12
13	1-13	26	2-13	...	156	12-13	169	13-13

represented the ANN's weights and biases, giving the least prediction error for the architecture trained.

MATLAB [42] software was utilized to program the ANNs and the grey wolf optimizer algorithm. The GWO parameters employed for training the 169 ANNs are provided in Table 3.

3.3. Empirical Model Evaluation. As mentioned previously, 169 different ANN architectures with two hidden layers were trained using the GWO algorithm. The hyperbolic tangent function was the common activation function of the hidden layers in all the networks. The output layer's activation function was chosen as the identity function. For simplicity, in the following sections, we will refer to ANNs as ANN 2L ($n_1 - n_2$), where 2L indicates two hidden layers and n_1 and n_2 define the number of neurons in the first and second hidden layers, respectively.

The top 10 models among the 169 trained models were selected based on their MSE values. The ANNs and their training and testing data results are provided in Tables 4 and 5. Figure 5 offers an illustration of the association between the network's architecture and its performance measure and the RMSE graph of the investigated architectures based on the number of neurons in the first and second hidden layers.

According to Tables 4 and 5, the ANN 2L (7-13) and ANN 2L (9-6) models result in the lowest MSE and RMSE values for the training and testing data. The MSE on test data of ANN 2L (9-6) is less than ANN 2L (7-13), suggesting better generalization capability; however, since the performance of ANN 2L (7-13) is consistent for training and testing, it is chosen as the top model trained using the GWO algorithm for additional investigation. On training data, ANN 2L (7-13) has MAE, MSE, EF, and VAF values of 3.53, 27.55, 0.95, and 0.95, respectively. For testing data, ANN 2L (7-13) has MAE, MSE, EF, and VAF values of 3.43, 27.10, 0.94, and 0.94, respectively. It is noteworthy that the error metrics for training and testing the data were computed based on their original values and not their normalized range of $[-1, 1]$, as is sometimes used in literature.

Figure 6 shows the amount of RMSE for the top three GWO-ANN models, and Figures 7 and 8 offer a graphical illustration of the ANN 2L (7-13) model's performance, including comparing the empirical model's predicted and experimental values for the training, testing, and all data, respectively. The predicted values are plotted near the $y=x$ line, which confirms the model's accuracy.

The floating bar chart of the prediction errors of the ANN 2L (7-13) on testing data is depicted in Figure 9. For most samples, the prediction error is less than 5 MPa, and for

TABLE 3: Different variables in the GWO algorithm.

Parameter	Value
Max generations	500
Search agents	30

TABLE 4: Statistical indexes of the top 10 models in the neural networks on training data.

Network designation	ME	MAE	MSE	RMSE	AAE	EF	VAF (%)
ANN-GWO (7-13)	0.387	3.53	27.55	5.249	0.08	0.95	0.95
ANN-GWO (12-4)	-0.420	4.37	33.17	5.760	0.11	0.94	0.94
ANN-GWO (9-6)	-0.120	4.21	36.31	6.025	0.10	0.93	0.93
ANN-GWO (8-3)	0.002	3.72	40.34	6.352	0.08	0.93	0.92
ANN-GWO (7-3)	0.713	4.99	46.48	6.818	0.11	0.91	0.91
ANN-GWO (11-4)	-0.390	5.54	51.50	7.177	0.12	0.90	0.90
ANN-GWO (7-2)	0.171	5.43	57.41	7.577	0.12	0.89	0.89
ANN-GWO (7-5)	0.550	6.09	65.40	8.087	0.13	0.88	0.88
ANN-GWO (11-9)	0.597	5.79	67.50	8.216	0.14	0.87	0.87
ANN-GWO (12-12)	-0.280	6.46	73.13	8.551	0.14	0.86	0.86

other less accurate predictions, the prediction error is less than about 15 MPa; however, for one test sample, the error is about 25 MPa. The ANN 2L (7-13) empirical model is chosen for further analysis.

4. Comparison of Different Approaches

Two separate models were developed to assess the grey wolf optimizer algorithm's performance on training an ANN for predicting SCC compressive strength. For this purpose, an ANN was trained using the genetic algorithm and a multivariable linear regression model.

4.1. Genetic Algorithm Model. The 169 architectures provided in Table 3 were utilized for training the GA to identify the best performing ANN. Consequently, the highest-

TABLE 5: Statistical indexes of the top 10 models in the neural networks on testing data.

Network designation	ME	MAE	MSE	RMSE	AAE	EF	VAF (%)
ANN-GWO (7-13)	-0.47	3.43	27.10	5.206	0.07	0.94	0.94
ANN-GWO (12-4)	-0.38	4.66	44.88	6.699	0.09	0.90	0.90
ANN-GWO (9-6)	-0.42	3.33	22.16	4.708	0.06	0.95	0.95
ANN-GWO (8-3)	0.18	3.37	27.65	5.258	0.06	0.94	0.94
ANN-GWO (7-3)	0.25	4.89	41.75	6.462	0.09	0.90	0.90
ANN-GWO (11-4)	-0.57	4.63	38.49	6.204	0.09	0.91	0.91
ANN-GWO (7-2)	0.43	4.63	41.33	6.429	0.09	0.91	0.90
ANN-GWO (7-5)	0.83	5.97	78.57	8.864	0.11	0.82	0.82
ANN-GWO (11-9)	0.27	5.15	47.22	6.871	0.09	0.89	0.89
ANN-GWO (12-12)	-1.28	5.17	53.29	7.30	0.09	0.88	0.88

(a)

(b)

FIGURE 5: RMSE values for the two-hidden-layer ANN for (a) training data and (b) testing data.

FIGURE 6: RMSE for the top three models for GWO-ANN models.

FIGURE 7: Experimental versus predicted values of SCC's compressive strength for the ANN-GWO (7-13) model using training and testing data. (a) Training and (b) testing.

performing ANN had 11 and 9 neurons in the first and second hidden layers, respectively. To distinguish between ANN trained using GWO and GA, the networks will be designated with ANN-GWO 2L (7-13) and ANN-GA 2L (11-9) for GWO-trained and GA-trained networks, respectively.

The GA parameters used for the ANN-GA 2L (11-9) training were determined by trial and error (see Table 6). The floating bar chart of the prediction error of the GA-trained neural network is given in Figure 10. Compared to the GWO-trained neural network, this network has a much higher prediction error. The predicted versus experimental

values of the compressive strength of SCC are illustrated in Figure 11. The points on the plot are much farther from the $y=x$ line than the ANN-GWO 2L (7-13) model.

4.2. Multiple Regression Model. For simplicity, a multiple linear regression (MLR) model [43] was developed in Minitab software version 19 using the same dataset. Each of the variables' impact can be estimated using the regression coefficients [43, 44]. The final regression equation is as follows:

FIGURE 8: Experimental versus predicted values of SCC's compressive strength for the ANN-GWO (7-13) model using all data.

FIGURE 9: Floating bar chart of errors of ANN-GWO (7-13) model using testing data.

TABLE 6: Genetic algorithm parameters.

Parameter	Value
Max generations	150
Recombination (%)	15
Crossover (%)	50
Crossover method	Single point

FIGURE 10: Floating bar chart of errors of ANN-GA (7-13) model using testing data.

$$f'_c = -173.2 + (0.2804 \times C) + (0.1715 \times LP) + (0.1813 \times F) + (0.3068 \times GGBFS) + (0.3192 \times SF) + (0.3245 \times RHA) + (0.06691 \times CA) + (0.07217 \times FA) + (0.0145 \times W) - (2.852 \times SP) + (2.99 \times VMA). \quad (11)$$

In equation (11), C refers to cement, CA refers to coarse aggregate, FA refers to fine aggregate, W refers to water, LP refers to limestone powder, F refers to fly ash, GGBFS refers to ground granulated blast furnace slag, SF refers to silica fume, RHA refers to rice husk ash, SP refers to superplasticizers, VMA refers to viscosity modifying admixtures, and f'_c refers to compressive strength. The floating bar chart of the prediction error of the multiple linear regression

model on testing data is plotted in Figure 12. Compared to GWO-trained and GA-trained ANNs, the multiple linear regression model performs better than the GA-trained model but lacks the accuracy of the ANN-GWO 2L (7-13) model. Figures 11 and 13 show the experimental versus predicted values of ANN-GA (7-13) and MLR models based on testing data. The resulting predictions of the model are plotted near the $y=x$ line, depicting its high accuracy.

FIGURE 11: Compressive strength experimental versus predicted values for the ANN-GA (7-13) model using testing data.

FIGURE 12: Floating bar chart of errors of multiple linear regression model using testing data.

FIGURE 13: Compressive strength experimental versus predicted values for the multiple regression model using testing data.

The statistical indices of MAE, MSE, EF, and VAF for ANN-GWO 2L (7-13), ANN-GA 2L (11-9), and multiple linear regression models based on the training and testing data are provided in Tables 7 and 8, respectively. The highest-performing empirical model is ANN-GWO 2L (7-13), followed by the multiple linear regression and ANN-GA 2L (11-9).

For a visual comparison of the three models, the Taylor diagram plot is shown in Figure 14, and the predictions of the three models for the testing data are plotted in Figure 15.

4.3. Sensitivity Analysis. Given that the ANN-GWO 2L (7-13) model offers superior performance compared to the ANN-GA 2L (11-9) and multiple linear regression models, we conducted a sensitivity analysis to measure the relative influence of each of the 11 input parameters on SCC's compressive strength. For this purpose, we implemented the profile method suggested by Lek [45, 46] in MATLAB software [42]. The general idea behind this method is to vary each input variable while others are kept at fixed values. In this application, the range of input variables was divided into several equal intervals called the scale. The remaining variables were set to m different fixed values, the network's output was calculated for the entire range of the selected variable, and this resulted in m groups of outputs. Finally, the m groups of outputs were combined by calculating the median output for every input case. The fixed values utilized for each variable were their min, Q1, median, Q3, and max. For further details, please refer to the paper by Lek [44, 45], where they offer in-depth coverage of the implementation and theory of the procedure.

Figure 16 illustrates the relative importance and influence of the 11 input parameters on the response

variable (SCC compressive strength). The figure shows that the most influential parameter is the cement content, where 22.3% of the response variable variation is attributed to cement content. The next three most influential parameters are GGBFS, RHA, and fly ash, with 18.3%, 17.3%, and 11.6% of the response variable variation. The least influential parameters are fine aggregate (FA) content and viscosity modifying admixtures (VMAs) with 0.3% and 1.5% of the response variable variation attributed to them.

Since the relative importance of FA (fine aggregate), W (water), LP (limestone powder), and VMA (viscosity modifying admixtures) is small (less than 5%), they were removed from the model, and all 169 architectures were trained with 7 inputs instead of 11 as in the previous models. The statistics of the top 3 models selected based on small MSE for training and testing data are given in Tables 9 and 10.

The top-performing ANN-GWO (11-4) model has 114.99 and 178.12 for MSE of training and testing, respectively. Compared to 27.55 and 27.10 training and testing MSE of the 11-input ANN-GWO (7-13) model, these values are higher and indicate less accuracy than 7-input models. Although 7-input models are simpler, they will not be considered in the following sections.

4.4. The Predictive Model and ANN Weights. This study's top two empirical models are ANN-GWO 2L (7-13) and multiple linear regression (MLR) models. MLR can be efficiently utilized; however, the ANN-GWO 2L (7-13) model would not be applicable unless the source code is provided. Hence, the weights and biases of the ANN-GWO model are provided in this section. As noted

TABLE 7: Training data statistics of the ANN-GWO, ANN-GA, and multiple linear regression models.

Network designation	ME	MAE	MSE	RMSE	AAE	EF	VAF (%)
ANN-GWO (7-13)	0.387	3.53	27.55	5.25	0.08	0.95	95
ANN-GA (11-9)	0.180	12.24	238.04	15.43	0.27	0.45	45
Multiple linear regression	0.210	9.64	135.42	11.64	0.19	0.69	69

TABLE 8: Testing data statistics of the ANN-GWO, ANN-GA, and multiple linear regression models.

Network designation	ME	MAE	MSE	RMSE	AAE	EF	VAF (%)
ANN-GWO (7-13)	-0.47	3.43	27.10	5.206	0.07	0.94	94
ANN-GA (11-9)	2.38	13.92	296.79	17.23	0.39	0.45	46
Multiple linear regression	-0.57	8.68	104.96	10.24	0.20	0.80	80

FIGURE 14: Comparison between models using normal distribution.

FIGURE 15: Prediction comparison of all models using testing data.

FIGURE 16: The relative influence of the input parameters on SCC's compressive strength.

TABLE 9: Statistics of top 3 seven-input ANN-GWO models on training data.

Network designation	ME	MAE	MSE	RMSE	AAE	EF	VAF (%)
ANN-GWO (11-4)	-0.65	8.44	114.99	10.72	0.18	0.73	74
ANN-GWO (7-5)	-0.42	10.15	151.70	12.32	0.21	0.65	65
ANN-GWO (11-9)	-0.69	8.46	115.76	10.76	0.17	0.73	73

TABLE 10: Statistics of top 3 seven-input ANN-GWO models on testing data.

Network designation	ME	MAE	MSE	RMSE	AAE	EF	VAF (%)
ANN-GWO (11-4)	0.00	9.59	178.12	13.35	0.22	0.67	67
ANN-GWO (7-5)	0.21	11.08	213.68	14.62	0.27	0.60	60
ANN-GWO (11-9)	1.16	10.38	220.29	14.84	0.25	0.59	59

previously, the ANNs input data for each variable must be initially normalized by equation (8) and the max/min values in Table 1. Also, the network output must be denormalized using equation (15). The network input is an 11×1 vector entitled $\mathbf{a}^{(1)}$. The SCC's compressive strength is computed as follows:

$$\mathbf{a}^{(2)} = \tanh(\mathbf{g}^{(1)} \times \mathbf{a}^{(1)} + \mathbf{b}_1), \quad (12)$$

$$\mathbf{a}^{(3)} = \tanh(\mathbf{g}^{(2)} \times \mathbf{a}^{(2)} + \mathbf{b}_2), \quad (13)$$

$$f_c^{\text{predict(normalized)}} = \tanh(\mathbf{g}^{(3)} \times \mathbf{a}^{(3)} + b_3), \quad (14)$$

$$f_c^{\text{predict}} = \frac{f_c^{\text{predict(normalized)}} + 1}{2} \times (f_{\max} - f_{\min}) + f_{\min}, \quad (15)$$

where $\tanh$ is the hyperbolic tangent function, $f_c^{\text{predicted}}$ is the predicted compressive strength value, $f_{\max}$ is the maximum compressive strength, $f_{\min}$ is the minimum compressive strength, and weight (θ) and bias (b) matrices are given as follows:

$$\begin{aligned}
g^{(1)} &= 10^{-1} \times \begin{bmatrix} -0.6194 & -0.2206 & -0.0151 & 6.1881 & 0.1487 & -0.9968 & 2.0303 & 0.0873 & -0.7791 & -1.0420 & -1.6074 \\ 0.8284 & 0.8310 & 2.0030 & 1.0801 & 6.0824 & 4.0657 & 1.1979 & 1.5929 & -5.9237 & -3.5269 & 0.4059 \\ 2.0025 & 8.8626 & 0.1649 & 1.7660 & 0.2993 & -0.5561 & 0.1163 & -0.2132 & -0.0472 & -1.3294 & 0.0545 \\ -4.2205 & -3.7997 & 0.2685 & -0.0024 & 9.1619 & 9.8318 & -4.1632 & 1.8509 & -0.0027 & 9.7644 & -4.3976 \\ 6.7981 & -0.6536 & 3.8180 & 0.5124 & -0.0010 & 8.3232 & -0.2558 & 0.1766 & -0.0206 & -5.9788 & 2.0385 \\ 0.3487 & 4.0695 & -8.9906 & -7.5046 & -0.6353 & -0.1641 & 0.7121 & -0.0754 & 0.4580 & -0.6695 & 3.4874 \\ 1.2532 & -0.0059 & 3.0367 & 1.3686 & -8.5865 & 0.1229 & -0.7933 & 0.0066 & 0.8516 & -9.0864 & 1.2612 \end{bmatrix}, \\
g^{(2)} &= 10^{-1} \times \begin{bmatrix} 7.7662 & -0.0075 & -5.5702 & -2.3893 & 8.9517 & -0.3040 & -10.0000 \\ 2.6686 & -2.0868 & -1.1851 & 0.0051 & 2.0922 & 0.2481 & 7.2222 \\ 5.1130 & 1.2574 & 0.7545 & -1.6037 & 9.0786 & 0.1501 & -2.2145 \\ 0.0021 & -0.8814 & 0.0225 & -0.1633 & -0.0855 & 3.6363 & 0.5126 \\ -1.0368 & -0.3596 & 9.9966 & -0.2479 & 0.5489 & -0.4463 & -0.2311 \\ 6.5330 & -0.3197 & 0.7840 & 0.2275 & -3.7988 & -0.4614 & 2.3398 \\ 0.4126 & 0.3222 & -1.6344 & 0.3127 & 6.2291 & -0.8820 & 0.4210 \\ 2.8054 & -0.1841 & 7.6013 & 8.2659 & -0.0482 & -6.6024 & -1.4229 \\ -0.0056 & 9.9021 & 5.4498 & -0.0435 & -1.9104 & 0.7023 & -0.0953 \\ 0.3688 & -2.1788 & 0.1478 & -1.2148 & -0.8717 & 4.8577 & -3.5485 \\ -4.2497 & 1.3134 & 1.1919 & -0.1691 & -1.8386 & -0.1244 & 0.5231 \\ 0.7077 & -1.1718 & 0.0225 & 0.4649 & 0.0413 & 4.3178 & 0.7039 \\ -0.1877 & 4.8291 & 0.4972 & -0.0089 & 0.5178 & -0.9476 & 0.0715 \end{bmatrix}, \\
g^{(3)} &= 10^{-1} \times [8.2740 \ 0.0320 \ 8.8125 \ -0.1908 \ 9.8406 \ 0.6408 \ 0.6793 \ -0.4794 \ -0.6800 \ -0.4178 \ -0.0662 \ -2.7360 \ -0.6860], \\
b_1 &= 10^{-1} \times [0.0629 \ 0.4745 \ -0.1349 \ -0.1407 \ 5.7461 \ 1.1466 \ -9.9939]^T, \\
b_2 &= 10^{-1} \times [0.2021 \ -0.0738 \ 0.0027 \ 4.7061 \ 6.3830 \ -0.6189 \ -0.1089 \ 7.3148 \ 4.5942 \ -0.0949 \ 0.3958 \ -0.0608 \ -0.5493]^T, \\
b_3 &= 0.0620.
\end{aligned} \tag{16}$$

5. Conclusion

This paper aimed to predict the compressive strength of self-compacting concrete (SCC). To this end, a total of 205 experimental results were obtained from the literature. The grey wolf optimizer (GWO) algorithm was employed to train the artificial neural network models. The following points were concluded:

- (1) For predicting the compressive strength of self-compacting concrete, ANN-GWO 2L (7-13) model provides superior performance compared to other ANN models. The model's Mean Squared Error and efficiency using test data were 27.10 and 0.94, respectively.
- (2) A less accurate, but easy to implement, multiple regression model is provided to simplify the prediction process.
- (3) Benchmarking the ANN-GWO 2L (7-13), ANN-GA 2L (11-9), and multiple regression models portrays the ANN-GWO's superior accuracy, followed by the multiple regression and ANN-GA models.
- (4) The sensitivity analysis on the ANN-GWO 2L (7-13) model suggests that the most influential parameters

on compressive strength of self-compacting concrete are cement content, ground granulated blast furnace slag, rice husk ash, and fly ash. The fine aggregate content was the least influential factor.

- (5) The weights and biases of the best performing model, ANN-GWO 2L (7-13), are extracted, and a predictive matrix-based equation is developed for predicting the compressive strength of self-compacting concrete.

Data Availability

The datasets are available in Table 1 and at <https://link.springer.com/article/10.1007/s00521-017-3007-7>.

Conflicts of Interest

The authors declare no conflicts of interest regarding this paper.

References

- [1] P. G. Asteris and K. G. Kolovos, "Self-compacting concrete strength prediction using surrogate models," *Neural Computing & Applications*, vol. 31, pp. 409–424, 2019.

- [2] H. E. C. Moncef Nehdi and M. Hesham El Naggar, "Predicting performance of self-compacting concrete mixtures using artificial neural networks," *ACI Materials Journal*, vol. 98, 2001.
- [3] X. Ding, C. Li, Y. Li, Y. Lu, C. Song, and S. Zhao, "Experimental and numerical study on stress-strain behavior of self-compacting SFRC under uniaxial compression," *Construction and Building Materials*, vol. 185, pp. 30–38, 2018.
- [4] J. Li, C. Wu, H. Hao, and Y. Su, "Experimental and numerical study on steel wire mesh reinforced concrete slab under contact explosion," *Materials & Design*, vol. 116, pp. 77–91, 2017.
- [5] D. Sathyan, K. B. Anand, A. J. Prakash, and B. Premjith, "Modeling the fresh and hardened stage properties of self-compacting concrete using random kitchen sink algorithm," *International Journal of Concrete Structures and Materials*, vol. 12, p. 24, 2018.
- [6] H. Mashhadban, S. S. Kutanaei, and M. A. Sayarinejad, "Prediction and modeling of mechanical properties in fiber reinforced self-compacting concrete using particle swarm optimization algorithm and artificial neural network," *Construction and Building Materials*, vol. 119, pp. 277–287, 2016.
- [7] M. Uysal and H. Tanyildizi, "Estimation of compressive strength of self compacting concrete containing polypropylene fiber and mineral additives exposed to high temperature using artificial neural network," *Construction and Building Materials*, vol. 27, pp. 404–414, 2012.
- [8] J. Zhang, G. Ma, Y. Huang, J. Sun, F. Aslani, and B. Nener, "Modelling uniaxial compressive strength of lightweight self-compacting concrete using random forest regression," *Construction and Building Materials*, vol. 210, pp. 713–719, 2019.
- [9] A. Hasanzade-Inallu, P. Zarfam, and M. Nikoo, "Modified imperialist competitive algorithm-based neural network to determine shear strength of concrete beams reinforced with FRP," *Journal of Central South University*, vol. 2019, pp. 3156–3174, 2019.
- [10] L. Sadowski, M. Nikoo, and M. Nikoo, "Concrete compressive strength prediction using the imperialist competitive algorithm," *Computers and Concrete*, vol. 22, pp. 355–363, 2018.
- [11] P. G. Asteris and V. G. Mokos, "Concrete compressive strength using artificial neural networks," *Neural Computing & Applications*, vol. 32, pp. 11807–11826, 2020.
- [12] P. G. Asteris, M. Apostolopoulou, A. D. Skentou, and A. Moropoulou, "Application of artificial neural networks for the prediction of the compressive strength of cement-based mortars," *Computers and Concrete*, vol. 24, no. 4, 2019.
- [13] B. K. R. Prasad, H. Eskandari, and B. V. V. Reddy, "Prediction of compressive strength of SCC and HPC with high volume fly ash using ANN," *Construction and Building Materials*, vol. 23, pp. 117–128, 2009.
- [14] R. Siddique, P. Aggarwal, and Y. Aggarwal, "Prediction of compressive strength of self-compacting concrete containing bottom ash using artificial neural networks," *Advances in Engineering Software*, vol. 42, no. 10, 2011.
- [15] M. Uysal and H. Tanyildizi, "Predicting the core compressive strength of self-compacting concrete (SCC) mixtures with mineral additives using artificial neural network," *Construction and Building Materials*, vol. 25, pp. 4105–4111, 2011.
- [16] P. G. Asteris, K. G. Kolovos, M. G. Douvika, and K. Roinos, "Prediction of self-compacting concrete strength using artificial neural networks," *European Journal of Environmental and Civil Engineering*, vol. 20, pp. s102–s122, 2016.
- [17] B. G. Aiyer, D. Kim, N. Karingattikkal, P. Samui, and P. R. Rao, "Prediction of compressive strength of self-compacting concrete using least square support vector machine and relevance vector machine," *KSCE Journal of Civil Engineering*, vol. 18, pp. 1753–1758, 2014.
- [18] A. R. Ghanizadeh, H. Abbaslou, A. T. Amlashi, and P. Alidoust, "Modeling of bentonite/sepiolite plastic concrete compressive strength using artificial neural network and support vector machine," *Frontiers of Structural and Civil Engineering*, vol. 13, pp. 215–239, 2019.
- [19] P. G. Asteris, A. Ashrafiyan, and M. Rezaie-Balf, "Prediction of the compressive strength of self-compacting concrete using surrogate models," *Computers and Concrete*, vol. 24, no. 2, 2019.
- [20] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [21] A. Hasanzade-Inallu, "Grey wolf optimizer-based ANN to predict compressive strength of AFRP-confined concrete cylinders," *Soil Structure Interactions*, vol. 3, pp. 23–32, 2018.
- [22] G. Ellis, "Feed-forward," in *Control System Design Guide*-Elsevier, Amsterdam, Netherlands, 2004.
- [23] M. Nikoo, B. Aminnejad, and A. Lork, "Predicting shear strength in FRP-reinforced concrete beams using Bat algorithm-based artificial neural network," *Advances in Materials Science and Engineering*, vol. 2021, Article ID 5899356, 2021.
- [24] S. Shekhar, M. B. Amin, and P. Khandelwal, *Generalization Performance of Feed-Forward Neural Networks*, E. B. T.-N. N. Gelenbe, Ed., North-Holland, Amsterdam, Netherlands, 1992.
- [25] M. Fan, Z. Zhang, and C. Wang, "Optimization method for load frequency feed forward control," in *Mathematical Models and Algorithms for Power System Optimisation* Elsevier, Amsterdam, Netherlands, 2019.
- [26] S. Haykin, *Neural Networks and Learning Machines*, Pearson Prentice Hall, New Jersey, NJ, USA, 2008.
- [27] A. Géron, *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, Inc., Newton, MA, USA, 2017.
- [28] M. Nikoo, L. Sadowski, F. Khademi, and M. Nikoo, "Determination of damage in reinforced concrete frames with shear walls using self-organizing feature map," *Applied Computational Intelligence and Soft Computing*, vol. 2017, pp. 1–10, 2017.
- [29] L. D. Mech, "Alpha status, dominance, and division of labor in wolf packs," *Canadian Journal of Zoology*, pp. 1196–1203, 1999.
- [30] C. Muro, R. Escobedo, L. Spector, and R. P. Coppinger, "Wolf-pack (*Canis lupus*) hunting strategies emerge from simple rules in computational simulations," *Behavioural Processes*, vol. 88, pp. 192–197, 2011.
- [31] H. J. H. Brouwers and H. J. Radix, "Self-compacting concrete: theoretical and experimental study," *Cement and Concrete Research*, vol. 35, no. 11, 2005.
- [32] P. Dinakar, K. P. Sethy, and U. C. Sahoo, "Design of self-compacting concrete with ground granulated blast furnace slag," *Materials & Design*, vol. 43, pp. 161–169, 2013.
- [33] E. Güneyisi, M. Gesoglu, O. A. Azez, and H. Ö. Öz, "Effect of nano silica on the workability of self-compacting concretes having untreated and surface treated lightweight aggregates," *Construction and Building Materials*, vol. 115, pp. 371–380, 2016.
- [34] S. A. Memon, M. A. Shaikh, and H. Akbar, "Utilization of rice husk ash as viscosity modifying agent in self compacting concrete," *Construction and Building Materials*, vol. 25, pp. 1044–1048, 2011.

- [35] M. E. Rahman, A. S. Muntohar, V. Pakrashi, B. H. Nagaratnam, and D. Sujun, "Self compacting concrete from uncontrolled burning of rice husk and blended fine aggregate," *Materials & Design*, vol. 55, pp. 410–415, 2014.
- [36] M. Şahmaran, İ. Ö. Yaman, and M. Tokyay, "Transport and mechanical properties of self consolidating concrete with high volume fly ash," *Cement and Concrete Composites*, vol. 31, pp. 99–106, 2009.
- [37] B. Sukumar, K. Nagamani, and R. Srinivasa Raghavan, "Evaluation of strength at early ages of self-compacting concrete with high volume fly ash," *Construction and Building Materials*, vol. 22, pp. 1394–1401, 2008.
- [38] H. Zhao, W. Sun, X. Wu, and B. Gao, "The properties of the self-compacting concrete with fly ash and ground granulated blast furnace slag mineral admixtures," *Journal of Cleaner Production*, vol. 95, pp. 66–74, 2015.
- [39] J. Li and A. D. Heap, *A Review of Spatial Interpolation Methods for Environmental Scientists*, Australian Government, Canberra, Australia, 2008.
- [40] A. Hasanzade-Inallu, "Pouya hassanzadeh Inallou, babak Eskandarinezhad, prediction of compressive strength of concrete with manufactured sand using neural networks and Bat algorithm," *Soil Structure Interactions*, vol. 4, pp. 52–63, 2019.
- [41] G. J. Bowden, G. C. Dandy, and H. R. Maier, "Input determination for neural network models in water resources applications. Part 1—background and methodology," *Journal of Hydrology*, vol. 301, pp. 75–92, 2005.
- [42] MATLAB: The MathWorks, Natick, MA, USA, 2018.
- [43] M. Nikoo, F. Torabian Moghadam, and Ł. Sadowski, "Prediction of concrete compressive strength by evolutionary artificial neural networks," *Advances in Materials Science and Engineering*, vol. 2015, Article ID 849126, 9 pages, 2015.
- [44] D. K. Panesar, M. Aqel, D. Rhead, and H. Schell, "Effect of cement type and limestone particle size on the durability of steam cured self-consolidating concrete," *Cement and Concrete Composites*, vol. 80, 2017.
- [45] M. Gevrey, I. Dimopoulos, and S. Lek, "Review and comparison of methods to study the contribution of variables in artificial neural network models," *Ecological Modelling*, vol. 160, no. 3, 2003.
- [46] S. Lek, M. Delacoste, P. Baran, I. Dimopoulos, J. Lauga, and S. Aulagnier, "Application of neural networks to modelling nonlinear relationships in ecology," *Ecological Modelling*, vol. 90, pp. 39–52, 1996.