

Article

# Predicting Compressive Strength of Concrete Containing Recycled Aggregate Using Modified ANN with Different Optimization Algorithms

Amirreza Kandiri <sup>1</sup>, Farid Sartipi <sup>2</sup> and Mahdi Kioumarsi <sup>3,\*</sup>

<sup>1</sup> Department of Civil Engineering, Science and Research Branch, Azad University, Tehran 1477893855, Iran; amir.kandiri@iconsmat.com.au

<sup>2</sup> Institute of Construction Material, Sydney, NSW 2747, Australia; farid.sartipi@iconsmat.com.au

<sup>3</sup> Department of Civil Engineering and Energy Technology, OsloMet–Oslo Metropolitan University, 0166 Oslo, Norway

\* Correspondence: mahdik@oslomet.no; Tel.: +47-67-23-87-45

**Abstract:** Using recycled aggregate in concrete is one of the best ways to reduce construction pollution and prevent the exploitation of natural resources to provide the needed aggregate. However, recycled aggregates affect the mechanical properties of concrete, but the existing information on the subject is less than what the industry needs. Compressive strength, on the other hand, is the most important mechanical property of concrete. Therefore, having predictive models to provide the required information can be helpful to convince the industry to increase the use of recycled aggregate in concrete. In this research, three different optimization algorithms including genetic algorithm (GA), salp swarm algorithm (SSA), and grasshopper optimization algorithm (GOA) are employed to be hybridized with artificial neural network (ANN) separately to predict the compressive strength of concrete containing recycled aggregate, and a M5P tree model is used to test the efficiency of the ANNs. The results of this study show the superior efficiency of the modified ANN with SSA when compared to other models. However, the statistical indicators of the hybrid ANNs with SSA, GA, and GOA are so close to each other.

**Keywords:** concrete; compressive strength; artificial neural network; genetic algorithm; salp swarm algorithm; grasshopper optimization algorithm; M5P tree

**Citation:** Kandiri, A.; Sartipi, F.; Kioumarsi, M. Predicting Compressive Strength of Concrete Containing Recycled Aggregate Using Modified ANN with Different Optimization Algorithms. *Appl. Sci.* **2021**, *11*, 485. <https://doi.org/10.3390/app11020485>

Received: 15 November 2020

Accepted: 31 December 2020

Published: 6 January 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Every year, a massive proportion of construction and demolished waste (C&DW) is produced all around the world. For instance, the USA produces around 250 million tons [1], China produces almost two billion tons [2], and the European Union is responsible for about 900 million tons of C&DW [3]. Using C&DWs as recycled aggregate has two benefits. First, it reduces the amount of C&DW that needs to be disposed. Second, it reduces the demand for regular aggregate.

Recycled aggregate as an environmental alternative in concrete projects is a promising environmental solution to reduce the carbon footprint of construction projects. Sourced from C&DW, it comes with varying properties depending on the type of the structure being demolished/built. As in some buildings, the dominant material used might be timber, concrete, or steel which ultimately changes the portion of recycled aggregate. The properties of aggregates vary a lot depending on their source and the manufacturing process. The influence of these two factors, i.e., the source and the process, is mainly on the porosity, water absorption, and saturated surface dried ratio, all of which significantly affect the final performance of concrete, made with these aggregates [3]. In most of the cases, a poor mechanical performance results after using recycled aggregate

in concrete and that is the major reason behind the lack of tendency from the consumer's perspective in using more of this environmental alternative [4–7].

As the amount of recycled aggregate that used in concrete increases, the elastic modulus, compressive strength and density of the concrete decrease [8–11]. However, the use of recycled aggregate is allowed in specific amounts in some standards [8]. On the other hand, in case of service life and durability, there are some studies that showed these properties of concrete with recycled aggregate are not as good as those of conventional concrete [12,13] while others concluded that concrete with recycled aggregate has better durability than conventional concrete [14]. While there is no commercial approach for the use of high-quality recycled aggregate, the consumers can still be motivated to use more of recycled aggregate upon the availability of more data.

Artificial intelligence and data mining techniques have proven to be solutions to provide the end users with a reliable information in regards to the mechanical properties of concrete. Recently, predictive models based on artificial intelligence methods have been used to forecast the mechanical properties of concrete and predict the effect of changes in its mixture design on its behavior [15–57]. For instance, Behnood and Golafshani [37], has used M5P three model to estimate mechanical properties of concrete containing used foundry sand. Their results show that the M5P three model has an acceptable performance to predict the properties of concrete. Zhang et al. [23] have developed a model by hybridizing beetle antennae search (BAS) and multi-output least square support vector regression (MOLSSVR) to predict concrete's compressive strength (CS) and permeability coefficient of pervious. The outcomes of this research indicated that their proposed model performs better than support vector regression (SVR), MOLSSVR, logistic regression, and modified ANN with firefly algorithm. In another research, Ashrafiyan et al. [54] used chi-square automatic interaction detection, M5P tree model, M5rule, and random forest to forecast mechanical properties of roller-compacted concrete pavement. It was observed that the predicted values of random forest had lower errors and higher correlation in comparison with other models. Kandiri et al. [40] developed a hybrid ANN with multi-objective salp swarm algorithm to estimate CS of concrete with ground granulated blast furnace slag. In that research, it was showed that 13 out of 19 ANNs outperformed M5P tree model. In another Article, by hybridizing ANFIS and ANN with grey wolf optimizer, Golafshani et al. [49] showed that an ANN-based model was more efficient than a modified ANFIS. Recently, probabilistic approaches have been used to obtain the uncertainties that influence concrete characteristics [58–61]. For instance, Ramezani et al. [58] used the Kelly-Tyson theory to propose a probabilistic model for predicting the flexural strength of carbon nanotubes–cement nanocomposites. The results showed that the proposed model is able to obtain the experimental observation with an acceptable performance.

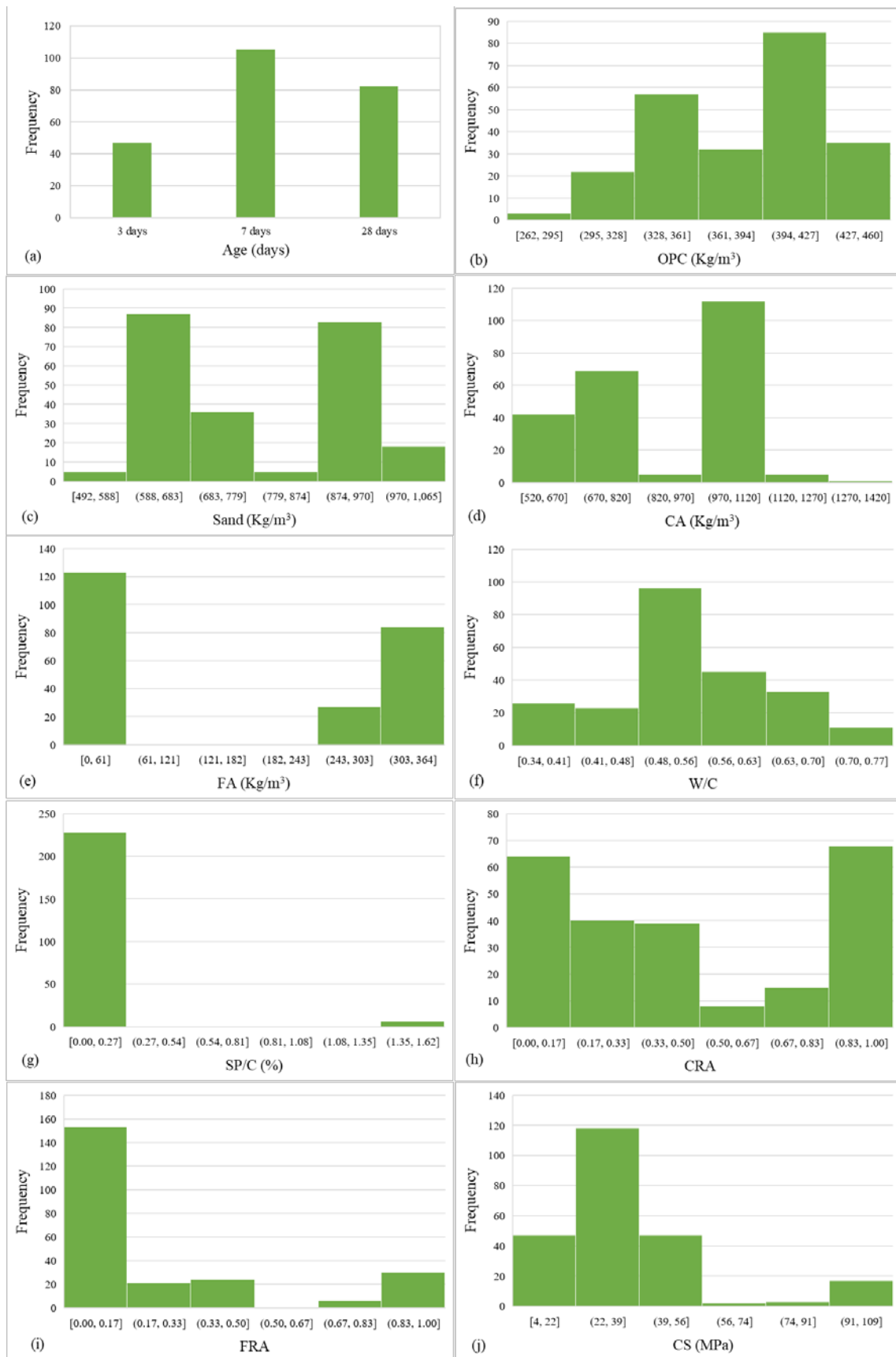
However, ANN's performance is directly affected by its architecture. The number of its hidden layers and neurons in them is constant for each ANN. In fact, once the network is set, these parameters are unchangeable. Usually, these parameters are obtained by trial-and-error method, in which a lot of time is spent on reaching the optimum architecture, and sometimes obtaining the optimum architecture is impossible. Therefore, an algorithm that can optimize an ANN architecture is needed. In this study, ANN's architecture is optimized with three optimization algorithms including grasshopper optimization, salp swarm algorithm, and genetic algorithm. Then, the results of these models are compared with M5P tree model, which is a classification-based regression model. To develop the models, data had been collected from 12 different scientific resources all with mutual mixing and experimental methods to avoid inconsistency of the output analysis [62–71]. The next section explains the data gathering procedure followed by a section of describing optimization methods, ANN, the modified proposed models, and M5P three model. After that, the results are analyzed and compared with each other.

## 2. Data Gathering

For developing a predictive model, there is a need for a valid dataset. The dataset needs to include all the parameters, which are possible to be effective on the outputs as inputs, except those that are constant among all records. In the present research, a dataset including 234 patterns was collected from literature [62–68,70–75]. In all patterns, the curing condition were the same, and the used cement was ordinary Portland cement (OPC). Moreover, the superplasticizer (SP) in all mixes was Polycarboxylic, and the type of specimen in all recorded patterns were cube with the same size (100 mm by 100 mm). The inputs of the models in this study are the testing age (TA), the amount of OPC, sand (S), coarse aggregate (CA), fine aggregate (FA), the ratio of water to cement (W/C), superplasticizer to cement by percentage (SP/C), the ratio of recycled coarse aggregate to total coarse aggregate (CRA), and the ratio of recycled fine aggregate to total fine aggregate (FRA). The only output variable of the models is the CS. Figure 1 shows the histogram of the inputs and output variables. Furthermore, the anatomical statistics of the gathered dataset are represented in Table 1.

**Table 1.** Anatomical statistics of the inputs and output.

Variable	Min	Max	Mean	Standard Deviation	Skewness	Kurtosis
OPC (kg/m <sup>3</sup> )	262.00	460.00	386.68	39.35	−0.38	−0.07
W/C	0.34	0.77	0.55	0.09	−0.27	−0.02
CA (kg/m <sup>3</sup> )	520.00	1295.00	877.55	222.53	−0.13	−1.66
FA (kg/m <sup>3</sup> )	0.00	364.00	154.98	165.06	0.16	−1.93
CRA	0.00	1.00	0.47	0.40	0.19	−1.53
FRA	0.00	1.00	0.21	0.35	1.42	0.45
Sand (kg/m <sup>3</sup> )	492.00	1065.00	781.96	144.13	0.24	−1.41
SP/C (%)	0.00	1.62	0.04	0.26	6.04	34.79
Age (Days)	3.00	28.00	13.56	10.74	0.57	−1.60



**Figure 1.** Output and input variables' histogram: (a) ages of samples, (b) the amount of OPC, (c) the amount of sand, (d) the amount of coarse aggregate, (e) the amount of fine aggregate, (f) the ratio of water to cement, (g) the ratio of super-plasticizer to cement by percentage, (h) the ratio of recycled coarse aggregate to total coarse aggregate, (i) the ratio of recycled fine aggregate to total fine aggregate, and (j) compressive strength.

### 3. Optimization Algorithms

Big optimization problems with many parameters cannot be solved by accurate mathematical methods. Therefore, heuristic and metaheuristic algorithms are served to find the best answer possible in a convenient time. Metaheuristic algorithms that are developed inspired by nature are the most popular ones [30,40]. Most of these algorithms include exploration and exploitation. In the exploration phase the possible solutions that are far from each other are studied. In the exploitation phase, the possible solutions that are close are checked. In fact, exploration phase helps the algorithm to avoid local optimums.

#### 3.1. Genetic Algorithm (GA)

The genetic algorithm is a metaheuristic algorithm for solving optimization problems which was proposed by John Holland [76]. In this algorithm, data are recorded in chromosomes, and each individual's chromosomes is allocated to a pattern. Every two individuals make two offspring by crossover and mutation methods. GA includes mutation and crossover that are responsible for exploration and exploitation phases, respectively. A number of individuals equal to the number of the initial population are going to survive based on their fitness functions. In other words, the fitness value of each individual (parents and offspring) are calculated and those who have better fitness value have better chances to live. In this study, the survivors are selected by roulette wheel. In this method, the chance of surviving chance is defined as follow:

$$X_k = \frac{1 - \frac{1}{F_k}}{n} \quad (1)$$

where,  $F_k$  is the fitness function for the  $k$ th individual,  $X_k$  is the probability of choosing  $k$ th individual,  $n$  is the population size, and it is obvious that  $\sum_{k=1}^n X_k = 1$ . Algorithm 1 represents different steps of a GA algorithm. This algorithm has been used to train an ANN in a number of previous studies [17,32,44,47,57,77].

---

#### Algorithm 1. Pseudocode of genetic algorithm.

---

- 1: Generate initial population
  - 2:  $i = 1$
  - 3: **while** ( $i < \text{number of iterations}$ )
  - 4:   Calculate the fitness function of the each individual
  - 5:   Select parents
  - 6:   Recombine pairs of parents
  - 7:   Create offspring from parents via crossover
  - 8:   Apply mutation to offspring
  - 9: **end while**
  - 10: Return the individuals that obtain the best fitness
- 

#### 3.2. Salp Swarm Algorithm (SSA)

Salps belong to Salpidae group, which are look like jellyfishes, with a body shape like transparent barrel [78]. Usually, they tend to create a chain to have fast coordination for finding food. SSA is inspired by their swarm intelligence. The salp chain has a leader, which is the salp at the front, and followers, which are the rest of the group. The leader is responsible for exploration and the followers handle the exploitation phase. Each salp has a position, which is defined in a search space with  $n$  dimensions, where  $n$  is the decision variables' number in the optimization problem. The following equation updates the leader position.

$$P_r^1 = \begin{cases} FP_j + d_1((UB_r - LB_r)d_2 + LB_r) & d_3 \geq 0 \\ FP_j - d_1((UB_r - LB_r)d_2 + LB_r) & d_3 < 0 \end{cases} \quad (2)$$

where  $P_r^1$  is  $r$ th dimension of the leader position,  $FP_j$  is the position of the food,  $UB_r$  and  $LB_r$  are the upper and the lower bounds in the  $r$ th dimension, respectively. In addition,  $d_1$  balances exploration and exploitation phases;  $d_2$  and  $d_3$  are random numbers in  $[0,1]$ .  $d_1$  is calculated as follows:

$$d_1 = 2e^{-\left(\frac{4t}{T}\right)^2} \quad (3)$$

where  $T$  is the maximum number of iterations and  $t$  represents the current iteration. The position of the followers are calculated as follows:

$$V_r^i = \frac{1}{2}(V_r^i + V_r^{i-1}) \quad (4)$$

where  $V_r^i$  is position of the  $i$ th salp in the  $r$ th dimension. Now, it is possible to simulate the salp swarm. Algorithm 2 illustrates pseudo cod of SSA. This algorithm has been used to train an ANN in a number of previous studies [35,40,43,79].

---

**Algorithm 2.** Pseudocode of SSA.

---

- 1: Initiate salp population considering the upper and lower bounds
  - 2:  $i = 1$
  - 3: **while** ( $i \leq$  number of iterations)
  - 4:   Calculate fitness function for each salp
  - 5:   Choose the best salp as the food source
  - 6:   Determine the value of  $d_1$  using Equation (3)
  - 7:   **for** each salp
  - 8:     **if** ( $i = 1$ )
  - 9:       Update the position of the leading salp by Equation (2)
  - 10:    **else**
  - 11:      Update the position of the follower salp by Equation (4)
  - 12:    **end**
  - 13: **end**
  - 14: Correct the positions of the salps considering upper and lower bounds
  - 15: **end**
  - 16: Return the food source
- 

### 3.3. Grasshopper Optimization Algorithm (GOA)

Grasshoppers are kinds of insects that are observed individually in nature. They live in a huge swarm in both nymph and adulthood phases [80,81]. In contrast to their nymph phase in which their swarm takes move in small steps, in adulthood phase they take long-range steps. The GOA algorithm is defined based on their behavior where the small steps are considered for exploitation phase and big steps are used for exploration phase. Like SSA, every grasshopper search agent has a position ( $X_j$ ) made of  $n$ -dimension, which is defined as follows:

$$X_j = c_1S_j + c_2Z_j + c_3W_j \quad (5)$$

where  $S_j$ ,  $Z_j$ , and  $W_j$  are, respectively, social interaction, gravity force, and wind advection on the  $j$ th search agent, and  $c_1$ ,  $c_2$ , and  $c_3$  are random number between zero and one to make random behavior. The following equation discuss the social interaction ( $S_j$ ):

$$S_r = \sum_{\substack{m=1 \\ m \neq r}}^n s(d_{rm}) \widehat{d_{rm}} \tag{6}$$

where  $d_{rm}$  is the distance between the  $r$ th and  $m$ th grasshopper, computed as  $d_{rm} = |x_m - x_r|$ ,  $\widehat{d_{rm}}$  is a unit vector from the  $r$ th to  $m$ th grasshopper, computed as  $\widehat{d_{rm}} = \frac{x_m - x_r}{d_{rm}}$ , and  $s$  is a function to describe the social forces' strength represented as follows:

$$s(r) = ia \times e^{\frac{-r}{lc}} - e^{-r} \tag{7}$$

where  $lc$  and  $ia$  are the intensity of attractive and attraction length scale, respectively. Based on the distance between two grasshoppers, they apply force on each other. This force could be absorption for far grasshoppers and repulsion for close grasshoppers. However, there is an exact value of distance, in which grasshoppers apply no force on each other, which called comfort zone. Moreover,  $Z_i$  and  $W_i$  are calculated as follows:

$$Z_i = -z\widehat{e}_z \tag{8}$$

$$W_i = u\widehat{e}_w \tag{9}$$

where  $z$  and  $\widehat{e}_z$  are the constant of gravity and a unity vector towards the earth's center, respectively, and  $u$  and  $\widehat{e}_w$  are the constant of gravity drift and a unity vector in the wind's direction, respectively. A modified version of the Equation (5) is represented as follows:

$$X_i^q = h \left( \sum_{\substack{j=1 \\ j \neq i}}^N h \frac{ub_q - lb_q}{2} s(|x_j^q - x_i^q|) \frac{x_j - x_i}{d_{ij}} \right) + \widehat{T}_q \tag{10}$$

where  $ub_q$  and  $lb_q$  are the upper bound and the lower bound in the  $q$ th dimension, respectively,  $\widehat{T}_q$  is the  $q$ th dimension of the target position,  $h$  is a decreasing coefficient to shrink the comfort zone. In the first iteration the rate of exploration is higher than that in the final iterations. Therefore,  $h$  should decrease as the algorithm get close to its end. The  $h$  parameter is calculated as follows:

$$h = h_{max} - IT \frac{h_{max} - h_{min}}{R} \tag{11}$$

where  $IT$  is the number of maximum iterations and  $h_{max}$  and  $h_{min}$  are 1 and 0.00001, respectively. Algorithm 3 represents the different steps of the GOA.

---

**Algorithm 3.** Pseudocode of GOA.

---

- 1: Initiate random grasshopper population considering the upper and lower bounds
  - 2: Calculate the fitness function for each grasshopper
  - 3: Target = best grasshoppers
  - 4:  $B$  = Maximum number of iterations
  - 5:  $b = 1$
  - 6: **while** ( $b < B$ )
  - 7:   Update  $h$  using Equation (11)
  - 8:    $N$  = the number of grasshoppers
  - 9:   **for** each grasshopper
  - 10:     Normalize the distance between grasshoppers in [1,4]
  - 11:     Update the position of the current grasshopper using Equation (10)
-

---

```

12:   Correct the grasshopper position considering boundaries
13: end
14: Update Target
15:  $b = b + 1$ 
16: end
17: Return Target

```

---

### 3.4. Artificial Neural Network (ANN)

ANN is an AI based method that simulates human brain to learn machines. ANN can solve new problems using past experiences like human brain. The most popular type of ANN is multi-layer perceptron (MLP). An MLP has an input layer (IL), which receives inputs from the patterns, a number of hidden layers (HL), which is responsible for processing, and an output layer (OL) that returns the output. IL and OL include a number of neurons equal to inputs and outputs, respectively. Figure 2 illustrates a schematic MLP.

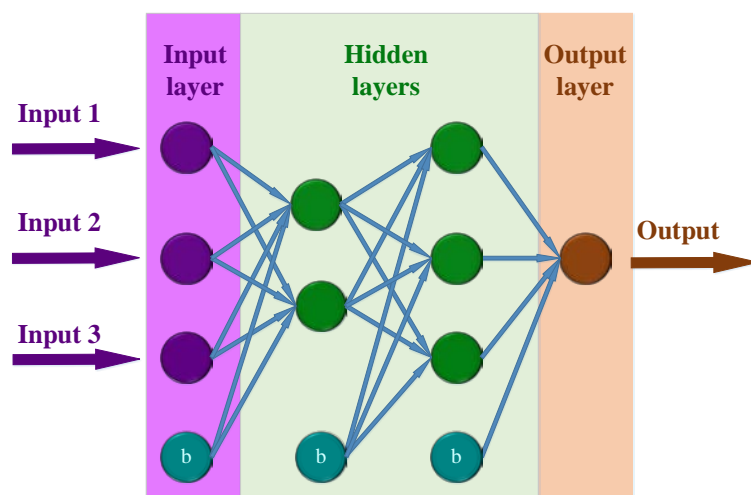


Figure 2. A schematic MLP.

Every node is linked to the nodes in the next layer with linking weights. The IL's nodes just receive the inputs and give them to the next layer. HLs' neurons take the outputs of the past layer's neurons, sum them, and put them in the activation function. The OL's nodes take their inputs from the previous layers' nodes, sum them, and return the answer as the network's outputs. There are various activation functions for ANNs such as sigmoid, tangent sigmoid, linear, and hyperbolic tangent sigmoid to name a few. In fact, the main problem of an ANN is to obtain the best weights for links where there are a number of algorithms for solving this problem. The most used problem-solving algorithm for ANNs is feed-forward back-propagation (FFBP) [82,83]. In FFBP algorithm, weights are predicted from the IL to OL, but improved from the OL to IL by learning algorithms. Among learning algorithms, Levenberg-Marquardt (LM), due to its better speed and performance, is the best one to predict concrete behavior [40,84]. In the LM algorithm, the updated weights and bias are calculated as follows:

$$wb_{t+1} = wb_t - [J^T] + \mu I]^{-1} J^T e \quad (12)$$

where  $wb_{t+1}$  and  $J$  are the updated weights and biases and the Jacobian matrix that includes the first derivatives of the network errors concerning the weights and biases.  $\mu$ ,  $I$ , and  $e$  are a positive real number damping factor, identity matrix, and vector of ANNs error, respectively. A schematic neuron is shown in Figure 3.



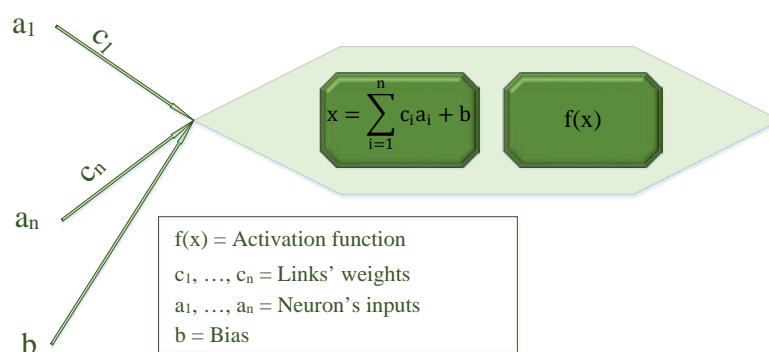


Figure 3. A schematic neuron.

The performance of an ANN is affected by its structure, and trial-and-error is the most commonly used method for obtaining the best structure. Using this method can cost a huge amount of time and energy, although there is a possibility of not gaining the best architecture. Hence, having a systematic method to reach the ANN’s best architecture is necessary. In the present study, an ANN is modified using GA, SSA, and GOA, separately. The gene of an individual and the position of a salp and grasshopper are divided into two parts. The first part is allocated to the number of HL. According to the Figure 4, the *i*th value of the first part can be zero or one. If it is zero, the *i*th HL is deactivated, and it is activated if the *i*th value of the first part is one. The second part, on the other hand, is related to the number of neurons in each HL. For instance, if the *j*th value of the second part is 11, it means there is 11 neurons in the *j*th HL.

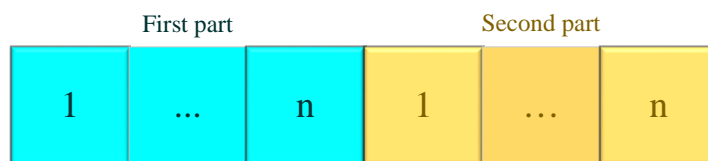


Figure 4. Position of a salp and a grasshopper and chromosome of an individual.

The process of each model is almost the same as its optimization algorithm, except that their fitness are calculated by ANN’s error, and ANN uses RMSE to compute the error.

$$RMSE = \sqrt{\frac{1}{Q} \sum_{j=1}^Q (M_q - V_q)^2} \tag{13}$$

where *Q* is number of records, *M<sub>q</sub>* is the Measured results and *V<sub>q</sub>* is predicted values related to the *q*th record.

### 3.5. M5P Tree

The M5P tree is the improved version of M5 tree that consist of three phases: dividing the input space, building the tree and deriving the knowledge [85]. The M5P tree, on the other hand, includes four Phases. The first phase includes building a full tree by splitting the data using the selected parameters and a division criterion. By using standard deviation reduction parameter (SDRP), the search space is split. SDRP is represented as follows:

$$SDRP = sd(D) - \sum_j \frac{|d_r|}{|D|} \times sd(d_r) \tag{14}$$

where || is the set’s number elements, *d<sub>r</sub>* is *r*th sub-space gained by splitting the node according to selected divided parameter, *sd* is the standard deviation, and *D* is the dataset

which reaches the node. This process is applied recursively until the patterns at a node either contains only a small number of sets or having negligible variations. The following phase includes developing a linear regression at each node. In the next phase, a pruning process is executed to catalyze the computing efficiency and avoid over-fitting. In the last phase, to deal with the sharp discontinuities that are results of data branching, the smoothing process is executed. The smoothing parameter is represented as follows:

$$A' = \frac{ra + sb}{r + s} \quad (15)$$

where  $A'$  is the calculated value passed up to the next higher node;  $a$ ,  $b$ ,  $r$ , and  $s$  are, respectively, the calculated target at this node, the model's estimated output at the current node, the cardinal number of training set that reach the node, and a smoothing parameter.

#### 4. Results

Before running the models, the data need to be prepared. In the first step, because of having different inputs with different scales, the dataset should be normalized in  $[-1,1]$  using following equation:

$$i_n = \frac{2(i - i_{min})}{(i_{max} - i_{min})} - 1 \quad (16)$$

where  $i$  is an input value,  $i_n$ ,  $i_{max}$ , and  $i_{min}$  are normal, maximum and minimum values of the  $i$ , respectively. Furthermore, every ANN requires a dataset to train the network, another data set to validate the training process, and a whole other data set to test the network. Usually the datasets are allocated to these phases by randomizing the patterns. The drawback of this method is overfitting. As a result, in the present research, k-fold cross-validation method is employed in which the data records are randomized and split into k fold [26,28,40,82,86–88]. Then, the model is run k times, and it uses a fold in each time for validating and testing. For instance, in the  $i$ th run, the  $i$ th fold is served as validation and testing dataset, and other folds are served for training. When the runs are finished, the average error of k times running the network is represented as the error of the network. Using this method results in that all records are served as training, validating, and testing.

Every model has its own adjustment parameters that need to be determined. The trial-and-error method is the most popular method for this purpose. Therefore, it is employed in the present research. The adjustment parameters of GA, SSA, and GOA are represented in Table 2.

**Table 2.** Adjustment parameters of modified ANNs.

Parameters	Values
Number of iterations	100
Size of populations	30
Hidden layer's maximum number of neurons	16
Maximum number of hidden layers	3
The number of folds	5
ANNs training algorithm	Levenberg-Marquardt
Activation function of output layer's neuron	Linear
Activation function of hidden layers' neurons	Hyperbolic tangent sigmoid

After running all models, three ANNs with different architecture, which are optimized by three different algorithms, are made that are illustrated in Figure 5. Moreover, M5P tree model is visualized in Figure 6. According to the Figure 5, hybridized ANN with GA (ANNGA) has a couple of the Hs and eleven neurons in each of them. Moreover, the hybridized ANN with SSA (ANNSSA) contains two Hs with seven neurons in the HL number one and nine neurons in the HL number two, and the hybridized ANN with GOA

(ANNGOA) has six neurons in its only HL. Hence, the ANNGOA has the simplest structure while the ANNGA has the biggest structure, and the ANNSSA is in between them. Biases and weights of the ANNs and the coefficient of the M5P tree is represented in the Appendix A.

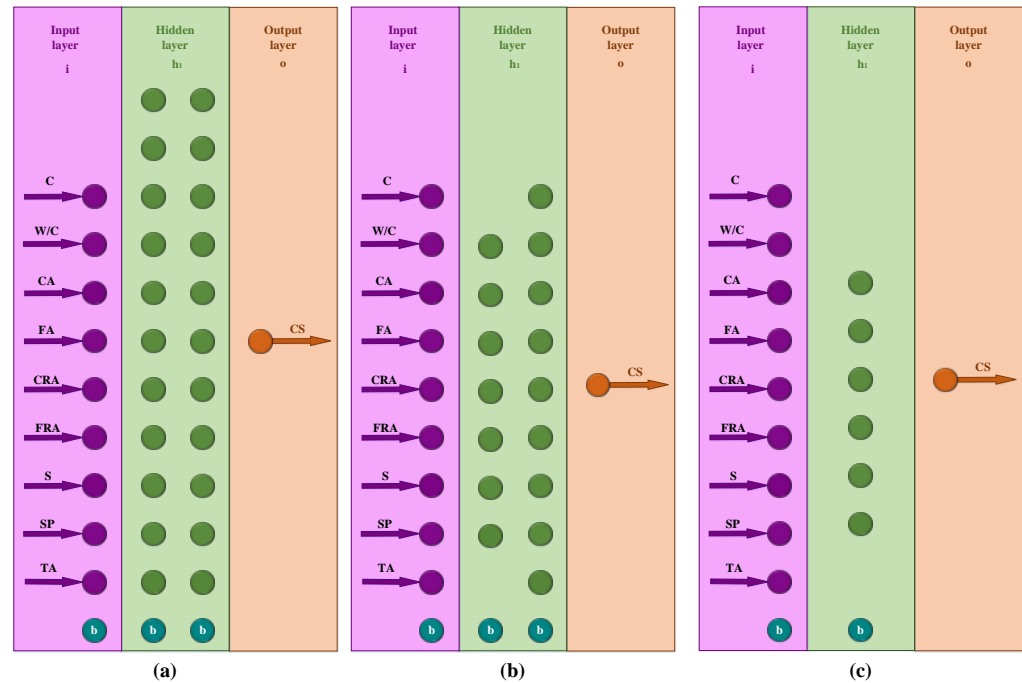


Figure 5. Architecture of (a) hybridized ANN with GA (ANNGA), (b) hybridized ANN with SSA (ANNSSA), and (c) hybridized ANN with GOA (ANNGOA).

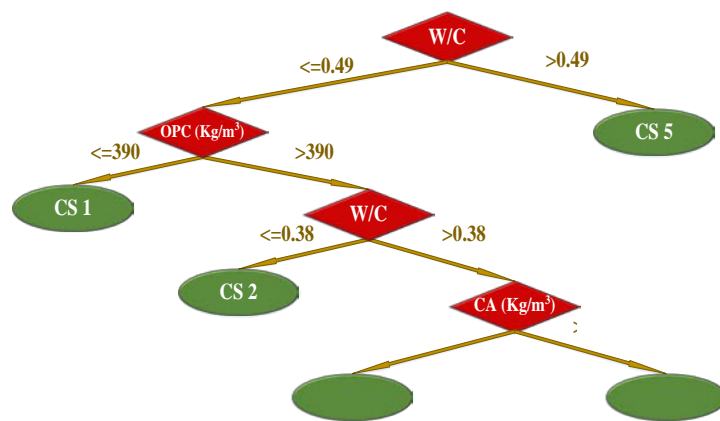


Figure 6. M5P tree model.

In the present study to have a good comparison among the four models, in addition to RMSE, Pearson correlation coefficient (R), mean absolute percentage error (MAPE), mean absolute error (MAE), mean absolute bias error (MBE), scatter index (SI), and relative square error (RSE) are employed, which are formulated as follow:

$$R = \frac{Q \sum_{i=1}^Q M_q V_q}{(Q \sum_{i=1}^Q M_q^2 - (\sum_{i=1}^Q M_q)^2)(Q \sum_{i=1}^Q V_q^2 - (\sum_{i=1}^Q V_q)^2)} \tag{17}$$

$$MAPE = \frac{100}{Q} \sum_{i=1}^Q \frac{V_q - M_q}{V_q} \tag{18}$$

$$MAE = \frac{1}{Q} \sum_{i=1}^Q |M_q - V_q| \quad (19)$$

$$MBE = \frac{1}{Q} \sum_{i=1}^Q (V_q - M_q) \quad (20)$$

$$SI = RMSE/\bar{M} \quad (21)$$

$$RSE = \frac{\sum_{i=1}^Q (V_q - M_q)^2}{\sum_{i=1}^Q (\bar{M} - M_q)^2} \quad (22)$$

where  $\bar{M}$  is mean value of measured results, and other parameters are explained in the previous section. MBE indicates that the model overestimates (MBE > 0) or underestimates (MBE < 0). SI determines that the performance of the model is “excellent” ( $0 \leq SI < 0.1$ ), “good” ( $0.1 \leq SI < 0.2$ ), “fair” ( $0.2 \leq SI < 0.3$ ), or “poor” ( $0.3 \leq SI$ ). Table 3 represents these indicators.

**Table 3.** Statistic indicator of models.

Models	RMSE (MPa)	MAE (MPa)	RSE	MAPE (%)	MBE (MPa)	SI	R
ANNSSA	2.734	1.890	0.017	6.216	0.156	0.075	0.992
ANNGA	2.944	2.086	0.019	6.797	−0.227	0.081	0.990
ANNGOA	3.027	2.227	0.020	7.790	0.037	0.083	0.990
M5P tree	7.098	5.502	0.149	21.458	0.179	0.196	0.947

According to Table 3, RMSE of ANNSSA is the lowest with the value of 2.734 MPa, while the M5P has the highest RMSE, which is over 2.5 times bigger than that of ANNSSA. The ANNGA and ANNGOA have the second and third least RMSE, respectively. In the case of MAE, the ANNSSA M5P tree has the highest cost by the value of 5.502 MPa followed by the ANNGOA of which MAE is lower than that of the M5P tree by almost 60%, the same value of ANNGA and ANNSSA are less than that of the M5P tree by 62% and 66%, respectively. According to RSE, the ANNSSA is the most accurate model with the RSE value of 0.017 while the M5P tree model is the least accurate one with the RSE value of 0.149, and ANNGA and ANNGOA are in between. The ANNSSA and ANNGA have almost the same MAPE while this value of the ANNGOA and the M5p tree are more than that of the ANNSSA by 25% and 245%, respectively. MBE indicates that the ANNSSA, the ANNGOA, and the M5P tree overestimate the compressive strength while the ANNGA underestimates that. According to SI, the M5P tree performs well, and other models performances are excellent. The R-values of all models are 0.99, which show a great correlation between measured and predicted compressive strength, except the R-value of M5P tree, which is 0.94. Figure 7 illustrates the predicted vs measured compressive strength, in which it can be seen that the scatter around the base line (linear regression) in ANNSSA is more than others. Moreover, Figure 8 compares MAE, RMSE, and MAPE in order to sort the models based on their efficiency. According to this figure, the ANNSSA is the most efficient model followed closely by the ANNGA and the ANNGOA respectively. Furthermore, the M5P tree is the least efficient model and has the highest error values. This order is true for MAE, RMSE, and MAPE.

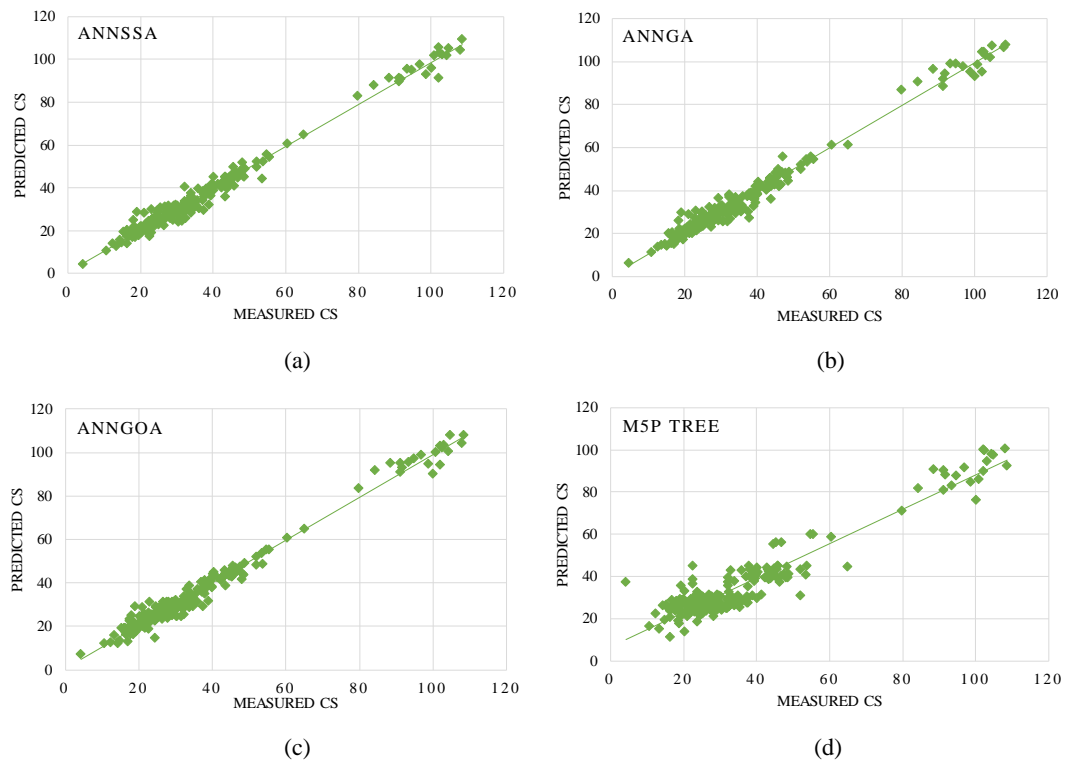
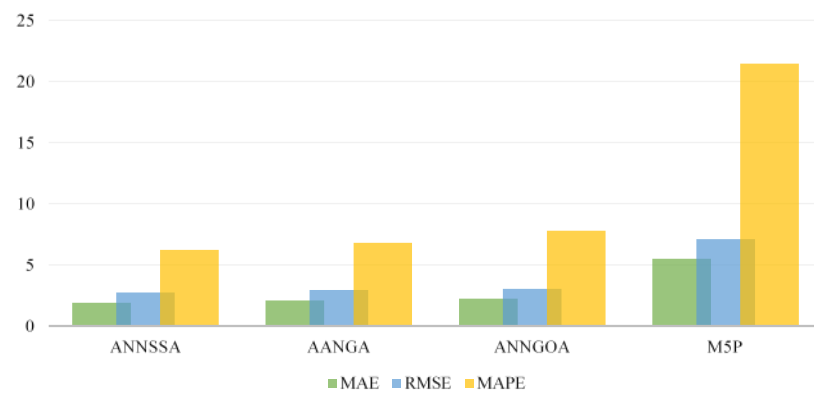
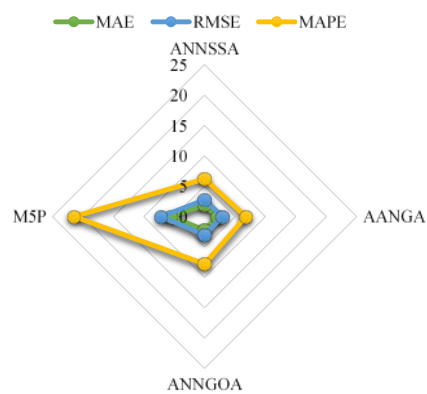


Figure 7. Predicted vs measured compressive strength for (a) ANNSSA, (b) ANNGA, (c) ANNGOA, and (d) M5P three.



(a)



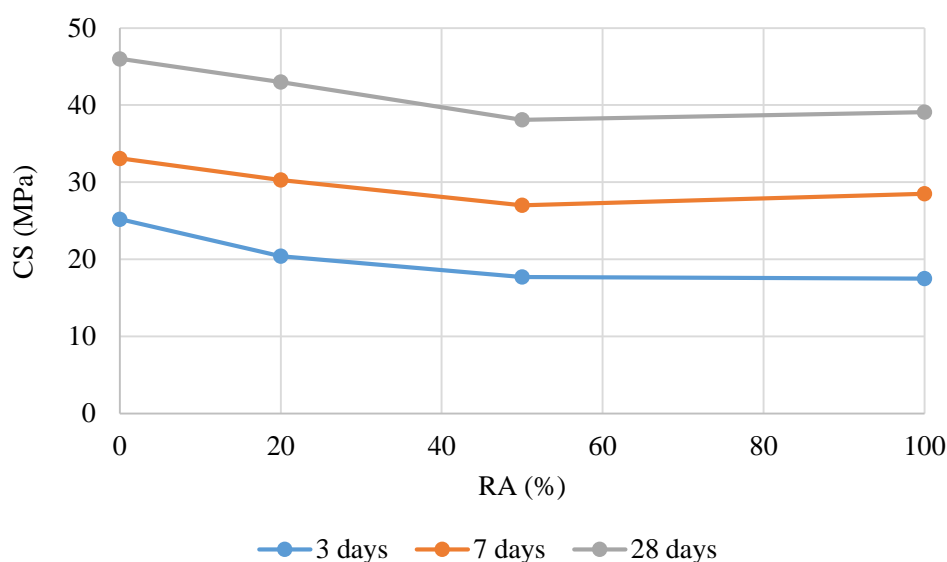
(b)

Figure 8. (a) A bar chart and (b) a radar chart to compare models' results overlay.

## 5. Conclusions

A simple ANN saves time and energy, which are spent on obtaining CS of concrete containing recycled aggregate. However, the ANN should be accurate as well to be useful. Otherwise, it just adds to higher costs. In this study, three different optimization algorithms (GA, SSA, and GOA) are employed to modify ANN for that purpose, and M5P tree served as a good comparison model. Results show that all ANNs have better performances than the M5P tree model. Moreover, the ANN that is hybridized with SSA has the lowest values of error, and the second simplest architecture. ANNGOA, on the other hand, has the simplest structure, with the highest values of error, and finally, ANNGA despite of its architecture, which is the most complex one, is the second most accurate model among the ANNs. Although all the ANNs are so close in both accuracy and complexity, the ANNSSA is recommended if accuracy is needed and the ANNGOA should be used if there is a need for simplicity.

The compressive strength (CS) of concrete containing different amounts of recycled aggregate (RA) is shown in Figure 9. According to this figure, increasing the amount of RA to 50% causes a decrease in the CS value at all ages. However, the CS value of the samples with 100% RA is higher than that of the samples with 50% RA at the age of 7 and 28 days, while the CS values of these two classes are almost equal at the age of 3 days.



**Figure 9.** The CS of concrete with different amounts of RA.

**Author Contributions:** Conceptualization, A.K. and M.K.; methodology, A.K.; software, A.K.; validation, A.K. and M.K.; formal analysis, A.K.; investigation, A.K.; data curation, F.S.; writing—original draft preparation, A.K. and F.S.; writing—review and editing, A.K. and M.K.; supervision, M.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Appendix A**

*Appendix A.1. Biases and Weights of the ANNGOA Model*

Weights of links between input and hidden layer

$$= \begin{bmatrix} 0.55 & 1.24 & 0.55 & -2.28 & -0.19 & 0.75 & 2.44 & -0.41 & 0.72 \\ 0.28 & 1.46 & 1.31 & -0.88 & 0.02 & 0.04 & -0.38 & -0.36 & 0.27 \\ -1.28 & -1.06 & -1.56 & -0.37 & -0.20 & 0.31 & 1.11 & 1.30 & 0.15 \\ -0.20 & -0.69 & -1.74 & -0.10 & -0.18 & 0.01 & -0.01 & 0.23 & 0.53 \\ 0.36 & 0.91 & 1.68 & -0.45 & 0.12 & -0.16 & -4.12 & 2.09 & -0.13 \\ -0.78 & 0.08 & 1.55 & 1.01 & 0.03 & 0.30 & 3.01 & 0.30 & -0.07 \end{bmatrix}$$

Biases of the input layer =  $[-0.52 \quad -1.32 \quad -0.03 \quad 0.26 \quad 0.14 \quad 0.43]$

Weights of links between hidden and output layer

=  $[0.50 \quad -1.13 \quad -2.18 \quad 1.08 \quad 1.31 \quad 1.54]$

Biases of the hidden layer =  $[-1.30]$

*Appendix A.2. Biases and Weights of the ANSSA Model*

Weights of links between input and the first hidden layer

$$= \begin{bmatrix} -0.88 & -2.09 & 0.52 & 1.13 & 0.34 & 0.28 & -1.36 & -1.13 & 0.63 \\ -0.29 & 0.28 & 2.25 & -0.64 & 2.00 & -1.59 & 0.42 & 0.03 & -0.10 \\ 0.14 & -0.13 & -1.22 & -0.21 & -0.23 & -0.07 & -0.27 & 0.18 & 0.28 \\ -1.79 & -1.60 & -1.36 & -0.27 & -0.11 & 0.91 & -2.98 & -1.35 & 0.27 \\ 0.68 & 0.73 & -0.60 & 2.16 & -1.08 & 1.13 & -0.19 & 0.16 & -1.07 \\ 1.48 & 0.68 & -1.70 & -1.82 & -0.22 & 0.06 & -1.12 & 0.57 & 0.44 \\ -1.42 & 0.73 & 0.78 & 3.01 & 1.20 & -0.58 & -0.61 & -0.07 & -1.38 \end{bmatrix}$$

Biases of the input layer =  $[0.39 \quad 0.10 \quad 0.19 \quad 0.38 \quad -0.61 \quad -0.16 \quad 0.25]$

Weights of links between the first and the last hidden layer

$$= \begin{bmatrix} 0.64 & 0.48 & -1.89 & 0.84 & -0.96 & -0.21 & 1.38 \\ -0.80 & -1.09 & -0.55 & 0.76 & 1.39 & 0.65 & 1.57 \\ -1.46 & -0.29 & 1.08 & -1.76 & -1.14 & -0.32 & 0.30 \\ -0.04 & 0.63 & -1.38 & -1.24 & -0.46 & -0.85 & -1.15 \\ 0.15 & -1.68 & -1.05 & -0.74 & 0.07 & 1.20 & 0.28 \\ 2.02 & 1.06 & -0.89 & -0.03 & -1.30 & -0.46 & -1.14 \\ 0.03 & 1.49 & 0.20 & 0.15 & 1.31 & 0.92 & -0.53 \\ 0.69 & -0.24 & 1.26 & -1.73 & -0.39 & -0.76 & 0.39 \\ 1.45 & -0.52 & -0.17 & -0.10 & -1.21 & -0.85 & 1.46 \end{bmatrix}$$

Biases of the first hidden layer

=  $[-1.59 \quad 0.90 \quad 0.96 \quad 0.31 \quad -0.19 \quad 1.10 \quad 0.84 \quad -1.04 \quad -0.08]$

Weights of links between the last hidden and output layer

=  $[-1.00 \quad 0.50 \quad -0.15 \quad 0.43 \quad -0.74 \quad 0.45 \quad 1.31 \quad 1.19 \quad 0.94]$

Biases of the last hidden layer =  $[-1.70]$

*Appendix A.3. Biases and Weights of the ANNGA Model*

Weights of links between input and the first hidden layer

$$= \begin{bmatrix} -0.67 & -1.27 & -0.76 & 0.42 & 0.12 & -0.20 & -0.27 & -0.29 & 0.62 \\ 1.11 & -0.30 & -1.36 & 0.06 & -0.21 & 0.48 & 0.93 & 0.40 & -0.15 \\ -1.11 & 0.11 & -0.80 & 0.50 & 0.82 & -0.19 & -0.45 & -1.36 & 0.89 \\ -0.71 & 1.12 & 1.35 & 0.18 & 0.28 & -0.44 & 0.34 & 0.168 & 0.45 \\ -1.45 & -0.71 & -0.95 & -0.37 & -0.14 & -0.45 & -0.46 & -0.29 & -0.45 \\ 0.68 & -1.47 & 1.36 & -1.05 & -0.01 & -0.20 & 0.04 & -0.27 & 0.98 \\ -0.45 & -0.71 & 1.04 & 0.48 & -0.28 & 0.65 & -0.01 & 0.37 & -1.04 \\ 1.18 & 1.40 & 0.03 & -1.73 & -0.15 & -0.50 & -1.52 & -0.10 & 0.31 \\ 0.27 & 1.04 & 0.42 & -0.48 & 0.42 & -1.58 & 1.07 & -0.24 & 0.21 \\ 1.07 & 0.35 & -0.83 & -0.50 & 0.20 & 0.48 & -0.19 & 0.22 & -0.97 \\ 0.77 & -0.49 & -0.61 & -0.74 & -0.477 & 0.14 & 1.23 & 0.68 & -0.39 \end{bmatrix}$$

Biases of the input layer

=  $[0.48 \quad -0.92 \quad 0.22 \quad 0.34 \quad -0.78 \quad 0.93 \quad -0.52 \quad -1.03 \quad -0.55 \quad -0.86 \quad 0.79]$

$$\begin{aligned}
 &\text{Weights of links between the first and the last hidden layer} \\
 &= \begin{bmatrix} 0.24 & 0.66 & 1.06 & -0.94 & 1.44 & -1.10 & 0.71 & -0.02 & -0.78 & -0.83 & -0.14 \\ 0.96 & 1.16 & -0.54 & -0.73 & 0.81 & 1.35 & -1.14 & 1.44 & -1.24 & 0.32 & -0.66 \\ -0.03 & -0.74 & -0.58 & -0.68 & -0.41 & -0.24 & -0.31 & -0.79 & -0.71 & -0.19 & -0.07 \\ -0.13 & -0.67 & 0.22 & 0.19 & 0.81 & 1.02 & 1.33 & -0.36 & -1.29 & 0.34 & -1.04 \\ -1.01 & 0.56 & 0.79 & -0.57 & -0.60 & -0.50 & -0.61 & 0.62 & -0.97 & -1.15 & -0.98 \\ -0.70 & 0.52 & -0.83 & -0.59 & -0.46 & 0.49 & -0.25 & 1.08 & -0.82 & 0.50 & -0.82 \\ -0.65 & -0.04 & 0.04 & 0.74 & -0.54 & -1.26 & -0.11 & 0.07 & 0.08 & 0.70 & -1.10 \\ -0.41 & -0.46 & -0.82 & -1.19 & -0.26 & 0.84 & 0.89 & 0.25 & -0.04 & -0.56 & 0.54 \\ -0.42 & 1.38 & -0.23 & 0.73 & 0.26 & -0.85 & -0.28 & 0.83 & 0.50 & 0.73 & -0.96 \\ -0.81 & -0.11 & -0.02 & 0.05 & 0.13 & 0.25 & 0.76 & 0.88 & -0.10 & -0.33 & -0.30 \\ 0.50 & 0.09 & 0.21 & -1.47 & -0.51 & -0.95 & -0.41 & -0.76 & -0.61 & -0.28 & -0.48 \end{bmatrix} \\
 &\text{Biases of the first hidden layer} \\
 &= [0.55 \quad 0.45 \quad -0.69 \quad -0.34 \quad -0.26 \quad -0.65 \quad 0.68 \quad 0.81 \quad -0.31 \quad 0.94 \quad -0.34] \\
 &\text{Weights of links between the last hidden and output layer} \\
 &= [-1.39 \quad -0.28 \quad 0.88 \quad -0.06 \quad 0.06 \quad -0.26 \quad -0.46 \quad -0.41 \quad 0.11 \quad -0.85 \quad -0.09] \\
 &\text{Biases of the last hidden layer} = [-0.45]
 \end{aligned}$$

Appendix A.4. Calculated Coefficient of the MP5 Tree Model

Table A1. Calculated coefficient for the M5P tree.

Linear Models	Coefficients									
	OPC	W/C	CA	FA	CRA	FRA	Sand	SP	TA	Bias
CS1	-0.3456	-227.63	0.0548	0.0277	12.8748	1.5747	0.0148	0	0.466	226.3128
CS2	-0.1483	-130.427	-0.0374	0.0277	-2.681	1.5747	0.0148	0	0.4618	180.8033
CS3	-0.2059	-130.427	-0.0393	0.0277	-3.4258	1.5747	0.0148	0	0.5897	204.2375
CS4	-0.2193	-130.427	-0.0489	0.0277	-3.1624	1.5747	0.0148	0	0.5241	220.4764
CS5	-0.0233	7.2736	0.0539	0.0764	-5.6897	7.9546	0.0363	0	0.659	60.8239

References

- Lotfy, A.; Al-Fayez, M. Performance evaluation of structural concrete using controlled quality coarse and fine recycled concrete aggregate. *Cem. Concr. Compos.* **2015**, *61*, 36–43, doi:10.1016/j.cemconcomp.2015.02.009.
- Wang, J.; Wu, H.; Tam, V.W.Y.; Zuo, J. Considering life-cycle environmental impacts and society’s willingness for optimizing construction and demolition waste management fee: An empirical study of China. *J. Clean. Prod.* **2019**, *206*, 1004–1014, doi:10.1016/j.jclepro.2018.09.170.
- Hammoudi, A.; Moussaceb, K.; Belebchouche, C.; Dahmoune, F. Comparison of artificial neural network (ANN) and response surface methodology (RSM) prediction in compressive strength of recycled concrete aggregates. *Constr. Build. Mater.* **2019**, *209*, 425–436, doi:10.1016/j.conbuildmat.2019.03.119.
- Figuroa, J.D.; Fout, T.; Plasynski, S.; McIlvried, H.; Srivastava, R.D. Advances in CO<sub>2</sub> capture technology-The U.S. Department of Energy’s Carbon Sequestration Program. *Int. J. Greenh. Gas Control.* **2008**, *2*, 9–20.
- Rochelle, G.T. Amine Scrubbing for CO<sub>2</sub> Capture. *Science* **2009**, *325*, 1652–1654.
- Wang, Y.; Zhao, L.; Otto, A.; Robinius, M.; Stolten, D. A Review of Post-combustion CO<sub>2</sub> Capture Technologies from Coal-fired Power Plants. *Energy Procedia* **2017**, *114*, 650–665, doi:10.1016/j.egypro.2017.03.1209.
- Todhunter, A.; Crowley, M.; Sartipi, F.; Jegendran, K. Use of the by-products of post-combustion carbon capture in concrete production: Australian case study. *J. Constr. Mater.* **2019**, *1*, 1–22, doi:10.36756/jcm.v1.1.1.
- Velay-Lizancos, M.; Martinez-Lage, I.; Azenha, M.; Granja, J.; Vazquez-Burgo, P. Concrete with fine and coarse recycled aggregates: E-modulus evolution, compressive strength and non-destructive testing at early ages. *Constr. Build. Mater.* **2018**, *193*, 323–331, doi:10.1016/j.conbuildmat.2018.10.209.
- Medina, C.; Zhu, W.; Howind, T.; Sánchez De Rojas, M.I.; Frías, M. Influence of mixed recycled aggregate on the physical-mechanical properties of recycled concrete. *J. Clean. Prod.* **2014**, *68*, 216–225, doi:10.1016/j.jclepro.2014.01.002.
- Behnood, A.; Olek, J.; Glinicki, M.A. Predicting modulus elasticity of recycled aggregate concrete using M5’ model tree algorithm. *Constr. Build. Mater.* **2015**, *94*, 137–147, doi:10.1016/j.conbuildmat.2015.06.055.
- Etxeberria, M.; Marí, A.R.; Vázquez, E. Recycled aggregate concrete as structural material. *Mater. Struct.* **2006**, *40*, 529–541, doi:10.1617/s11527-006-9161-5.
- Quan, H.Z. Study on Strength and Durability of Concrete Containing Recycled Coarse Aggregate Manufactured with Various Method. *Adv. Mater. Res.* **2011**, 1015–1018, doi:10.4028/www.scientific.net/amr.287-290.1015.
- Thomas, C.; Cimentada, A.; Polanco, J.; Setién, J.; Méndez, D.; Rico, J. Influence of recycled aggregates containing sulphur on properties of recycled aggregate mortar and concrete. *Compos. Part B Eng.* **2013**, *45*, 474–485, doi:10.1016/j.compositesb.2012.05.019.



14. Richardson, A.; Coventry, K.; Bacon, J. Freeze/thaw durability of concrete with recycled demolition aggregate compared to virgin aggregate concrete. *J. Clean. Prod.* **2011**, *19*, 272–277, doi:10.1016/j.jclepro.2010.09.014.
15. Sadowski, Ł.; Piechówka-Mielnik, M.; Widziszowski, T.; Gardynik, A.; Mackiewicz, S. Hybrid ultrasonic-neural prediction of the compressive strength of environmentally friendly concrete screeds with high volume of waste quartz mineral dust. *J. Clean. Prod.* **2019**, *212*, 727–740, doi:10.1016/j.jclepro.2018.12.059.
16. Han, T.; Siddique, A.; Khayat, K.H.; Huang, J.; Kumar, A. An ensemble machine learning approach for prediction and optimization of modulus of elasticity of recycled aggregate concrete. *Constr. Build. Mater.* **2020**, *244*, 118271, doi:10.1016/j.conbuildmat.2020.118271.
17. Chandwani, V.; Agrawal, V.; Nagar, R. Modeling slump of ready mix concrete using genetic algorithms assisted training of Artificial Neural Networks. *Expert Syst. Appl.* **2015**, *42*, 885–893, doi:10.1016/j.eswa.2014.08.048.
18. Delgado, J.M.P.Q.; Silva, F.; Azevedo, A.; Silva, D.; Campello, R.; Santos, R. Artificial neural networks to assess the useful life of reinforced concrete elements deteriorated by accelerated chloride tests. *J. Build. Eng.* **2020**, *31*, 101445, doi:10.1016/j.jobe.2020.101445.
19. Özcan, F.; Atis, C.D.; Karahan, O.; Uncuoğlu, E.; Tanyildizi, H. Comparison of artificial neural network and fuzzy logic models for prediction of long-term compressive strength of silica fume concrete. *Adv. Eng. Softw.* **2009**, *40*, 856–863, doi:10.1016/j.advengsoft.2009.01.005.
20. Liu, G.; Zheng, J. Prediction Model of Compressive Strength Development in Concrete Containing Four Kinds of Gelled Materials with the Artificial Intelligence Method. *Appl. Sci.* **2019**, *9*, 1039, doi:10.3390/app9061039.
21. Golafshani, E.M.; Pazouki, G. Predicting the compressive strength of self-compacting concrete containing fly ash using a hybrid artificial intelligence method. *Comput. Concr.* **2018**, *22*, 419–437.
22. Getahun, M.A.; Shitote, S.; Gariy, Z.C.A. Artificial neural network based modelling approach for strength prediction of concrete incorporating agricultural and construction wastes. *Constr. Build. Mater.* **2018**, *190*, 517–525, doi:10.1016/j.conbuildmat.2018.09.097.
23. Zhang, J.; Huang, Y.; Ma, G.; Sun, J.; Nener, B. A metaheuristic-optimized multi-output model for predicting multiple properties of pervious concrete. *Constr. Build. Mater.* **2020**, *249*, 118803, doi:10.1016/j.conbuildmat.2020.118803.
24. Farooq, F.; Amin, M.N.; Khan, K.; Sadiq, M.R.; Javed, M.F.; Aslam, F.; Alyousef, R. A Comparative Study of Random Forest and Genetic Engineering Programming for the Prediction of Compressive Strength of High Strength Concrete (HSC). *Appl. Sci.* **2020**, *10*, 7330, doi:10.3390/app10207330.
25. Golafshani, E.M.; Ashour, A. Prediction of self-compacting concrete elastic modulus using two symbolic regression techniques. *Autom. Constr.* **2016**, *64*, 7–19, doi:10.1016/j.autcon.2015.12.026.
26. Naderpour, H.; Rafiean, A.H.; Fakharian, P. Compressive strength prediction of environmentally friendly concrete using artificial neural networks. *J. Build. Eng.* **2018**, *16*, 213–219, doi:10.1016/j.jobe.2018.01.007.
27. Saridemir, M.; Topçu, I.B.; Özcan, F.; Severcan, M.H.; Saridemir, M. Prediction of long-term effects of GGBFS on compressive strength of concrete by artificial neural networks and fuzzy logic. *Constr. Build. Mater.* **2009**, *23*, 1279–1286, doi:10.1016/j.conbuildmat.2008.07.021.
28. Kioumarsis, M.; Azarhomayun, F.; Haji, M.; Shekarchi, M. Effect of Shrinkage Reducing Admixture on Drying Shrinkage of Concrete with Different w/c Ratios. *Materials* **2020**, *13*, 5721, doi.org/10.3390/ma13245721.
29. Ren, Q.; Li, M.; Zhang, M.; Shen, Y.; Si, W. Prediction of Ultimate Axial Capacity of Square Concrete-Filled Steel Tubular Short Columns Using a Hybrid Intelligent Algorithm. *Appl. Sci.* **2019**, *9*, 2802, doi:10.3390/app9142802.
30. Behnood, A.; Golafshani, E.M. Predicting the compressive strength of silica fume concrete using hybrid artificial neural network with multi-objective grey wolves. *J. Clean. Prod.* **2018**, *202*, 54–64, doi:10.1016/j.jclepro.2018.08.065.
31. Lv, Z.; Liu, C.; Zhu, C.; Bai, G.; Qi, H. Experimental Study on a Prediction Model of the Shrinkage and Creep of Recycled Aggregate Concrete. *Appl. Sci.* **2019**, *9*, 4322, doi:10.3390/app9204322.
32. Shahnewaz, M.; Alam, M.S. Genetic algorithm for predicting shear strength of steel fiber reinforced concrete beam with parameter identification and sensitivity analysis. *J. Build. Eng.* **2020**, *29*, 101205, doi:10.1016/j.jobe.2020.101205.
33. Behnood, A.; Behnood, V.; Gharehveran, M.M.; Alyamac, K.E. Prediction of the compressive strength of normal and high-performance concretes using M5P model tree algorithm. *Constr. Build. Mater.* **2017**, *142*, 199–207, doi:10.1016/j.conbuildmat.2017.03.061.
34. Atici, U. Prediction of the strength of mineral admixture concrete using multivariable regression analysis and an artificial neural network. *Expert Syst. Appl.* **2011**, *38*, 9609–9618, doi:10.1016/j.eswa.2011.01.156.
35. Kandiri, A.; Fotouhi, F. Prediction of the module of elasticity of green concretes containing ground granulated blast furnace slag using hybridized multi-objective ANN and Salp swarm algorithm. *J. Constr. Mater.* **2021**, *2*, 2, doi:10.36756/JCM.v2.2.2.
36. Uysal, M.; Tanyildizi, H. Predicting the core compressive strength of self-compacting concrete (SCC) mixtures with mineral additives using artificial neural network. *Constr. Build. Mater.* **2011**, *25*, 4105–4111, doi:10.1016/j.conbuildmat.2010.11.108.
37. Behnood, A.; Golafshani, E.M. Machine learning study of the mechanical properties of concretes containing waste foundry sand. *Constr. Build. Mater.* **2020**, *243*, 118152, doi:10.1016/j.conbuildmat.2020.118152.
38. Marai, M.A.; Ahmed, M.A.; Mousa, S.E.-B. Neural networks for predicting compressive strength of structural light weight concrete. *Constr. Build. Mater.* **2009**, *23*, 2214–2219, doi:10.1016/j.conbuildmat.2008.12.003.

39. Xu, J.; Chen, Y.; Xie, T.; Zhao, X.; Xiong, B.; Chen, Z. Prediction of triaxial behavior of recycled aggregate concrete using multi-variable regression and artificial neural network techniques. *Constr. Build. Mater.* **2019**, *226*, 534–554, doi:10.1016/j.conbuildmat.2019.07.155.
40. Kandiri, A.; Golafshani, E.M.; Behnood, A. Estimation of the compressive strength of concretes containing ground granulated blast furnace slag using hybridized multi-objective ANN and salp swarm algorithm. *Constr. Build. Mater.* **2020**, *248*, 118676, doi:10.1016/j.conbuildmat.2020.118676.
41. Golafshani, E.M.; Behnood, A. Estimating the optimal mix design of silica fume concrete using biogeography-based programming. *Cem. Concr. Compos.* **2019**, *96*, 95–105.
42. Topçu, I.B.; Saridemir, M. Prediction of compressive strength of concrete containing fly ash using artificial neural networks and fuzzy logic. *Comput. Mater. Sci.* **2008**, *41*, 305–311, doi:10.1016/j.commatsci.2007.04.009.
43. Kandiri, A.; Fotouhi, F. Prediction of the creep coefficient of green concretes containing ground granulated blast furnace slag using hybridized multi-objective ANN and Salp swarm algorithm. *J. Constr. Mater.* **2021**, *2*, 2, doi:10.36756/jcm.v2.2.1r.
44. Yuan, Z.; Wang, L.-N.; Ji, X. Prediction of concrete compressive strength: Research on hybrid models genetic based algorithms and ANFIS. *Adv. Eng. Softw.* **2014**, *67*, 156–163, doi:10.1016/j.advengsoft.2013.09.004.
45. Behnood, A.; Verian, K.P.; Gharehveran, M.M. Evaluation of the splitting tensile strength in plain and steel fiber-reinforced concrete based on the compressive strength. *Constr. Build. Mater.* **2015**, *98*, 519–529, doi:10.1016/j.conbuildmat.2015.08.124.
46. Li, Z.; Jin, Z.; Zhao, T.; Wang, P.; Zhao, L.; Xiong, C.; Kang, Y.; Kang, A.Y. Service Life Prediction of Reinforced Concrete in a Sea-Crossing Railway Bridge in Jiaozhou Bay: A Case Study. *Appl. Sci.* **2019**, *9*, 3570, doi:10.3390/app9173570.
47. Yan, F.; Lin, Z.; Wang, X.; Azarmi, F.; Sobolev, K. Evaluation and prediction of bond strength of GFRP-bar reinforced concrete using artificial neural network optimized with genetic algorithm. *Compos. Struct.* **2017**, *161*, 441–452, doi:10.1016/j.compstruct.2016.11.068.
48. Dantas, A.T.A.; Leite, M.B.; Nagahama, K.D.J. Prediction of compressive strength of concrete containing construction and demolition waste using artificial neural networks. *Constr. Build. Mater.* **2013**, *38*, 717–722, doi:10.1016/j.conbuildmat.2012.09.026.
49. Golafshani, E.M.; Behnood, A.; Arashpour, M. Predicting the compressive strength of normal and High-Performance Concretes using ANN and ANFIS hybridized with Grey Wolf Optimizer. *Constr. Build. Mater.* **2020**, *232*, 117266, doi:10.1016/j.conbuildmat.2019.117266.
50. Solhmirzaei, R.; Salehi, H.; Kodur, V.; Naser, M. Machine learning framework for predicting failure mode and shear capacity of ultra high performance concrete beams. *Eng. Struct.* **2020**, *224*, 111221, doi:10.1016/j.engstruct.2020.111221.
51. Xu, J.; Zhao, X.; Yu, Y.; Xie, T.; Yang, G.; Xue, J. 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. *Constr. Build. Mater.* **2019**, *211*, 479–491, doi:10.1016/j.conbuildmat.2019.03.234.
52. Ahmadi, M.; Kheyroddin, A.; Dalvand, A.; Kioumars, M. New empirical approach for determining nominal shear capacity of steel fiber reinforced concrete beams. *Constr. Build. Mater.* **2020**, *234*, 117293, doi:10.1016/j.conbuildmat.2019.117293.
53. Moradi, M.J.; Roshani, M.M.; Shabani, A.; Kioumars, M. Prediction of the Load-Bearing Behavior of SPSW with Rectangular Opening by RBF Network. *Appl. Sci.* **2020**, *10*, 1185, doi:10.3390/app10031185.
54. Ashrafi, A.; Yaseen, Z.M.; Masoumi, P.; Asadi-Shiadeh, M.; Yaghoobi-Chenari, M.; Mosavi, A.; Nabipour, N. Classification-Based Regression Models for Prediction of the Mechanical Properties of Roller-Compacted Concrete Pavement. *Appl. Sci.* **2020**, *10*, 3707, doi:10.3390/app10113707.
55. Bilim, C.; Atiş, C.D.; Tanyildizi, H.; Karahan, O. Predicting the compressive strength of ground granulated blast furnace slag concrete using artificial neural network. *Adv. Eng. Softw.* **2009**, *40*, 334–340, doi:10.1016/j.advengsoft.2008.05.005.
56. Tenza-Abril, A.; Villacampa, Y.; Solak, A.; Baeza-Brotons, F. Prediction and sensitivity analysis of compressive strength in segregated lightweight concrete based on artificial neural network using ultrasonic pulse velocity. *Constr. Build. Mater.* **2018**, *189*, 1173–1183, doi:10.1016/j.conbuildmat.2018.09.096.
57. Shahnewaz, M.; Rteil, A.; Alam, M.S. Shear strength of reinforced concrete deep beams—A review with improved model by genetic algorithm and reliability analysis. *Structures* **2020**, *23*, 494–508, doi:10.1016/j.istruc.2019.09.006.
58. Ramezani, M.; Kim, Y.H.; Sun, Z. Probabilistic model for flexural strength of carbon nanotube reinforced cement-based materials. *Compos. Struct.* **2020**, *253*, 112748, doi:10.1016/j.compstruct.2020.112748.
59. Yu, B.; Tang, R.-K.; Li, B. Probabilistic bond strength model for reinforcement bar in concrete. *Probab. Eng. Mech.* **2020**, *61*, 103079, doi:10.1016/j.probengmech.2020.103079.
60. Dubey, V.; Noshadravan, A. A probabilistic upscaling of microstructural randomness in modeling mesoscale elastic properties of concrete. *Comput. Struct.* **2020**, *237*, 106272, doi:10.1016/j.compstruc.2020.106272.
61. Lizarazo-Marriaga, J.M.; Higuera, C.; Guzmán, I.; Fonseca, L. Probabilistic modeling to predict fly-ash concrete corrosion initiation. *J. Build. Eng.* **2020**, *30*, 101296, doi:10.1016/j.job.2020.101296.
62. Kou, S.-C.; Poon, C.S.; Etcheberria, M. Influence of recycled aggregates on long term mechanical properties and pore size distribution of concrete. *Cem. Concr. Compos.* **2011**, *33*, 286–291, doi:10.1016/j.cemconcomp.2010.10.003.
63. Casuccio, M.; Torrijos, M.; Giaccio, G.; Zerbino, R. Failure mechanism of recycled aggregate concrete. *Constr. Build. Mater.* **2008**, *22*, 1500–1506, doi:10.1016/j.conbuildmat.2007.03.032.
64. Poon, C.-S.; Kou, S.C.; Lam, L. Influence of recycled aggregate on slump and bleeding of fresh concrete. *Mater. Struct.* **2006**, *40*, 981–988, doi:10.1617/s11527-006-9192-y.

65. Ridzuan, A.; Diah, A.; Hamir, R.; Kamarulzaman, K. The influence of Recycled Aggregate on the Early Compressive Strength and Drying Shrinkage of Concrete. In *Structural Engineering, Mechanics and Computation*; Elsevier: Amsterdam, The Netherlands, 2001; pp. 1415–1422.
66. Andreu, G.; Miren, E. Experimental analysis of properties of high performance recycled aggregate concrete. *Constr. Build. Mater.* **2014**, *52*, 227–235, doi:10.1016/j.conbuildmat.2013.11.054.
67. Poon, C.-S.; Shui, Z.; Lam, L.; Fok, H.; Kou, S. Influence of moisture states of natural and recycled aggregates on the slump and compressive strength of concrete. *Cem. Concr. Res.* **2004**, *34*, 31–36, doi:10.1016/s0008-8846(03)00186-8.
68. Ismail, S.; Ramli, M. Engineering properties of treated recycled concrete aggregate (RCA) for structural applications. *Constr. Build. Mater.* **2013**, *44*, 464–476, doi:10.1016/j.conbuildmat.2013.03.014.
69. Corinaldesi, V. Mechanical and elastic behaviour of concretes made of recycled-concrete coarse aggregates. *Constr. Build. Mater.* **2010**, *24*, 1616–1620, doi:10.1016/j.conbuildmat.2010.02.031.
70. Evangelista, L.; De Brito, J. Mechanical behaviour of concrete made with fine recycled concrete aggregates. *Cem. Concr. Compos.* **2007**, *29*, 397–401, doi:10.1016/j.cemconcomp.2006.12.004.
71. Adnan, S.; Rahman, I.A.; Saman, H.M. Recycled aggregate as coarse aggregate replacement in concrete mixes. *HBRC* **2011**, *9*, 193–200.
72. Kou, S.-C.; Poon, C.-S.; Wan, H.-W. Properties of concrete prepared with low-grade recycled aggregates. *Constr. Build. Mater.* **2012**, *36*, 881–889, doi:10.1016/j.conbuildmat.2012.06.060.
73. Xiao, J.; Li, J.; Zhang, C. Mechanical properties of recycled aggregate concrete under uniaxial loading. *Cem. Concr. Res.* **2005**, *35*, 1187–1194, doi:10.1016/j.cemconres.2004.09.020.
74. Rahal, K. Mechanical properties of concrete with recycled coarse aggregate. *Build. Environ.* **2007**, *42*, 407–415, doi:10.1016/j.buildenv.2005.07.033.
75. Holland, J. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intel-Ligence*; MIT Press: Cambridge, MA, USA, 1992.
76. Sahoo, S.; Mahapatra, T.R. ANN Modeling to study strength loss of Fly Ash Concrete against Long term Sulphate Attack. *Mater. Today Proc.* **2018**, *5*, 24595–24604.
77. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191, doi:10.1016/j.advengsoft.2017.07.002.
78. Kang, F.; Li, J.; Dai, J. Prediction of long-term temperature effect in structural health monitoring of concrete dams using support vector machines with Jaya optimizer and salp swarm algorithms. *Adv. Eng. Softw.* **2019**, *131*, 60–76, doi:10.1016/j.advengsoft.2019.03.003.
79. Rogers, S.M.; Matheson, T.; Despland, E.; Dodgson, T.; Burrows, M.; Simpson, S.J. Mechanosensory-induced behavioural gregarization in the desert locust *Schistocerca gregaria*. *J. Exp. Biol.* **2003**, *206*, 3991–4002, doi:10.1242/jeb.00648.
80. Simpson, S.J.; McCaffery, A.R.; Hägele, B.F. A behavioural analysis of phase change in the desert locust. *Biol. Rev.* **2007**, *74*, 461–480, doi:10.1111/j.1469-185x.1999.tb00038.x.
81. Nazari, A.; Sanjayan, J.G. Modelling of compressive strength of geopolymer paste, mortar and concrete by optimized support vector machine. *Ceram. Int.* **2015**, *41*, 12164–12177, doi:10.1016/j.ceramint.2015.06.037.
82. Coello, C.A.C. Evolutionary multi-objective optimization: Some current research trends and topics that remain to be explored. *Front. Comput. Sci. China* **2009**, *3*, 18–30, doi:10.1007/s11704-009-0005-7.
83. Golafshani, E.M.; Behnood, A. Application of soft computing methods for predicting the elastic modulus of recycled aggregate concrete. *J. Clean. Prod.* **2018**, *176*, 1163–1176, doi:10.1016/j.jclepro.2017.11.186.
84. Quinlan, J.R. Learning with continuous classes. In Proceedings of the Australian Joint Conference on Artificial Intelligence, Hobart, TAS, Australia, 16–18 November 1992; pp. 343–348.
85. Hamdia, K.M.; Lahmer, T.; Nguyen-Thoi, T.; Rabczuk, T. Predicting the fracture toughness of PNCs: A stochastic approach based on ANN and ANFIS. *Comput. Mater. Sci.* **2015**, *102*, 304–313, doi:10.1016/j.commatsci.2015.02.045.
86. Chou, J.-S.; Chong, W.K.; Bui, D.-K. Nature-Inspired Metaheuristic Regression System: Programming and Implementation for Civil Engineering Applications. *J. Comput. Civ. Eng.* **2016**, *30*, 04016007, doi:10.1061/(asce)cp.1943-5487.0000561.
87. Chou, J.-S.; Pham, A.-D. Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength. *Constr. Build. Mater.* **2013**, *49*, 554–563, doi:10.1016/j.conbuildmat.2013.08.078.
88. Bui, D.-K.; Nguyen, T.; Chou, J.-S.; Nguyen-Xuan, H.; Ngo, T.D. A modified firefly algorithm-artificial neural network expert system for predicting compressive and tensile strength of high-performance concrete. *Constr. Build. Mater.* **2018**, *180*, 320–333, doi:10.1016/j.conbuildmat.2018.05.201.