applied sciences

MDPI

Article

Predicting Compressive Strength of 3D Printed Mortar in Structural Members Using Machine Learning

Hamed Izadgoshasb [1], Amirreza Kandiri [2], Pshtiwan Shakor [3,*], Vittoria Laghi [4] and Giada Gasparini [4]

[1] DITEN Department, University of Genoa, 16145 Genoa, Italy; hamedizadgoshasb@gmail.com
[2] School of Civil Engineering, University College Dublin, Belfield, D04 V1W8 Dublin, Ireland; amirreza.kandiri@ucdconnect.ie
[3] School of Civil and Environmental Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia
[4] Department of Civil, Chemical, Environmental and Materials Engineering, University of Bologna, 40136 Bologna, Italy; vittoria.laghi2@unibo.it (V.L.); giada.gasparini4@unibo.it (G.G.)
* Correspondence: pshtiwan.shakor@uts.edu.au or pshtiwan.shakor@alumni.uts.edu.au

Abstract: Machine learning is the discipline of learning commands in the computer machine to predict and expect the results of real application and is currently the most promising simulation in artificial intelligence. This paper aims at using different algorithms to calculate and predict the compressive strength of extrusion 3DP concrete (cement mortar). The investigation is carried out using multi-objective grasshopper optimization algorithm (MOGOA) and artificial neural network (ANN). Given that the accuracy of a machine learning method depends on the number of data records, and for concrete 3D printing, this number is limited to few years of study, this work develops a new method by combining both methodologies into an ANNMOGOA approach to predict the compressive strength of 3D-printed concrete. Some promising results in the iteration process are achieved.

Keywords: multi-objective optimization; artificial neural network; compressive strength; 3DP mortar; additive manufacturing

check for updates

Citation: Izadgoshasb, H.; Kandiri, A.; Shakor, P.; Laghi, V.; Gasparini, G. Predicting Compressive Strength of 3D Printed Mortar in Structural Members Using Machine Learning. *Appl. Sci.* **2021**, *11*, 10826. https://doi.org/10.3390/app112210826

Academic Editor: Jorge de Brito

Received: 28 September 2021
Accepted: 11 November 2021
Published: 16 November 2021

1. Introduction

Extrusion 3DP is known as one of the most promising techniques for the con-struction industry. Therefore, it is also referred to as "number one 3DP technology in Additive Manufacturing (AM) for construction" [1,2]. AM is a digital manufacturing technique to produce 3D printed parts and works based on a layer by layer printing which is supported by CAD files. This technique could help to construct complicated geometries which are hard to fabricate by formwork. In recent years, AM techniques endeavour to produce great structural members which can be suitable to build on Earth and beyond planet Earth.

Although there are many studies on predicting concrete mechanical strength for conventional concrete casting, there is not any convenient study or only a few on ma-chine learning (ML) for 3D printing concrete. For example, Dutta, et al. [3] used 1200 dataset samples for the evaluation of the concrete compressive strength of mix designs with coarse aggregates smaller than 20 mm. In their study, they have developed three models and they have selected the best suited model which can be applicable to use in predicting the compressive strength. According to their investigation, the cement con-tent in the concrete matrix has a major influence on the concrete compressive strength.

ML is another tool of digital transformation in the 21st century ML allows the computer to learn automatically from the entry data and make the decision and pre-dictions without having been specifically programmed [4].

Having ML technology with a large number of collected data would be beneficial to use as an Artificial Intelligence (AI) to prove the quality of production and predicting mechanical strength. [5]. It is vital to address that the data alone cannot provide enough information without having it analysed and interpreted correctly [6].

According to an earlier study by Obermeyer and Emanuel [6], AI includes three main AM applications, namely Powder-Bed Fusion, Directed Energy Deposition and Material Extrusion. However, it can be stated that all types of AM could be part of AI as long as sufficient information and data on them is available [7].

According to the study of Meng, et al. [8], all types of neural networks refer to ar-tificial neural networks (ANN). Usually, a neural network (NN) comprises an input layer which is usually one or more hidden layers and one or more output layers. The layer is created of numerous neurons and the data of each neuron is spread to the next layer based on different weights. In ANN, categorized under regular NN, the spreading neurons form as a cycle and are fed through NN. Moreover, while it is under training condition, each neuron's weight would be optimized by the learning instruction and new information is imported into the NN. The most popular learning rule for NN is the Back-Propagation (BP) algorithm [9,10] which controls weight based on the gradient descent.

Zhang, et al. [11] used NN in the AM process to predict the tensile strength for the printed part, the RMSE value recorded only 2%. Figure 1 describes the NN which was performed made during training when the output of each input is spreading backward in the NN to adjust the relevance of each input feature. After training, a new combina-tion of input features is spreading forward to guess the tensile strength value. Generally, NN displayed excellent performance in regression tasks, but it still requires to have some essential tuning of some hyperparameters, such as the number of hidden neurons and layers. Therefore, in this paper, a multi-objective optimization algorithm is used to find the optimal structure of the ANN for predicting compressive strength of 3DP concrete.

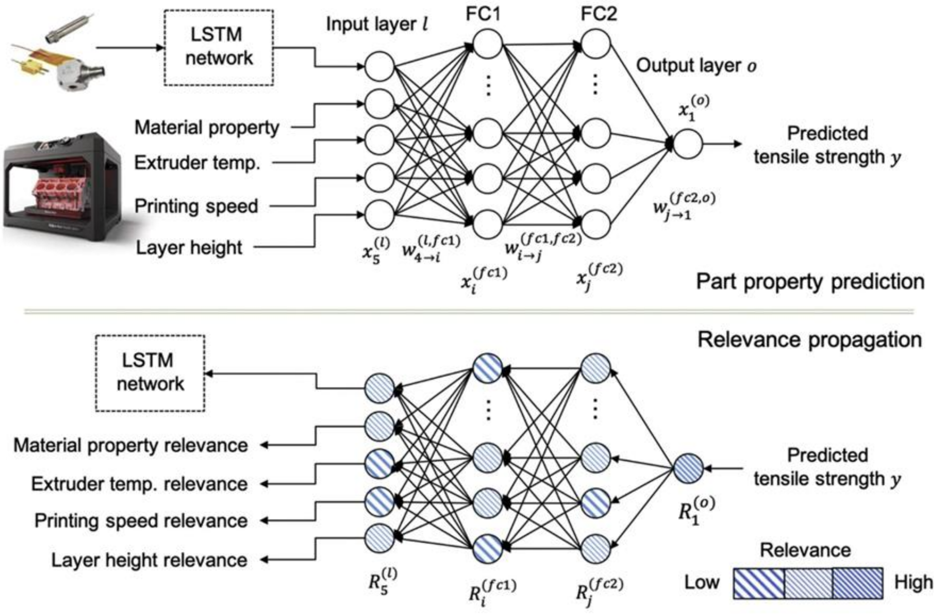


Figure 1. Propagation trained neural network for the filament polymer in fused filament fabrica-tion (FFF) 3DP: propagation forward to predict tensile strength (top) and propagation backward training to verify each input feature (bottom) reproduced with permission from Zhang et al., 2019 (Computers in Industry) published by Elsevier, 2019 [11].

2. Research Significance

The present study investigates and interprets the use of ML to be suitable for the AM-based process in 3D printing concrete/mortar applications. The study has collected data from 26 studies that have mentioned a mixed proportion of cementitious materials properly. In particular, the paper discusses the best-suited algorithm method for pre-dicting the mechanical strength of 3DP concrete, among which the most suited method in ML is selected. Then, literature studies are gathered in the mix design and are in-tensely used as input to predict the mechanical strength of concrete.

3. Methodology

The research's methodological approach is discussed in this section. The methodology of ANN is presented after introducing the GOA and MOGOA. Afterwards, a model is proposed which is a mix of ANN and MOGOA.

3.1. Grasshopper Optimization Algorithm

Grasshoppers are a species of insects that live in large hordes, despite being ob-served individually [12]. Their swarming activity begins while they are nymphs and continues throughout maturity [13]. Nymph grasshoppers migrate by hopping and leaping in a cylinder-like pattern [14]. Their colony travels slowly and in little incre-ments throughout this period. Therefore, the grasshopper's algorithm is based on the nature of the grasshopper's life exploration. This algorithm, like some other evolution algorithms, begins with inquiry and ends with exploitation. Whilst agents seek in the distance during the discovery phase, they explore the nearest region during the ex-ploitation phase. The following is the Equation (1) that models of grasshopper swarming attitudes:

$$\mathrm{P_i = x_2 Fg_i + x_1 S_i + x_3 Aw_i}, \tag{1}$$

where $\mathrm{Aw_i}$ and $\mathrm{Fg_i}$ represent wind advection and force of gravity on the grasshopper in position i, $\mathrm{S_i}$ indicates social contact as indicated in Equation (2), $\mathrm{P_i}$ represents the ith search agent's position, and $\mathrm{x_1}$, $\mathrm{x_2}$, and $\mathrm{x_3}$ are values that are chosen randomly in [0, 1] to provide random behaviour.

$$\mathrm{S_i} = \sum_{\substack{\mathrm{k=1} \\ \mathrm{k} \neq i}}^{\mathrm{N}} \mathrm{s\,(d_{ik})\,\hat{d_{ik}}} \tag{2}$$

Here, the distance between both the ith and jth grasshopper is $\mathrm{d_{ik}}$, which is calculated by Equation (3); $\widehat{\mathrm{d_{ik}}}$ is just the unit vector again from ith towards jth grasshopper, as obtained by Equation (4), where s is indeed a function to characterize the intensity of the social forces, as calculated by Equation (5).

$$\mathrm{d_{ij}} = \left|\mathrm{p_j - p_i}\right|; \tag{3}$$

$$\widehat{\mathrm{d_{ij}}} = \frac{\mathrm{p_j - p_i}}{\mathrm{d_{ij}}}; \tag{4}$$

$$\mathrm{s(l) = be^{\frac{-l}{m}} - e^{-l}}, \tag{5}$$

where m stands for the appealing length scale and b indicates attractive intensity. With Equation (5) for f = 0.5 and l = 1.5 is illustrated in Figure 2, where the force of social interaction is equal to zero for a distance of 2.079 units, and negative and positive for distances below and above 2.079 units, respectively. To put it another way, every grasshopper attracts grasshoppers that are further away than 2.079 units and repulses those that are closer. The safety zone is defined as 2.079 units in which neither soaking up nor disgust occurs. This function is approximately zero for distances greater than 4 units. As illustrated in Figure 2, implying that grasshoppers cannot impact those that are further away. Equations (6) and (7) are used to determine the $\mathrm{Fg_i}$ and $\mathrm{Aw_i}$.

$$\mathrm{Fg_i} = -\mathrm{g}\widehat{\mathrm{h_g}}; \tag{6}$$

$$\mathrm{Aw_i} = \mathrm{u}\widehat{\mathrm{h_w}}, \tag{7}$$

where $\widehat{\mathrm{h_w}}$ and $\widehat{\mathrm{h_g}}$ indicate unity vectors in the direction of wind toward the centre of the earth, respectively, and g is the gravitational constants and u is the drift's constant.

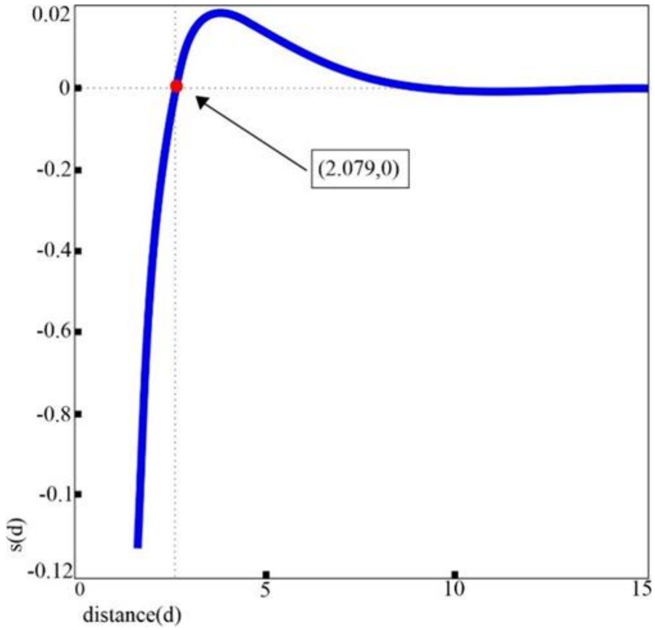


Figure 2. Function's in Equation (6).

The formula in Equation (1) cannot be proposed for solving optimization issues since the search agents reach the comfort zone quickly and the swarm is not able to converge to a specific point. As a result, that model may be updated to:

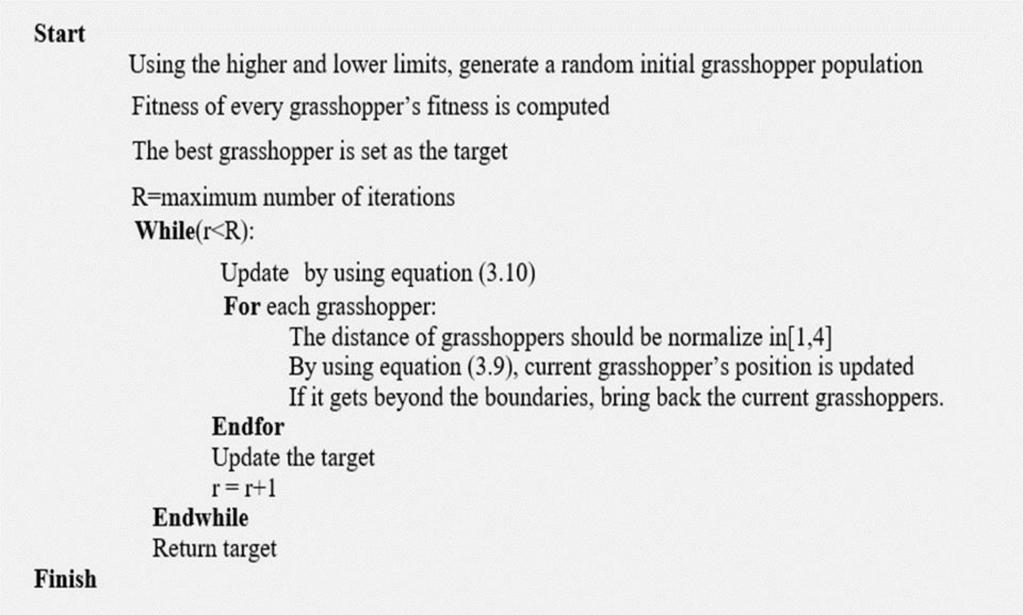
$$\mathrm{P_i^n} = \mathrm{m}\left(\sum_{\substack{k=1 \\ k \neq i}}^{\mathrm{N}} \mathrm{r_c}\frac{\mathrm{ub_n} - \mathrm{lb_n}}{2}\mathrm{s}\left(\left|\mathrm{p_j^n} - \mathrm{p_i^n}\right|\right)\frac{\mathrm{p_j} - \mathrm{p_i}}{d_{\mathrm{ik}}}\right) + \widehat{\mathrm{D_n}} \tag{8}$$

Here, $\mathrm{r_c}$ is a reduction coefficient to reduce the comfort zones, $\widehat{\mathrm{D_n}}$ is the target's nth dimension value, and $\mathrm{ub_n}$ is the nth dimension's upper bound, and $\mathrm{lb_n}$ is the nth dimension's lower bound.

To equalize the exploration and exploitation phases, the $\mathrm{r_c}$ a variable must be decreased as the number of steps increases. To put it another way, the rate of investigation in the early phases must be higher than that of the later ones. As a result, $\mathrm{r_c}$ may be calculated as follows:

$$\mathrm{r_c} = \mathrm{r_{cmax}} - \mathrm{r}\frac{\mathrm{r_{cmax}} - \mathrm{r_{cmin}}}{\mathrm{R}} \tag{9}$$

$\mathrm{r_{cmax}}$ and $\mathrm{r_{cmin}}$ are 1 and 0.00001, respectively, while R is the number of greatest iterations, and r is the number of the current iteration, respectively. The GOA's specific processes are depicted in Figure 3.



```
Start
        Using the higher and lower limits, generate a random initial grasshopper population
        Fitness of every grasshopper's fitness is computed
        The best grasshopper is set as the target
        R=maximum number of iterations
        While(r<R):
                Update  by using equation (3.10)
                For each grasshopper:
                        The distance of grasshoppers should be normalize in[1,4]
                        By using equation (3.9), current grasshopper's position is updated
                        If it gets beyond the boundaries, bring back the current grasshoppers.
                Endfor
                Update the target
                r = r+1
        Endwhile
        Return target
Finish
```

Figure 3. Pseudo code of GOA.

3.2. Multi-Objective Grasshopper Optimization Algorithm (MOGOA)

Single-objective and multi-objective optimization issues are two types of optimi-zation problems. Finding the optimum solution amongst these search agents is simple in single-objective algorithms. However, finding the optimum results in multi-objective algorithms (MOA) is quite difficult. As a result, MOGOA employs the dominance strategy. The two following agents A and B were required to apply. Agent A, according to this theory, has the upper hand against agent B if the following criteria are met:

$$\forall i \in (1, 2, \ldots, k) : f_i(\overrightarrow{Q}) \leq f_i(\overrightarrow{Y}) \ \& \ \exists i \in (1, 2, \ldots, k) : f_i(\overrightarrow{Q}) < f_i(\overrightarrow{Y}) \tag{10}$$

Here, Q and Y are the problem's solution vectors. All non-dominated solutions (NDS) are collected in a repository termed Pareto optimal using a multi-objective op-timization method. For example, in Figure 4, the black solutions (dots) are known as non-dominated, whereas the green solutions (dots) are known as a dominated (DS). The Pareto front is a collection of NDSs.

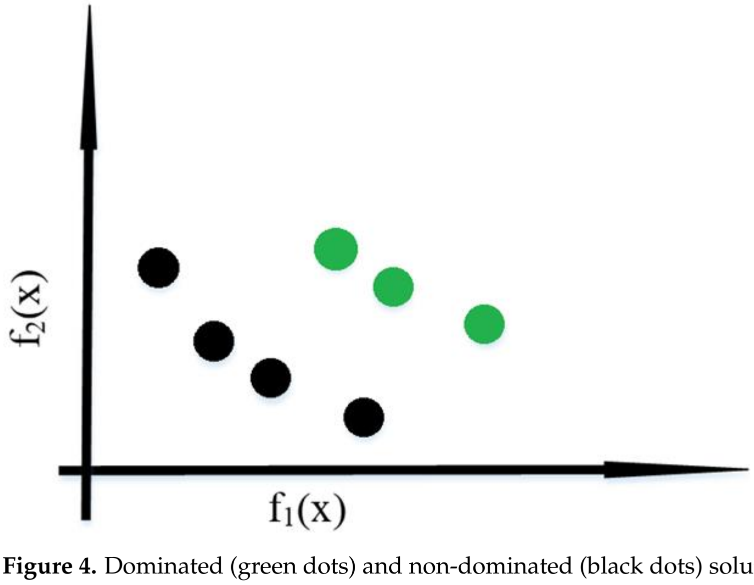
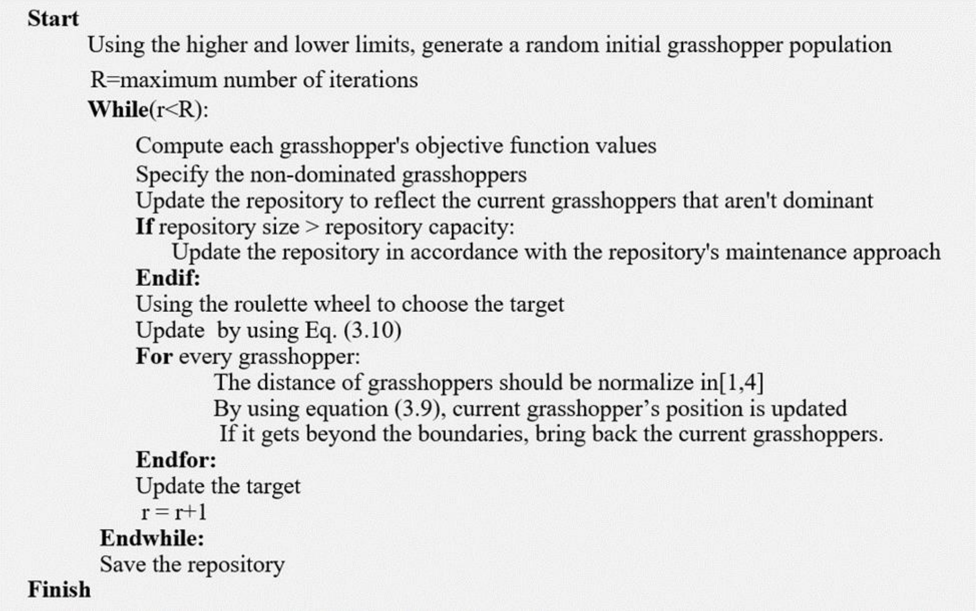


Figure 4. Dominated (green dots) and non-dominated (black dots) solutions.

Finding the target in each cycle of an MOA counted as another difficulty. Sin-gle-optimization algorithms pick the best solution as its goal with ease, while MOAs have no best solution and instead have a group of NDSs. First, the region is split into several identical neighbourhoods to solve this problem. The chance of selecting the target from each neighbourhood is then calculated as follows:

$$\mathrm{Z_i} = \frac{1}{\mathrm{C_i}}, \tag{11}$$

where $\mathrm{Z_i}$ denotes the likelihood of selecting the ith neighbourhood, and $\mathrm{C_i}$ represents the variety of solutions in that neighbourhood. During the last phase, a roulette wheel is used to select a target based on its probability. The repository's capacity must be restricted; else, the computational cost would rise. Every incoming NDS will also be checked to the solution inside the old archive at each cycle. There are three possibilities in this case. If at least one archival solution dominates a new NDS, it is discarded. They are swapped whenever a fresh NDS takes over a stored solution. Eventually, a new NDS should be uploaded to the archive if it neither dominates nor is dominated by any existing stored solution. Introducing a new solution to the repository, on the other hand, might be an issue if the repository was already full. In this scenario, a procedure similar to the one used to choose the target with an opposing scoring system is employed to select a solution. Figure 5 represents the different MOGO steps.



```
Start
    Using the higher and lower limits, generate a random initial grasshopper population
    R=maximum number of iterations
    While(r<R):
        Compute each grasshopper's objective function values
        Specify the non-dominated grasshoppers
        Update the repository to reflect the current grasshoppers that aren't dominant
        If repository size > repository capacity:
            Update the repository in accordance with the repository's maintenance approach
        Endif:
        Using the roulette wheel to choose the target
        Update  by using Eq. (3.10)
        For every grasshopper:
            The distance of grasshoppers should be normalize in[1,4]
            By using equation (3.9), current grasshopper's position is updated
            If it gets beyond the boundaries, bring back the current grasshoppers.
        Endfor:
        Update the target
        r = r+1
    Endwhile:
    Save the repository
Finish
```

Figure 5. Pseudo code of the MOGOA.

3.3. *Artificial Neural Network (ANN)*

Throughout the 1940s, an artificial neural network (ANN) model based on the human brain and neural system was created to learn machines. In comparison to tradi-tional computers, the human brain is significantly better at solving new problems and processing information, while being slower [15]. ANNs are developed to do tasks like those performed by the human brain, utilizing data gathered from previous experiences. To address issues, ANNs use a variety of methods, the most prominent of which being feed-forward back-propagation (FFBP) [16]. FFBPs are the most often utilized ANNs in prior studies [17] due to their ease of application and ability to forecast.

FFPBs are made up of several layers, each with a distinct number of neurons. The first layer among these networks is the "input layer," which receives inputs from the database and provides output values, while the last layer is the "output layer," which produces output values. Neurons have the same number as the number of input and output values. In addition, between such two levels, there may be one or more hidden layers that do the processing. Every neuron in the following layer is connected to all the other neurons in the layer below via weighted connections one by one. Neurons and hidden layers' number determines the ANN's accuracy; nevertheless, as the network's complexity increases, thus the accuracy increases [15,16,18]. Figure 6 illustrates a mul-ti-layer ANN with a hidden layer.

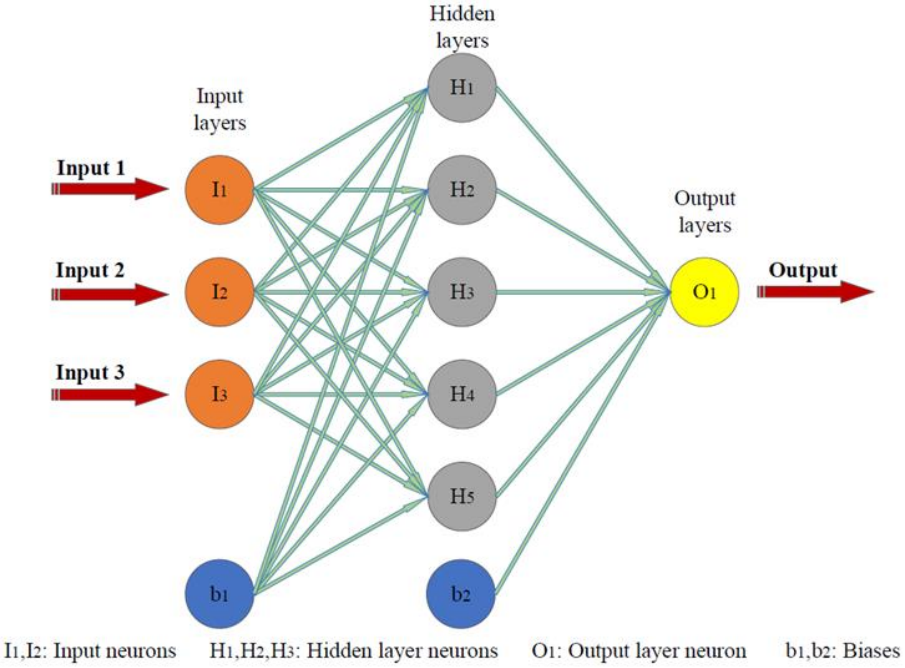


Figure 6. General schematic of multi-layer ANN.

Each neuron in the hidden layers gets several weighted inputs and a bias collects them and applies an activation function to produce a final value that is an input in the next layer. The input layer's neurons have no purpose other than to receive data from the dataset, while the neurons of the output layer return their output (network output). Typically, linear activation functions are employed in the output layer for neurons, while hyperbolic tangent sigmoid activation functions are used in the hidden lay-ers(Kandiri, Mohammadi Golafshani, and Behnood 2020). An ANN's schematic neuron is shown in Figure 7.

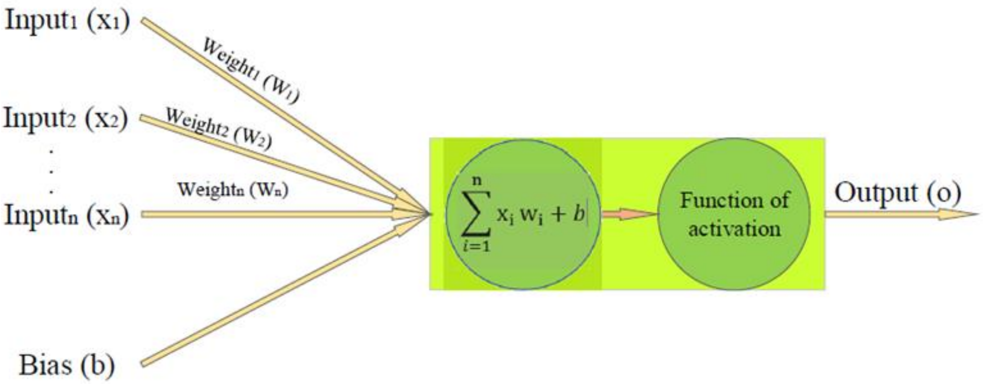


Figure 7. A schematic illustration of the neuron at an ANN.

The flux of information in Feed-forward ANNs flows from the input layer to the output layer, and the ANN predictions link weights throughout this process. In the backpropagation phase, meanwhile, learning algorithms such as gradient descent newton method, conjugate gradient, Levenberg-Marquardt algorithm, and qua-si-Newton method, to mention a few, are used to enhance the projected weights. In this research, Levenberg-Marquardt (LM) is employed since it performs better in the topic and is faster than other algorithms [16]. The revised weights and bias in the LM algo-rithm are computed as follows in Equation (12):

$$\mathrm{Wb}_{\mathrm{i+1}} = \mathrm{Wb}_{\mathrm{r}} - \left[\mathrm{J_m}^{\mathrm{r}}\mathrm{J_m} + \alpha\mathrm{I_m}\right]^{-1}\mathrm{J_m}^{\mathrm{R}}\varepsilon, \tag{12}$$

where $\mathrm{Wb}_{\mathrm{i+1}}$ is the updated biases and weights, $\mathrm{J_m}$ is the Jacobian matrix that considers the biases and weights for the first derivatives of the network errors, and $\mathrm{I_m}$, ε, and α are identity matrix, vector of ANNs error, and positive real numbers damping factor, respectively.

3.4. Proposed Model

The architecture of an ANN influences both speed and accuracy. ANN architectures are often characterized through trial and error. Nonetheless, a technique for achieving the optimum architecture must be developed. As a result, many optimal ANNs with varying amounts of hidden layers and neurons are required. MOGOA is crossed with an ANN in this research for that purpose. Figure 8 shows the various stages of MOGOA's hybridization with ANN (ANNMOGOA) with further explanation:

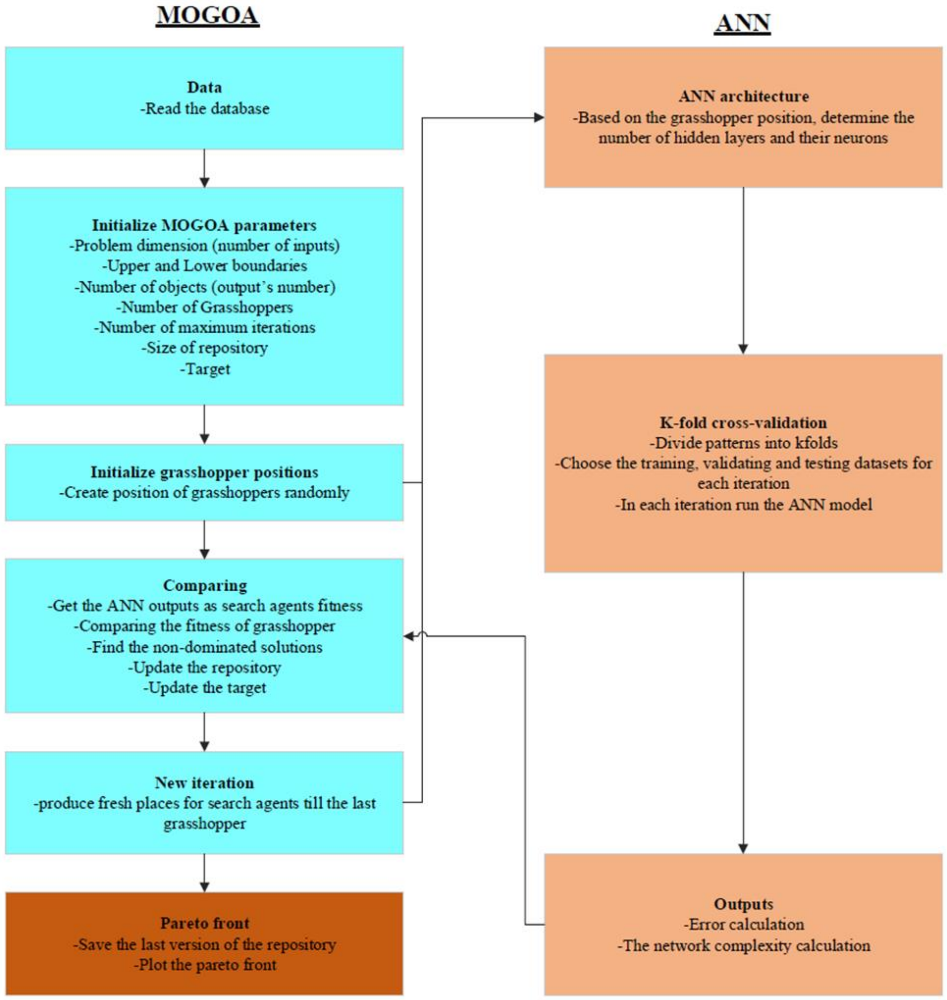

MOGOA
ANN
Data
-Read the database
Initialize MOGOA parameters
-Problem dimension (number of inputs)
-Upper and Lower boundaries
-Number of objects (output's number)
-Number of Grasshoppers
-Number of maximum iterations
-Size of repository
-Target
Initialize grasshopper positions
-Create position of grasshoppers randomly
Comparing
-Get the ANN outputs as search agents fitness
-Comparing the fitness of grasshopper
-Find the non-dominated solutions
-Update the repository
-Update the target
New iteration
-produce fresh places for search agents till the last grasshopper
Pareto front
-Save the last version of the repository
-Plot the pareto front
ANN architecture
-Based on the grasshopper position, determine the number of hidden layers and their neurons
K-fold cross-validation
-Divide patterns into kfolds
-Choose the training, validating and testing datasets for each iteration
-In each iteration run the ANN model
Outputs
-Error calculation
-The network complexity calculation

Figure 8. ANNMOGOA flowchart.

- Data Normalization

Because the network error increases when multiple types of input with varying ranges are used, all data should be standardized. The following equation was being used to adjust inputs and outputs in the [−1, 1] range in this investigation.

$$e_{norm} = \frac{2(e - e_{min})}{(e_{max} - e_{min})} - 1 \tag{13}$$

e_{norm}, e_{min}, and e_{max} are the normal, minimum, and maximum values of e, respectively.

- Cross-validation using K-fold

Every ANN requires data collection to comprehend the output and input rela-tionship and anticipate undiscovered patterns. Furthermore, its performance must be confirmed during this procedure, and it must be tested in the end phase. As a result, the dataset is divided into three categories: training, validating, and testing. The k-fold cross-validation method is used in this study to reduce network over-fitting [19–22]. Following randomizing the dataset, it is split into k distinct folds of equally sized and the network is run for k times in this approach. Its first fold is utilized as a validating and testing dataset, while the remaining folds will be used as a training dataset. The second fold is then used to validate and test the network for the second time. Eventually, in the kth iteration, the kth fold is used to validate and test the ANN while the other folds are used to train it. The final result is the mean of all these k times of executing the ANN's outputs. Every one of the patterns is being used for testing, validating, and training purposes as a result of this technique, making the ANN more accurate. The k-fold cross-validation approach is shown in Figure 9.

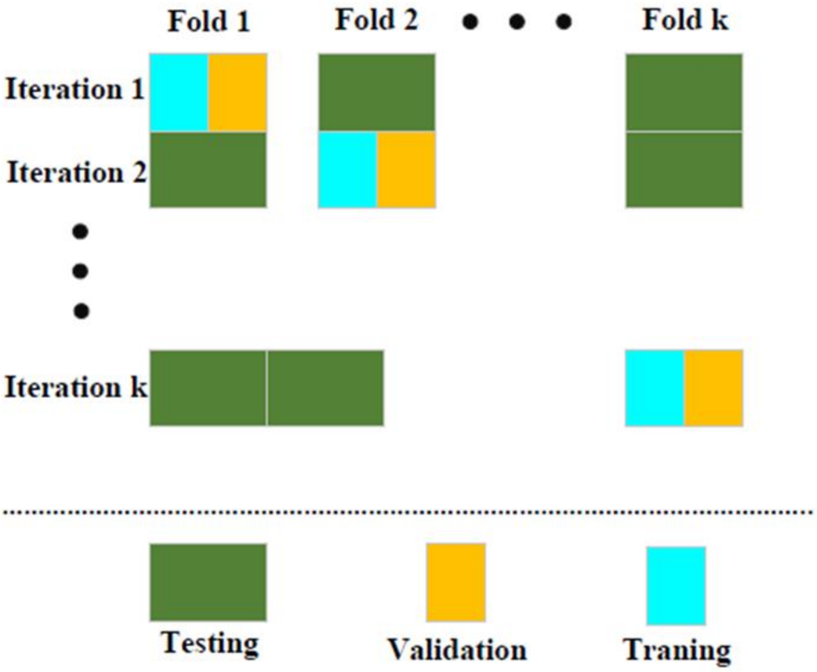


Figure 9. K-fold cross-validation.

- Initializing MOGOA's parameters

Every grasshopper position in this research is split into two halves, as shown in Figure 10. The first component is a binary coding system that assigns a numerical value to the numbers of hidden layers. To put it another way, if the first part's ith value is one, then ith hidden layer is active, and if its number is zero, the hidden layer is inactive. The number of hidden neurons in each hidden layer is indeed the subject of the second section. For example, part two of the position's ith value reflects the number of neurons in the ith hidden layer.

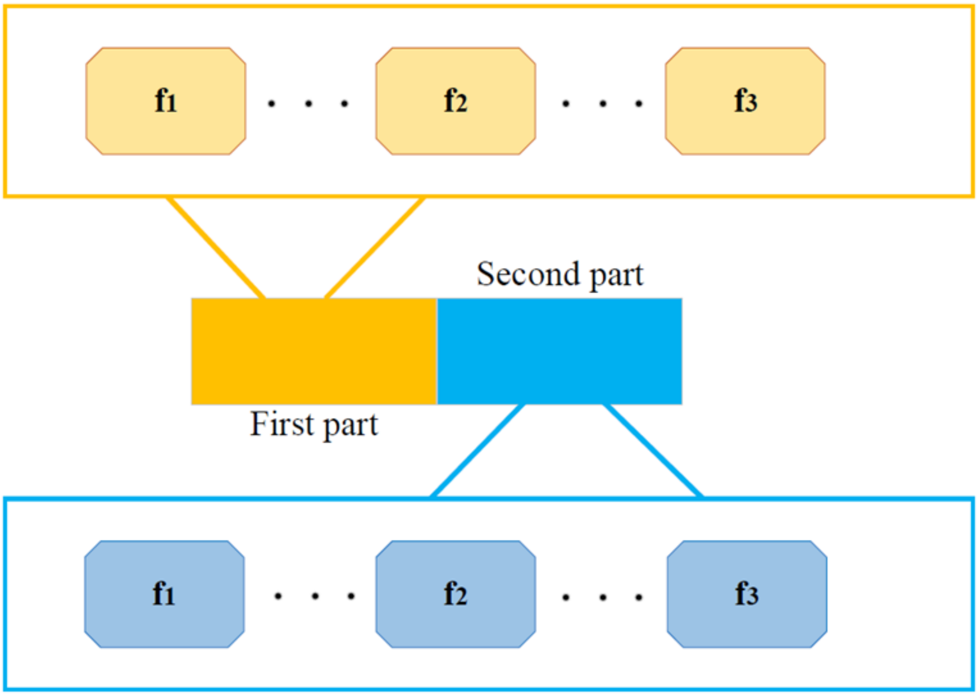


Figure 10. The structure of a grasshopper position.

- Grasshopper's positions initializing

Grasshoppers' starting positions are produced at random using upper and lower limits.

- Grasshopper's performance computation

ANNMOGOA is a method for simultaneously reducing complexity and errors to the minimum level. A network error is defined as the mean error of the k-fold cross-validation technique. OBJ is also useful for comparing network performance be-cause it displays the mean absolute error (MAE), Pearson correlation coefficient (R), and root mean square (RMSE).

$$R = \frac{P_d \sum_{i=1}^{P_d} R_e R_m}{\left(P_d \sum_{i=1}^{P_d} R_e^2 - \left(\sum_{i=1}^{P_d} R_e\right)^2\right)\left(P_d \sum_{i=1}^{P_d} R_m^2 - \left(\sum_{i=1}^{P_d} R_m\right)^2\right)}; \tag{14}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{P_d} \sum_{i=1}^{P_d} (R_e - R_m)^2}; \tag{15}$$

$$\mathrm{MAE} = \frac{1}{P_d} \sum_{i=1}^{P_d} |R_e - R_m|; \tag{16}$$

$$\mathrm{OBJ} = \left(\frac{T_t - T_{vt}}{T_t + T_{vt}}\right) \frac{\mathrm{RMSE}_t + \mathrm{MAE}_t}{R_t + 1} + \left(\frac{2T_{vt}}{T_t + T_{vt}}\right) \frac{\mathrm{RMSE}_{vt} + \mathrm{MAE}_{vt}}{R_{vt} + 1} \tag{17}$$

Here, T_{vt} is the number of patterns in the validation and testing part, T_t is the number of patterns in the training part, and P_d is the number of patterns in the dataset. The experimental analysis' result is R_e, and the model result is R_m, which is connected to the d^{th} record [23,24]. The network's complexity is defined in this study as the total of its weights and biases, which is calculated as follows:

$$Z_{wb} = \sum_{i=0}^{T} \left[U_i Q_i \left(Q_{i-U_{i+1}+2} + 1 \right) \right] + Q_{i+1} - Q_0 \tag{18}$$

Here, U_0 and U_{i+1} are both 1, Q_0 is the number of neurons in the ANN's input layer, Q_i is the number of neurons in the ANN's ith hidden layer, and Q_{i+1} is the number of neurons in the ANN's output layer [25].

- Using MOGOA

Non-dominated grasshoppers are described in this phase and evaluated to grasshoppers in the established repository. The complete none-dominated grasshoppers are then stored in the repository, then if the repository is full, a sufficient number of repository representatives are removed by using a technique described in the previous section. Following that, the location of the target will be adjusted using the method described above, and then the position of all grasshoppers will be adjusted. This process will be repeated till the last iteration is completed.

4. Materials and Data Collection

4.1. Mix Proportion of Mortar

In order to define the strength of a concrete or a mortar, it is crucial to understand and consider its mix design. Many methods exist to accurately design the mix propor-tion of concrete, such as Building Research Establishment (BRE). However, it can be difficult to apply to all concrete types, and in particular to zero-slump concrete, which are the most used ones in 3DP concrete applications.

In this paper four effective parameters were used as ANN's inputs: water-cement ratio (W/C), amount of coarse aggregate (CA), amount of fine aggregate (FA), amount of super-plasticizer (S). The only ANN's output is the compressive strength (CS). Figure 11 shows the detailed materials and frequency of used materials of cement and supplementary cementitious materials, such as fly ash (FA) and slag (S).

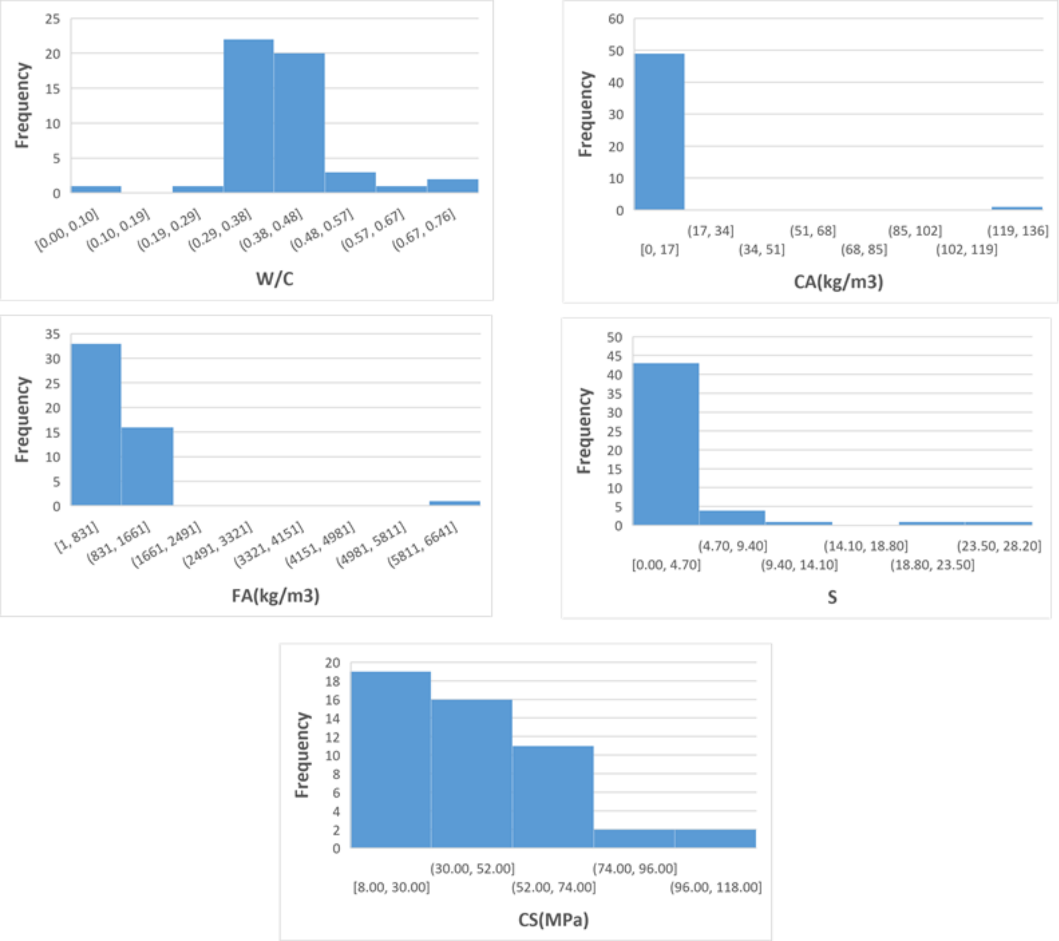


Figure 11. Histogram ANNs' inputs and outputs.

4.2. Fresh Properties of Mortar

The fresh state of concrete or mortar is considered the most vital stage in the 3DP process. The way of handling operations is reflected in the product aesthetics of the printed structure and mechanical strength properties. Figure 12 shows the general mix process and experimental programs which is necessary during the process of extrusion 3D printing.

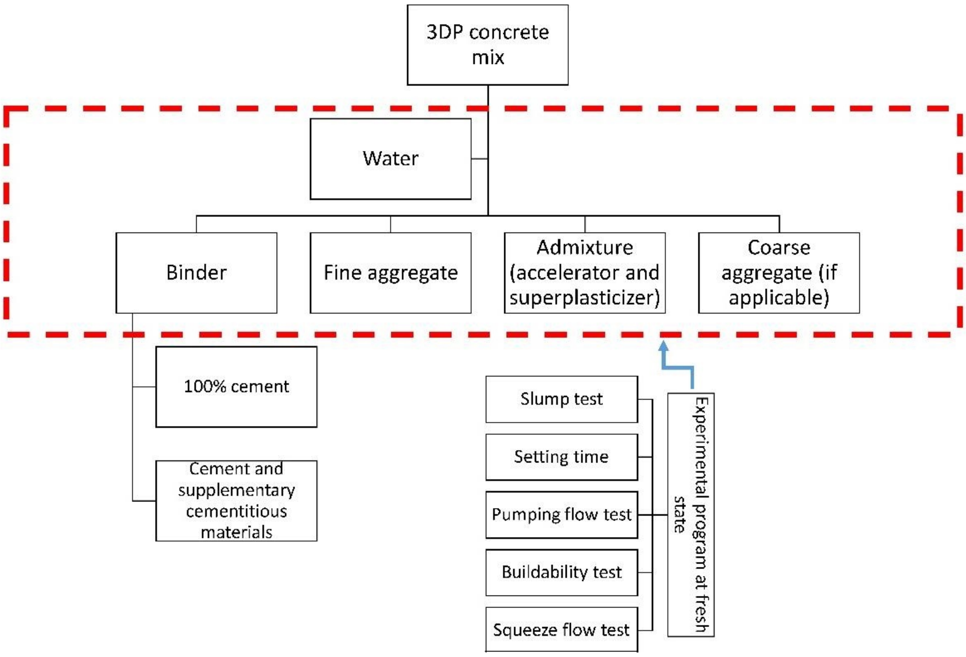

3DP concrete mix
Water
Binder
Fine aggregate
Admixture (accelerator and superplasticizer)
Coarse aggregate (if applicable)
100% cement
Cement and supplementary cementitious materials
Slump test
Setting time
Pumping flow test
Buildability test
Squeeze flow test
Experimental program at fresh state

Figure 12. Schematic illustration of basic mix materials and experimental test in 3DP concrete.

5. Results and Discussion

Table 1 shows the adjustment parameters determined through the trial-and-error method prior to running the model. After fitting the model 10 times, the Pareto front depicted in Figure 13 is the best Pareto front. There are two axes in this graph: com-plexity and error. The OBJ value was utilized as the error in this study, while the com-plexity was defined as the number of connecting weights in the ANN. As it can be ob-served, there are five non-dominated grasshoppers on the Pareto front after generating 150,000 distinct ANNs, and the complexity is growing as the error reduces. In other words, a more accurate network comes with a more sophisticated structure. The archi-tectures of the simplest algorithm (ANNMOGOA-3) and more complex algorithm (ANNMOGOA-1) are shown in Figure 14, and their weights and biases are listed in the Appendix B.

Table 1. Adjustment parameters of the proposed model.

Parameters	Values
The total number of runs	10
Neurons’ maximum number in each hidden layer	16
The maximum hidden layers’ number	3
The size of repository	50
Maximum number of iterations	100
Agent’ number	30
Activation function neurons of hidden layers	Hyperbolic tangent sigmoid
Training algorithm of ANNs	Levenberg–Marquardt
Activation function neurons of output layer	Linear

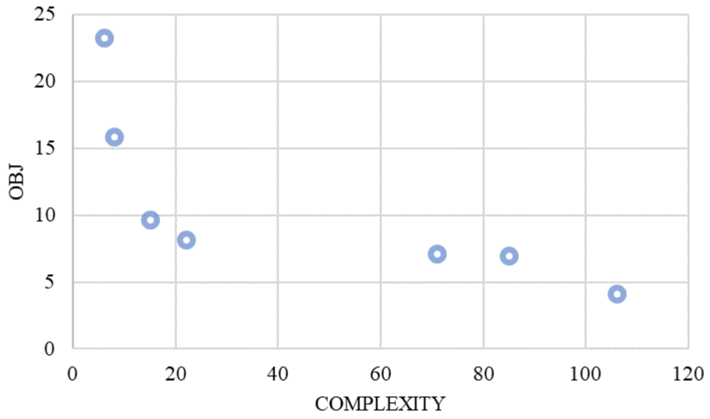


Figure 13. Non-dominated solutions (ANNs).

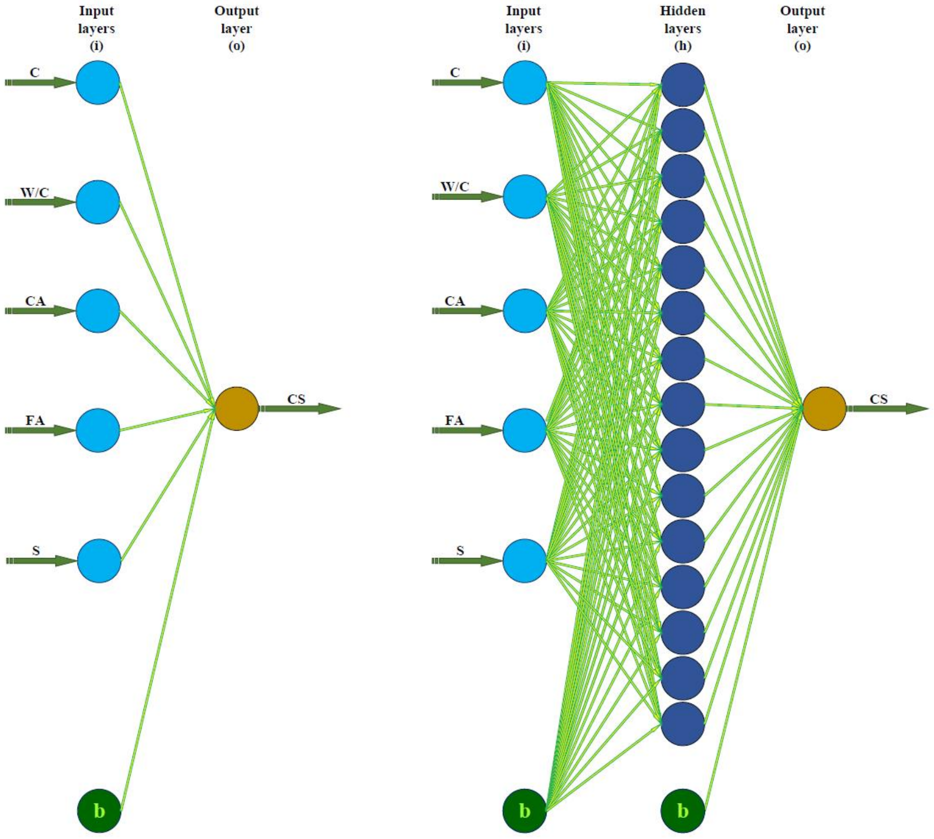


Figure 14. Architecture of ANNMOGOA-2 and ANNMOGOA-1.

A collection of statistical indicators is utilized to compare the behavior of the var-ious networks. Aside from the statistical indicators indicated in the technique section, the following parameters are used: mean bias error (MBE), mean absolute percentage error (MAPE), and scatter index (SI).

$$MBE = \frac{1}{D}\sum\nolimits_{i=1}^{D}(R_d - O_d); \tag{19}$$

$$MAPE = \frac{100}{D}\ \sum\nolimits_{i=1}^{D}\frac{R_d - O}{R_d}; \tag{20}$$

$$SI = RMSE/\overline{O} \tag{21}$$

O_d and O are the dth pattern's experimental and average value experimental outcomes, respectively, D is the total number of patterns, and R_d is the dth pattern's model result. If an ANN's SI value is greater than 0.3, it has "poor performance,"; between 0.2 and 0.3, it has "fair performance"; between 0.1 and 0.2, it has "good performance"; and for less than 0.1, it has "excellent performance", according to SI (Li et al., 2013). Table 2 also includes the designs and complexity of all five networks, as well as statistical indicators for ANNs.

Table 2. Information of NDSs model.

ANN	RMSE (MPa)	MAPE (%)	MAE (MPa)	MBE (MPa)	SI	R	OBJ (MPa)	Complexity	Structure
ANNMOGOA-1	4.49	8.75	2.58	0.19	0.11	0.98	4.12	106	5-15-1
ANNMOGOA-2	18.69	52.40	14.47	−1.14	0.44	0.49	23.28	6	5-1
ANNMOGOA-3	7.65	16.66	5.75	−0.70	0.18	0.93	8.167	22	5-3-1
ANNMOGOA-4	6.29	13.17	4.28	−0.01	0.15	0.96	6.99	85	5-12-1
ANNMOGOA-5	5.99	10.37	3.50	0.91	0.14	0.96	7.11	71	5-10-1
ANNMOGOA-6	10.74	20.25	6.58	0.01	0.25	0.87	9.69	15	5-2-1
ANNMOGOA-7	14.61	27.60	9.15	−0.83	0.35	0.73	15.84	8	5-1-1

Table 2 shows that ANNMOGOA-2 is the simplest network with 6 connecting weights and no hidden layers, followed by ANNMOGOA-7 with a single neuron in its hidden layer and 8 linking weights. ANNMOGOA-6, ANNMOGOA-3, ANNMOGOA-5, and ANNMOGOA-4 have 15, 22, 71, and 85 connecting weights, respectively. With fif-teen neurons in its only hidden layer and a complexity score of 106, ANNMOGOA-1 is the most complicated network. On the other hand, ANNMOGOA-1 is the most accurate model, with an OBJ value of 4.12, while ANNMOGOA-2 is the least accurate model, with an OBJ value of 23.28; all other models are in between. The RMSE of ANNMOGOA-1 is 4.49 MPa, which is about 75% lower than that of ANNMOGOA-2 with a value of 18.69. Furthermore, ANNMOGOA-2 has an MAE value of 14.47 MPa, which is about six times that of ANNMOGOA-1. Furthermore, when it comes to MAPE, the most accurate model is ANNMOGOA-1, which has a MAPE value of 8.75 percent, followed by ANNMOGOA-5, which has a MAPE value of 10.37 percent. The following five models are ANNMOGOA-4, ANNMOGOA-3, ANNMOGOA-6, ANNMOGOA-7, and ANNMOGOA-2, which have MAPE values of 13.17 percent, 16.66 percent, 20.25 percent, 27.60 percent, and 52.40 percent, respectively. So, ANNMOGOA-2 is the least accurate model. The Pearson correlation coefficient (R) measures the similarity between anticipated and measured values. This indication might be anything between 0 and 1. ANNMOGOA-1 has very good R-value that is equal 0.98. ANNMOGOA-2 and ANNMOGOA-7 have "poor performance", whereas ANNMOGOA-1, ANNMOGOA-3, ANNMOGOA-4, and ANNMOGOA-5 have "good performances", according to the SI value. ANNMOGOA-6 has "fair performances". MBE illustrates ANNMOGOA-1, ANNMOGOA-5, and ANNMOGOA-6 overestimate the compressive strength while ANNMOGOA-2, ANNMOGOA-3, ANNMOGOA-4, and ANNMOGOA-7 underestimate that. Figure 15 indicates the estimated values of models against measured values.

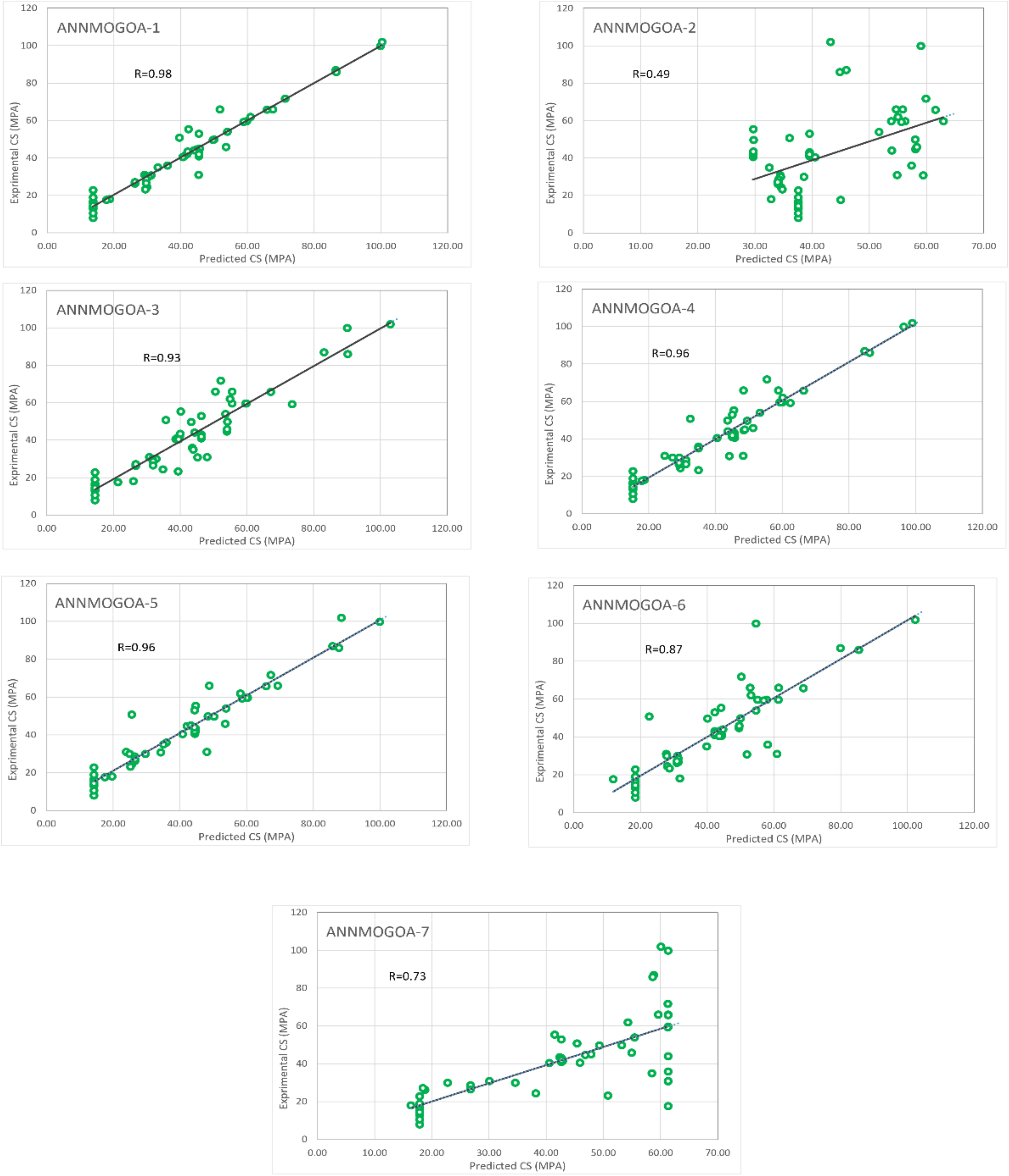


Figure 15. Predicted vs. experimental compressive strength.

A bar chart is included in Figure 16 to allow for a clearer comparison. They have the highest and lowest values for OBJ in ANNMOGOA-2 the ANNMOGOA-1, respectively, according to the bar chart, indicating that ANNMOGOA -1 is the best accurate model overall.

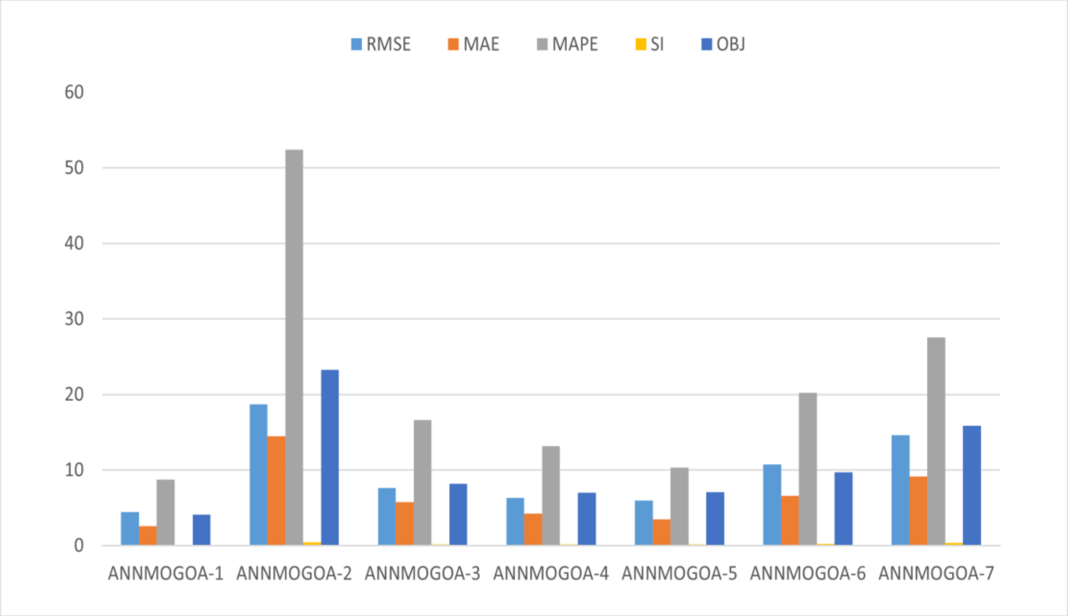


Figure 16. Models' statistical indicators.

Future work is necessary to input further information in the processes of iteration for machine learning such as admixtures, different types of cement and supplementary cementitious materials. These materials are factors to change the result of calculation; however, it could not be always perfect due to the authors and journal publishers not being mandated to provide all data's or sometimes due to secrets of production from industry not being permitted to publish those data. However, the concrete/mortar printing is not only affected by concrete mix proportion but also affected by other factors while handling the printing process and after printing processes such as humidity, temperature and curing condition.

6. Conclusions

Based on the present study, ANN can be used to predict the compressive strength of 3DP concrete. However, the difficulty of this method is that the accuracy of the model depends on the number of patterns. Unfortunately, there are still limited studies and patterns concerning 3DP concrete's compressive strength. In contrast to the low number of patterns in the field, proposed model has acceptable results and by increasing the number of studies through time more accurate models can be developed.

- Seven different networks with different complexities and accuracies are presented. From these algorithms, users can choose which model suits their project the most based on their limitations. It is obvious that if they need more accurate results they need to choose a network with a more complex structure.
- Since the maximum number of hidden layers in this study is three, it is proved that for predicting 3DP concrete's compressive strength, with a well-developed method, the results of a network with only one hidden layer can be accurate enough and there is no need for more complex net-works.
- The correlation coefficient of three out of seven networks (ANNMOGOA-1, ANNMOGOA-4, and ANNMOGOA-5) is more than 0.96, which is accurate enough to be accepted.
- Based on SI, four networks have good performance in predicting the compressive strength of 3DP concrete.
- Considering MAPE, the accuracy of ANNMOGOA-1 is about 92%, which is a high value of accuracy.

Author Contributions: Conceptualization, P.S.; methodology, H.I. and A.K.; software, H.I.; validation, A.K.; investigation, P.S.; data curation, V.L.; writing—original draft preparation, A.K., H.I. and P.S.; writing—review and editing, V.L. and P.S.; supervision, G.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research is not received any grant and external funding.

Data Availability Statement: The data and information reports will be available on request.

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

Appendix A

Table A1. The table shows the content of the material in the mix design of 3DP concrete or mortar.

Ref.	Cement (kg/m^3)	Water (kg/m^3)	W/C	Coarse Aggregate (kg/m^3)	Fine Aggregate (kg/m^3)	Superplasticizer (kg/m^3)	Compressive Strength (MPa) (28 days)
Shakor, et al. [26]	300	112	0.37		360	1.499852	50.82
Shakor, et al. [27]	300	1000	3.33		300	2.5	59.7
	300	1032	3.44		300	3.33	59.7
	300	1032	3.44	100	300	2.5	59.7
Nerella and Mechtcherine [28]	627	263.3	0.42		1391	4.7	71.8
	391	164.2	0.42		1260	7.82	99.9
Kazemian, et al. [29]	600	259	0.43		1379	0.3	44.7
	540	259	0.43		1357	0.864	49.9
	600	259	0.43		1379	0.36	45.1
	600	259	0.43		1379	0.9	45.9
Sanjayan, et al. [30]	300	114	0.38		450		13
	300	114	0.38		450		8
	300	114	0.38		450		16.8
	300	114	0.38		450		16
	300	114	0.38		450		13.9
	300	114	0.38		450		22.8
	300	114	0.38		450		14.5
	300	114	0.38		450		10.6
	300	114	0.38		450		19
Assaad, et al. [31]	615		0.43		1340	6.46	30.8
							26.9
Annapareddy, et al. [32]	21.6 slag 122.4 FA 12 SF	120			243		18.1
	138	906.9			221.7		23.1
Ding, et al. [33]	300	105			300	0.83	31
	300	108.375			300	1.03	30
	300	81.75			300	1.25	24.5
	300	118.5			300	1.85	23.3
Ting, et al. [34]	300 43 SF 86 FA	197.1429			514.2857		30
Malaeb, et al. [35]	37.5	144			48	0	40.6
	37.5	126			48	0.15	41.5
	37.5	117			48	0.3	42.3
	37.5	114			48	0.33	43.5
	37.5	108			48	0.39	55.4
Le, et al. [36]	300	111			642.8571	3	102

Table A1. *Cont.*

Ref.	Cement (kg/m^3)	Water (kg/m^3)	W/C	Coarse Aggregate (kg/m^3)	Fine Aggregate (kg/m^3)	Superplasticizer (kg/m^3)	Compressive Strength (MPa) (28 days)
Panda, et al. [37]	572.34 FA 35.52 Slag 101.86 SF 140.739 K-Si	144.09			1219.74	10.05	36
Weng, et al. [38]	300 300 FA 30 SF	90	0.3		150	1.3	49.7
Panda, et al. [39]	83.55 FA 5.04 Slag 10.08 SF	124.8			148.65	2.64	35
Khalil, et al. [40]	682.75	236.25			850	1.76	87
	675	236.25			800	1.76	86
Ma, et al. [41]	300	115.7143			514.2857	1.242857	43.1
	300	115.7143			514.2857	1.242857	41
	300	115.7143			514.2857	1.242857	41
	300	115.7143			514.2857	1.242857	43
	300	115.7143			514.2857	1.242857	53
	300	115.7143			514.2857	1.242857	42
Hack, et al. [42]	500	160	0.32		1180	5	59.3
Rushing, et al. [43]	300		0.47	300	690	975	40.5
Panda, et al. [44]	300 675 FA 250 SF	350			1220		54
Van Der Putten, et al. [45]	620.5	226.5			1241	0.93	62
Lee, et al. [46]	580 166 FA 83 SF	232			1146	8.29	66
Dressler, et al. [47]	600	270			1258	1.8	59.9
							64.8
							66
							65.7
Li, et al. [48]	259.2	345.6			864.1	17.3	44.09
							41.93
	480.2	327.4			589.4	13	17.66
							15.02
Joh, et al. [49]	576 79 SF 172 FA	240			1154	8.27	23.5 31
Meurer and Classen [50]	550	280			1172	22	65.8
Álvarez-Fernández, et al. [51]	100	50	0.5		250	0.5	26.3
	100	46	0.46		250	1	28.7
	100	46	0.46		250	1	26.6
	100	46	0.46		250	0.5	27.2

Appendix B

(1) Biases and weights of the ANNMOGOA-1 model

Input layer weights (IW)

$$= \begin{bmatrix} 1.25 & -1.00 & 1.35 & -0.18 & -0.67 & 4.01 & -0.31 & -1.07 & -0.55 & -1.40 & 0.41 & -1.73 & 1.18 & 0.03 & 0.66 \\ 1.12 & 1.55 & 0.42 & 0.08 & 0.86 & 0.07 & -0.48 & -0.44 & -0.91 & -0.79 & 0.25 & 1.25 & 0.80 & -0.07 & 1.01 \\ -0.53 & 1.57 & -0.61 & 0.47 & 1.32 & -0.88 & -1.16 & 0.91 & -2.19 & 1.71 & 1.42 & 0.32 & 0.88 & 0.85 & 1.48 \\ 0.24 & 0.26 & -1.29 & -2.30 & 1.95 & -0.49 & -0.34 & -1.54 & 1.87 & 0.60 & 3.65 & -1.77 & 0.35 & 0.10 & 1.19 \\ -0.21 & 0.77 & -0.49 & -1.46 & -0.51 & -1.36 & -0.93 & 0.04 & 0.93 & -2.79 & -4.55 & -1.63 & -0.07 & -1.18 & -2.66 \end{bmatrix}$$

$$\begin{aligned}&\text{Input layer bias (IB)}\\ &= \begin{bmatrix} 0.23 & -0.11 & 0.18 & -0.44 & -0.23 & 1.02 & 0.82 & -0.59 & 2.38 & -0.37 & -1.84 & 0.30 & -0.38 & -0.94 & -1.05 \end{bmatrix}\end{aligned}$$

$$\begin{aligned}&\text{Hidden layer weights (HW)}\\ &= \begin{bmatrix} 1.04 & 0.50 & 2.79 & -2.40 & 2.12 & -1.49 & -0.46 & -2.90 & -0.13 & 3.15 & -2.69 & 0.12 & 0.09 & -0.39 & -2.17 \end{bmatrix}\end{aligned}$$

$$\text{Hidden layer bias (HB)} = \begin{bmatrix} -1.64 \end{bmatrix}$$

(2) Biases and weights of the ANNMOGOA-2 model

$$\text{Input layer weights (IW)} = \begin{bmatrix} -0.05 \\ 0.30 \\ -0.07 \\ 0.33 \\ 0.09 \end{bmatrix}$$

$$\text{Input layer bias (IB)} = [0.05]$$

(3) Biases and weights of the ANNMOGOA -3 model

$$\text{Input layer weights (IW)} = \begin{bmatrix} -2.79 & -0.34 & 0.10 \\ 0.75 & -1.41 & 0.28 \\ -0.34 & 1.71 & 0.29 \\ -2.44 & 2.44 & 4.32 \\ -6.66 & -3.48 & -0.80 \end{bmatrix}$$

$$\text{Hidden layer weights (HW)} = \begin{bmatrix} 1.38 \\ -1.54 \\ 1.61 \end{bmatrix}$$

$$\text{Input layer bias (IB)} = \begin{bmatrix} 0.52 & -2.45 & 0.98 \end{bmatrix}$$

(4) Biases and weights of the ANNMOGOA -4 model

$$\text{IW} = \begin{bmatrix} 0.14 & -0.92 & -0.31 & -0.65 & 1.52 & 0.13 & 1.18 & -0.38 & -0.53 & -2.16 & -2.70 & -0.43 \\ -0.01 & 0.09 & 0.25 & -0.03 & -0.31 & -0.20 & -0.27 & -0.19 & -0.80 & 0.54 & 0.08 & -0.10 \\ -0.14 & -0.31 & 0.23 & -0.18 & -0.95 & -0.78 & 1.01 & 0.06 & -1.44 & -0.93 & 0.45 & -0.17 \\ 0.30 & 1.93 & 0.32 & -2.00 & 1.66 & -0.19 & 0.86 & 0.68 & -1.39 & -1.12 & 0.10 & -0.21 \\ -0.74 & 0.38 & -2.05 & 0.10 & 1.47 & 0.29 & -0.005 & 1.66 & 0.33 & 2.64 & 0.63 & -0.23 \end{bmatrix}$$

$$\text{IB} = \begin{bmatrix} 0.77 & 0.87 & -0.36 & -0.96 & -0.02 & 0.35 & 0.33 & 0.79 & -0.43 & 0.51 & 0.90 & 0.13 \end{bmatrix}$$

$$\text{HW} = \begin{bmatrix} 0.03 & -1.47 & 2.23 & -2.87 & -0.96 & 0.04 & -0.77 & 1.73 & 0.67 & 1.80 & -1.53 & -0.50 \end{bmatrix}$$

$$\text{HB} = [-0.59]$$

(5) Biases and weights of the ANNMOGOA-5 model

$$\text{IW} = \begin{bmatrix} -0.12 & -0.48 & -0.68 & -0.28 & 1.47 & 0.07 & -1.35 & -2.44 & 2.53 & -3.12 \\ 0.62 & -0.02 & -0.52 & 0.34 & 0.94 & -1.54 & 1.19 & 0.38 & -1.53 & -0.27 \\ -0.75 & 1.24 & -1.52 & 0.63 & 0.57 & -0.40 & -0.22 & 0.02 & 0.06 & -1.83 \\ -0.24 & 0.13 & -0.10 & -1.77 & 0.08 & -0.11 & -1.22 & -3.13 & 0.90 & -2.41 \\ 1.50 & 0.33 & -0.31 & 0.82 & -2.76 & 3.03 & -1.38 & 0.39 & -0.42 & 4.22 \end{bmatrix}$$

$$\text{IB} = \begin{bmatrix} -0.98 & -0.98 & 1.07 & -1.15 & 0.28 & 1.07 & 0.34 & -1.02 & 0.62 & 0.37 \end{bmatrix}$$

$$\text{HW} = \begin{bmatrix} -0.87 & 0.27 & -0.43 & 1.32 & 2.40 & 2.01 & 0.06 & -2.53 & -1.48 & 2.09 \end{bmatrix}$$

$$\text{HB} = [-0.10]$$

(6) Biases and weights of the ANNMOGOA-6 model

$$\begin{bmatrix} -0.06 & -4.18 \\ -2.70 & 2.56 \\ 1.92 & 1.35 \\ 1.43 & 17.16 \\ -0.24 & -6.02 \end{bmatrix}$$

$$\text{HW} = \begin{bmatrix} -1.32 \\ 1.22 \end{bmatrix}$$

$$\text{IB} = \begin{bmatrix} -0.79 & 1.13 \end{bmatrix}$$

$$\text{HB} = [-0.11]$$

(7) Biases and weights of the ANNMOGOA-7 model

$$\text{IW} = \begin{bmatrix} 0.93 \\ 80.02 \\ -3.71 \\ 0.38 \\ 9.29 \end{bmatrix}$$

$$\text{IB} = [85.37]$$

$$\text{HW} = [0.56]$$

$$\text{HB} = [-0.42]$$

References

1. Shakor, P.; Nejadi, S.; Paul, G.; Malek, S. Review of Emerging Additive Manufacturing Technologies in 3D Printing of Cementitious Materials in the Construction Industry. *Front. Built Environ.* **2019**, *4*, 85. [CrossRef]
2. Jiang, J.; Xu, X.; Stringer, J. Support structures for additive manufacturing: A review. *J. Manuf. Mater. Process.* **2018**, *2*, 64. [CrossRef]
3. Dutta, S.; Samui, P.; Kim, D. Comparison of machine learning techniques to predict compressive strength of concrete. *Comput. Concr.* **2018**, *21*, 463–470.
4. Al-Shayea, Q.K. Artificial neural networks in medical diagnosis. *Int. J. Comput. Sci. Issues* **2011**, *8*, 150–154.
5. Jiang, J.; Xiong, Y.; Zhang, Z.; Rosen, D.W. Machine learning integrated design for additive manufacturing. *J. Intell. Manuf.* **2020**, 1–14. [CrossRef]
6. Obermeyer, Z.; Emanuel, E.J. Predicting the future—Big data, machine learning, and clinical medicine. *N. Engl. J. Med.* **2016**, *375*, 1216. [CrossRef]
7. Yu, C.; Jiang, J. A perspective on using machine learning in 3D bioprinting. *Int. J. Bioprinting* **2020**, *6*, 253. [CrossRef] [PubMed]
8. Meng, L.; McWilliams, B.; Jarosinski, W.; Park, H.-Y.; Jung, Y.-G.; Lee, J.; Zhang, J. Machine Learning in Additive Manufacturing: A Review. *JOM* **2020**, *72*, 2363–2377. [CrossRef]
9. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
10. 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. [CrossRef]
11. Zhang, J.; Wang, P.; Gao, R.X. Deep learning-based tensile strength prediction in fused deposition modeling. *Comput. Ind.* **2019**, *107*, 11–21. [CrossRef]
12. Simpson, S.J.; McCAFFERY, A.R.; Hägele, B.F. A behavioural analysis of phase change in the desert locust. *Biol. Rev.* **1999**, *74*, 461–480. [CrossRef]
13. 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. [CrossRef] [PubMed]

14. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper optimisation algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [CrossRef]
15. Hamim, A.; Yusoff, N.I.M.; Omar, H.A.; Jamaludin, N.A.A.; Hassan, N.A.; El-Shafie, A.; Ceylan, H. Integrated finite element and artificial neural network methods for constructing asphalt concrete dynamic modulus master curve using deflection time-history data. *Constr. Build. Mater.* **2020**, *257*, 119549. [CrossRef]
16. 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. *Build. Mater.* **2020**, *248*, 118676. [CrossRef]
17. Tasdemir, Y. Artificial neural networks for predicting low temperature performances of modified asphalt mixtures. *Indian J. Eng. Mater. Sci.* **2009**, *16*, 237–244.
18. Behnood, A.; Golafshani, E.M. Machine learning study of the mechanical properties of concretes containing waste foundry sand. *Constr. Build. Mater.* **2020**, *243*, 118152. [CrossRef]
19. Prasad, B.R.; Eskandari, H.; Reddy, B.V. Prediction of compressive strength of SCC and HPC with high volume fly ash using ANN. *Constr. Build. Mater.* **2009**, *23*, 117–128. [CrossRef]
20. 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. [CrossRef]
21. 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. [CrossRef]
22. 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. [CrossRef]
23. Golafshani, E.M.; Talatahari, S. Predicting the climbing rate of slip formwork systems using linear biogeography-based programming. *Appl. Soft Comput.* **2018**, *70*, 263–278. [CrossRef]
24. Golafshani, E.M. Introduction of Biogeography-Based Programming as a new algorithm for solving problems. *Appl. Math. Comput.* **2015**, *270*, 1–12. [CrossRef]
25. Pala, M.; Özbay, E.; Öztaş, A.; Yuce, M.I. Appraisal of long-term effects of fly ash and silica fume on compressive strength of concrete by neural networks. *Constr. Build. Mater.* **2007**, *21*, 384–394. [CrossRef]
26. Shakor, P.; Nejadi, S.; Sutjipto, S.; Paul, G.; Gowripalan, N. Effects of deposition velocity in the presence/absence of E6-glass fibre on extrusion-based 3D printed mortar. *Addit. Manuf.* **2020**, *32*, 101069. [CrossRef]
27. Shakor, P.; Nejadi, S.; Paul, G. A Study into the Effect of Different Nozzles Shapes and Fibre-Reinforcement in 3D Printed Mortar. *Materials* **2019**, *12*, 1708. [CrossRef]
28. Nerella, V.N.; Mechtcherine, V. Studying the Printability of Fresh Concrete for Formwork-Free Concrete Onsite 3D Printing Technology (CONPrint3D). In *3D Concrete Printing Technology*; Sanjayan, J.G., Nazari, A., Nematollahi, B., Eds.; Butterworth-Heinemann: Oxford, UK, 2019; Chapter 16, pp. 333–347.
29. Kazemian, A.; Yuan, X.; Cochran, E.; Khoshnevis, B. Cementitious materials for construction-scale 3D printing: Laboratory testing of fresh printing mixture. *Constr. Build. Mater.* **2017**, *145*, 639–647. [CrossRef]
30. Sanjayan, J.G.; Nematollahi, B.; Xia, M.; Marchment, T. Effect of surface moisture on inter-layer strength of 3D printed concrete. *Constr. Build. Mater.* **2018**, *172*, 468–475. [CrossRef]
31. Assaad, J.J.; Hamzeh, F.; Hamad, B. Qualitative assessment of interfacial bonding in 3D printing concrete exposed to frost attack. *Case Stud. Constr. Mater.* **2020**, *13*, e00357. [CrossRef]
32. Annapareddy, A.; Li, M.; Tan, M.J.; Ting, A.G.H.; Tay, D.Y.W.; Ting, A.; Tay, D.; Annapareddy, A.; Li, M.; Tan, M. Effect of recycled glass gradation in 3D cementitious material printing. In Proceedings of the Conference on Progress in Additive Manufacturing (Pro-AM 2018), Singapore, 14–17 May 2018; Nanyang Technological University: Singapore, 2018.
33. Ding, T.; Xiao, J.; Zou, S.; Zhou, X. Anisotropic behavior in bending of 3D printed concrete reinforced with fibers. *Compos. Struct.* **2020**, *254*, 112808. [CrossRef]
34. Ting, G.H.A.; Tay, Y.W.D.; Qian, Y.; Tan, M.J. Utilization of recycled glass for 3D concrete printing: Rheological and mechanical properties. *J. Mater. Cycles Waste Manag.* **2019**, *21*, 994–1003. [CrossRef]
35. Malaeb, Z.; Hachem, H.; Tourbah, A.; Maalouf, T.; El Zarwi, N.; Hamzeh, F. 3d Concrete Printing: Machine And Mix Design. *Int. J. Civ. Eng. Technol.* **2015**, *6*, 14–22.
36. Le, T.T.; Austin, S.A.; Lim, S.; Buswell, R.A.; Gibb, A.G.F.; Thorpe, T. Mix design and fresh properties for high-performance printing concrete. *Mater. Struct.* **2012**, *45*, 1221–1232. [CrossRef]
37. Panda, B.; Leite, M.; Biswal, B.B.; Niu, X.; Garg, A. Experimental and numerical modelling of mechanical properties of 3D printed honeycomb structures. *Measurement* **2018**, *116*, 495–506. [CrossRef]
38. Weng, Y.; Li, M.; Tan, M.J.; Qian, S. Design 3D printing cementitious materials via Fuller Thompson theory and Marson-Percy model. *Constr. Build. Mater.* **2018**, *163*, 600–610. [CrossRef]
39. Panda, B.; Paul, S.C.; Tan, M.J. Anisotropic mechanical performance of 3D printed fiber reinforced sustainable construction material. *Mater. Lett.* **2017**, *209*, 146–149. [CrossRef]
40. Khalil, N.; Aouad, G.; El Cheikh, K.; Rémond, S. Use of calcium sulfoaluminate cements for setting control of 3D-printing mortars. *Constr. Build. Mater.* **2017**, *157*, 382–391. [CrossRef]
41. Ma, G.; Li, Z.; Wang, L. Printable properties of cementitious material containing copper tailings for extrusion based 3D printing. *Constr. Build. Mater.* **2018**, *162*, 613–627. [CrossRef]

42. Hack, N.; Dressler, I.; Brohmann, L.; Gantner, S.; Lowke, D.; Kloft, H. Injection 3D Concrete Printing (I3DCP): Basic Principles and Case Studies. *Materials* **2020**, *13*, 1093. [CrossRef]
43. Rushing, T.S.; Stynoski, P.B.; Barna, L.A.; Al-Chaar, G.K.; Burroughs, J.F.; Shannon, J.D.; Kreiger, M.A.; Case, M.P. Investigation of Concrete Mixtures for Additive Construction. In *3D Concrete Printing Technology*; Sanjayan, J.G., Nazari, A., Nematollahi, B., Eds.; Butterworth-Heinemann: Oxford, UK, 2019; Chapter 7, pp. 137–160.
44. Panda, B.; Ruan, S.; Unluer, C.; Tan, M.J. Improving the 3D printability of high volume fly ash mixtures via the use of nano attapulgite clay. *Composites Part B* **2019**, *165*, 75–83. [CrossRef]
45. Van Der Putten, J.; Snoeck, D.; Van Tittelboom, K. 3D Printing of cementitious materials with superabsorbent polymers. In Proceedings of the Durable Concrete for Infrastructure under Severe Conditions-Smart Admixtures, Self-responsiveness and Nano-additions, Ghent, Belgium, 10–11 September 2019; Magnel Laboratory for Concrete Research: Ghent, Belgium, 2019.
46. Lee, D.-g.; Yoo, B.-H.; Son, H.-J. Development of Shrinkage Reducing Agent for 3D Printing Concrete. *J. Korea Acad. -Ind. Coop. Soc.* **2019**, *20*, 37–43.
47. Dressler, I.; Freund, N.; Lowke, D. The effect of accelerator dosage on fresh concrete properties and on interlayer strength in shotcrete 3D printing. *Materials* **2020**, *13*, 374. [CrossRef]
48. Li, V.C.; Bos, F.P.; Yu, K.; McGee, W.; Ng, T.Y.; Figueiredo, S.C.; Nefs, K.; Mechtcherine, V.; Nerella, V.N.; Pan, J. On the emergence of 3D printable engineered, strain hardening cementitious composites (ECC/SHCC). *Cem. Concr. Res.* **2020**, *132*, 106038. [CrossRef]
49. Joh, C.; Lee, J.; Park, J.; Yang, I.-H. Buildability and Mechanical Properties of 3D Printed Concrete. *Materials* **2020**, *13*, 4919. [CrossRef]
50. Meurer, M.; Classen, M. Mechanical Properties of Hardened 3D Printed Concretes and Mortars—Development of a Consistent Experimental Characterization Strategy. *Materials* **2021**, *14*, 752. [CrossRef] [PubMed]
51. Álvarez-Fernández, M.-I.; Prendes-Gero, M.-B.; González-Nicieza, C.; Guerrero-Miguel, D.-J.; Martínez-Martínez, J.E. Optimum Mix Design for 3D Concrete Printing Using Mining Tailings: A Case Study in Spain. *Sustainability* **2021**, *13*, 1568. [CrossRef]