Prediction Of Compressive Strength Of High-performance Concrete Using Multi-layer Perceptron

Xu Wu, Guifeng Yan*, Wei Zhang, and Yuping Bao

Department of BIM Research, Nantong Institute of Technology, Nantong 226002, Jiangsu, China

Corresponding author. E-mail: yanguifeng1020@163.com

Received: Apr.16, 2023; Accepted: Aug. 27, 2023

The correlations between the mechanical properties of HPCs and their mixture compositions are complex, non-linear, and complex to characterize employing standard statistical methods. This paper aimed to estimate HPC's compressive strength using a machine learning algorithm including Multi-layer Perceptron (MLP) with an HPC mixed collection of 168 samples via eight input variables. In addition, three meta-heuristic optimizers have been used for improving the efficiency and accuracy of MLP, which are included Dandelion Optimization (DO), Aquila Optimizer (AO), and Sooty Tern Optimization Algorithm (STOA). After fitting the presented models, the developed models' predictive generalization and efficiency ability is evaluated against a set of performance parameters. All models used were found to perform as suitable in predicting outcomes, which can be employed for saving time and energy. As a result, Aquila's optimization had the most accurate by MLP compared to other hybrid models. MLAO3 obtained $R^2 = 0.994$ and $RMSE = 1.27(\text{MPa})$, which are the most suitable result compared to other models.

Keywords: High-performance concrete; Compressive strength; Multi-Layer Perceptron; Dandelion Optimization; Aquila Optimizer; Sooty Tern Optimization Algorithm.

http://dx.doi.org/10.6180/jase.202407_27(7).0004

1. Introduction

High-Performance Concrete (HPC) is determined as concrete made from suitable, properly mixed materials, combined, laid, transported, compacted, and cured based on a chosen mix design so that the resulting concrete has a placed structure environment to which the stress it undergoes and it is exposed during its lifetime [1]. In general, mineral admixtures usage benefits structural concrete through improved durability properties, sustainability, and microstructural properties advantages primarily belong to the industrial or agricultural utilization by-products that can cause pollution [2, 3]. The practical products employed in concrete design have been known for many years, and many studies have been published on by-products as partial replacements for Portland cement and the practical use of waste.

If cement is tendentiously replaced by mineral additives, hydration and pozzolanic reactions decrease more uniform hydrated compounds, generate a decrease in calcium hydroxide, and reduce pore and particle size. Refinement promotes denser pastes and less penetration of cementitious fluids [4, 5]. Since the mid-1990s, there has been an essential improvement in the production of mixed concrete containing Portland cement and two or three supplemental cementitious materials. The benefits of these specific kinds compared to their respective binary systems have motivated extensive investigation in these studies [6, 7].

Adding these reactive supplemental cementitious materials inbound creates for tendentiously replacing Portland cement and develops unique properties over time. Thus, adding enormous amounts of inorganic admixtures

to concrete can significantly improve concrete's physical properties, including durability, mechanical, and physical [8]. The concrete industry can reduce or replace common Portland cement with less environmental impact, cleaner, and energy consumption in the future may lead to fewer building materials by incorporating a binary or ternary system [9]. Fly ash has high pozzolanic activity and micro filler properties similar to silica fume [10].

ACI Commission 211-2008 recommended a fly ash replacement level of 15-25% for high-strength concrete. Numerous reports have shown that concrete containing large amounts of Class F fly ash has excellent mechanical properties and durability, including low permeability to chloride ions and other aggressive agents. Partial weight replacement of blast furnace slag reduces early strength but increases late strength and significantly improves the hardened concrete microstructure and durability [11–13]. Chen et al. [14] investigated a cement-fly ash-silica ternary cement system with fly ash content of 0–30% under autoclave curing, and higher strength properties can be obtained by improving the autoclave compressive. Rossi [15] studied the HPC mechanical performance composites containing fly ash and cement, achieving 225 (MPa) CS. Yazıcı et al. [16] studied a consequential improvement in mechanical properties, a decrease in the hydration heat, and a reduction in the HPC shrinkage effect by partially replacing cement with fly ash. Urbonas et al. [17] used finer ground fly ash to compensate for the strength loss by reducing the cement amount in HPC.

Over the past decade, mixed design methods for HPC have been an active research area [18]. Traditional approaches to HPC design are according to past empirical and experience relationships like multivariate linear regression. However, these methods cannot model the complex nonlinear relationship between interacting factors such as final concrete properties and mix constituents [19–21]. Recently, machine learning (ML) [22] and soft computing (SC) [23] approaches have been investigated for various purposes in structures and materials experimental mechanics. Among these approaches, artificial neural networks (ANNs) have been employed as standard tools for tasks expressed as regression issues due to their broad applicability [24–28].

ANNs have been suggested as non-linear regression tools to replace simple multivariate linear regression models for concrete mixture design problems [29–33]. Kasperkiewicz et al. [34] used a fuzzy ARTMAP neural network to model the compressive strength (CS) of HPC blends with just 6 variables. Jakubek and Waszczyszyn [35] used fuzzy weighted neural networks for compressive estimation of HPC. Naderpour [36] employed ANN to predict the recycled concrete CS. Mashhadban used a particle swarm optimization algorithm and his ANN to predict the mechanical properties of fiber-reinforced self-compacting concrete [37]. Jueyendah et al. [38] estimated the flexural and CS of cement mortars, including nano-silica and micro-silica, by creating SVM models with 4 various kernels. SVM-RBF showed the highest accuracy among the 4 methods by multiple kernels.

The current article's novelty is introducing a machine-learning (ML) method for CS of HPC prediction. In this article, one of the subsets of ANN is used for modeling: Multi-layer Perceptron (MLP). MLP is examined in three layers. In addition, some algorithms are used to optimize the model, which includes three meta-heuristic algorithms to increase accuracy and reduce errors. The algorithms consist of Dandelion Optimization (DO), Aquila Optimizer (AO), and Sooty Tern Optimization Algorithm (STOA). Combining the optimizers with the corresponding model forms a hybrid model in the MLDO, MLAO, and MLST frameworks. In addition, several evaluators have been used to evaluate hybrid models to select the most suitable model, which will be fully introduced in the following sections.

2. Materials and methodology

2.1. Data gathering

In this paper, several university laboratories assessed a data set of 168 samples of regular Portland cement containing different additives and cured under normal situations [39]. Table 1 indicates the eight input parameters used, including fly ash (Fa), binder (B), micro-silica (MS), water (W), superplasticizer (SP), coarse aggregate (CA), total aggregate (TA), sample age (Age). In addition, Compressive Strength (CS) is defined as the output. Ratios of input variables related to binder content were presented. Furthermore, the maximum (Max), minimum (Min), average (Avg), and standard deviation (St. Dev.) of variables are shown in Table 1. The laboratory experiment applied as follows:

Cubes that were 100 mm in size were made for figuring out CS. Cylinders with a diameter of 100 mm were also prepared and made some samples that were 200 millimeters long were used to measure how much they could be squished. That experimental task made beams 350 mm long, 75 mm wide, and 50 mm deep. A notch was 25 mm deep in the middle of the beams, and used these beams to do fracture tests. All samples were soaked in water at a temperature of 27 degrees Celsius, following the way it is done in Hong Kong. The scientists tested the cube's strength at different times: 3, 7, 28, 90, and 180 days. More tests were done after 28 and 56 days. In addition, they made three

Table 1. Statistical properties of inputs and CS.

Indicators	Variables								
	Input								Targets
	B (Kg/m³)	*FA/B* %	*MS/B* %	*CA/B* %	*CA/TA* %	*SP/B* %	*W/B* %	*Age* (day)	*CS (MPa)*
Min	394	0.0	0.0	2.1720	0.6	0.0	0.3	28	24
Max	500	0.55	0.1099	2.9061	0.6787	2.6	0.5	180	107.8
Avg	418.38	0.241	2.906	2.736	0.6283	1.033	0.4	68.57	64.03
St. Dev.	40.01	0.148	0.6787	0.2789	0.0204	0.622	0.0816	64.03	15.30

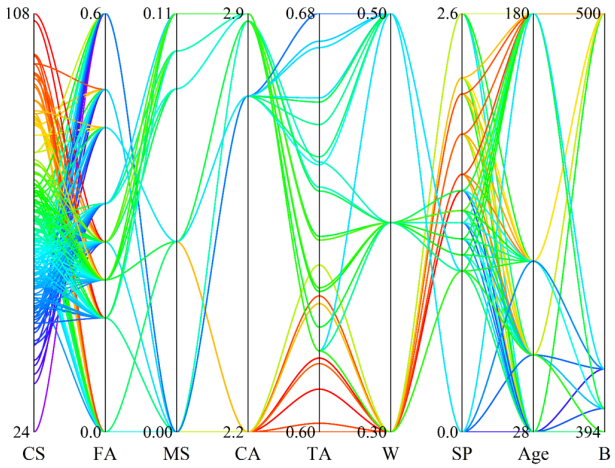

Fig. 1. The parallel plot of variables

specimens for each test of cube strength and fracture properties and two specimens for each test of cylinder strength, tensile splitting strength, and complete stress-strain curves.

Furthermore, Fig. 1 offers the parallel plot of variables based on CS. In Fig. 1, the relationship between inputs and CS is shown so that, for example, the lowest value of CS occurs when MS is at its lowest value and TA and W are at their highest value. On the other hand, the highest value of CS is obtained with the lowest value of CA, TA, and W. Moreover, Fig. 2 shows the impact of input on output variables.

2.2. Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron is a supervised ML technique similar to biological neural networks found in nature. The MLP

	B	FA/B	MS/B	CA/B	CA/TA	W/B	SP/B	age	CS
B	1								
FA/B	0.012	1							
MS/B	-0.289	-0.148	1						
CA/B	-0.997	-0.012	0.289	1					
CA/TA	-0.456	0.425	-0.327	0.414	1				
W/B	-0.810	-0.013	0.239	0.767	0.648	1			
SP/B	0.628	0.169	-0.001	-0.573	-0.694	-0.881	1		
age	0.142	0.013	-0.317	-0.142	0.151	-0.115	0.012	1	
CS	0.780	-0.350	-0.184	-0.757	-0.618	-0.792	0.602	0.408	1

Fig. 2. The impact of input on output.

is developed for learning inference and asset processes, formally called the prediction and training phases. MLP neural nets are widely used to model non-linear and complex processes occurring in the real world due to their adaptable approximation capabilities [40, 41]. The structure of MLP can be separated into 3 interconnected layers, including the input, hidden, and output layers. The input layer has some nodes equal to the predictor variables' number.

Furthermore, it has been shown that a single hidden layer MLP neural network can adequately model very complex non-linear functions with a hidden neurons sufficient number. The hidden layer neurons number is chosen, so the MLP neural network is essential for modeling the interest function. Few neurons tend to make the neural network perform poorly. Conversely, MLP neural nets with a neuron excessive number of are not only hard to train but prone to overfitting. The output layer nodes correspond to the modeled variable numbers.

To generalize the non-linear function (f), the function modeling task with one predictor uses an MLP neural net as $X \in R^D \rightarrow Y \in R^1$. Here, X and Y represent the input and output variables alternatively. The function (f) is defined in Eq. (1):

$$Y = f(X) = v_2 + M_2 \times (f_a\,(v_1 + M_1 \times X)) \tag{1}$$

Here M_1 and M_2 shows the hidden and output layers' weight matrixes alternatively. v_1 and v_2 are the hidden and output layers' bias vectors, respectively, and f_a is the function of activation.

The tan-sigmoid and log-sigmoid functions of activation are widely used. Their equations have been indicated alternatively in the following:

$$f_a(N) = \frac{1}{1+\exp(-N)} \tag{2}$$

$$f_a(N) = \frac{\exp(N) - \exp(-N)}{\exp(N) + \exp(-N)} \tag{3}$$

Here N shows the activation function of the input. Fig. 3 indicates the flowchart of MLP.

2.3. Aquila optimizer (AO)

The Aquila Optimizer algorithm simulates the manner of Aquila during a hunt by indicating the actions of the hunt's each step. Therefore, the optimization procedure of the AO algorithm is presented in 4 ways, including exploring the divergent explore space via outlining with a short gliding attack, selecting a explore area by flying high with a vertical dive, walking to prey, and flying low with a slow descent attack to utilize the convergent search space.

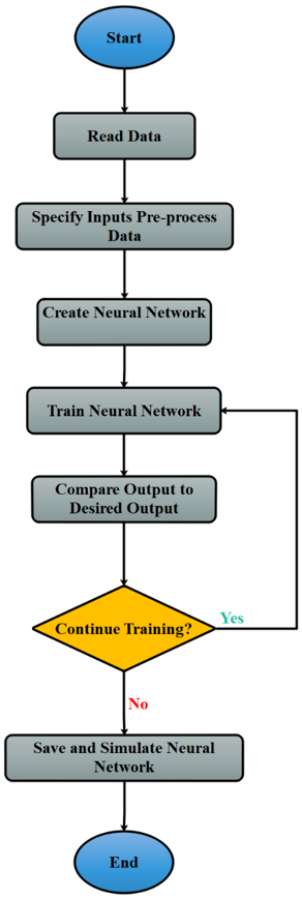

Fig. 3. The flowchart of MLP.

Aquila recognizes prey areas and selects the best hunting areas by hopping with a vertical hunchback in the first approach (S_1). AO searchers climb far to define regions of the explore space where loot is located. This manner is expressed as:

$$S_1(t+1) = S_{\text{best}}(t) \times \left(1 - \frac{\text{Iter}}{\text{Max Iter}}\right) + (S_m(t) - S_{\text{best}}(t) \times r) \tag{4}$$

Here $S_1(t+1)$ shows the next iteration's solution of t, $S_{\text{best}}(t)$ shows the solution of best achieved until t_m^{th} iteration, r shows the random value between 0 and 1, Iter and Max Iter are the current and maximum number of iterations, alternatively, and $S_m(t)$ is the mean value of locations of the current solution connected at the t^{th} iteration, which is computed as:

$$S_m(t) = \frac{1}{n}\sum_{i=1}^{n} S_i(t) \tag{5}$$

Here n shows the candidate solution's number. Aquila circles over the target prey, prepares the land, and then attacks if it finds a prey area from a high altitude in the second method (S_2). Here, the AO meticulously searches the chosen target prey's areas in practice for the attack. This manner is expressed as:

$$S_2(t+1) = S_{\text{best}}(t) \times L(D) + S_r(t) + (y - s) \times r \tag{6}$$

Here $S_2(t+1)$ shows the next iteration's solution of t, which is developed with the second explore approach, D shows the space of dimension, $S_r(t)$ shows a random solution at the t_m^{th} iteration in the range of (1, n), and $L(D)$ shows the distribution function of levy flight.

Suppose the prey area is designated, and the Aquila lands and is ready to charge. In that case, a recharge causes the Aquila to descend vertically to discover the prey response in the third approach. The approach is called Low Flying with a Slow Descent Attack. Also, the AO utilizes the target's chosen area to approach and attack its prey. This manner is expressed as:

$$S_3(t+1) = (S_{\text{best}}(t) - S_m(t)) \times c - r + ((ub - lb) \times r + lb) \times e \tag{7}$$

Here $S_3(t+1)$ the next iteration's solution of t is developed with the third explore approach. c and e show utilization tuning parameters and are fixed to small values (0.1), and ub, lb represent the lower and upper bound of the problem.

Aquila attacks prey on land following its stochastic movements when approaching the prey in the fourth approach. This approach is called walking and loot grabbing. Finally, AO attacks the booty in last place. This manner is represented as:

$$S_4(t+1) = Q \times S_{\text{best}}(t) - (O_1 \times S(t) \times r) - O_2 \times L(D) + r \times O_1 \tag{8}$$

Here $S_4(t+1)$ the next iteration's solution of t is developed with the third explore approach. Q indicates a quality function for balancing explore strategies, O_1 shows different movements of the AO used to track prey in flight, O_2 denotes the flight gradient of the AO employed for tracking the prey during the escape from the initial position (1) to the final position (t), and $S(t)$ shows the current solution at the t_m^{th} iteration. The flowchart of AO is demonstrated in Fig. 4.

2.4. Dandelion Optimization (DO)

Dandelion Optimization (DO) performs iterative optimization and population evolution according to population initialization and other metaheuristic algorithms inspired by

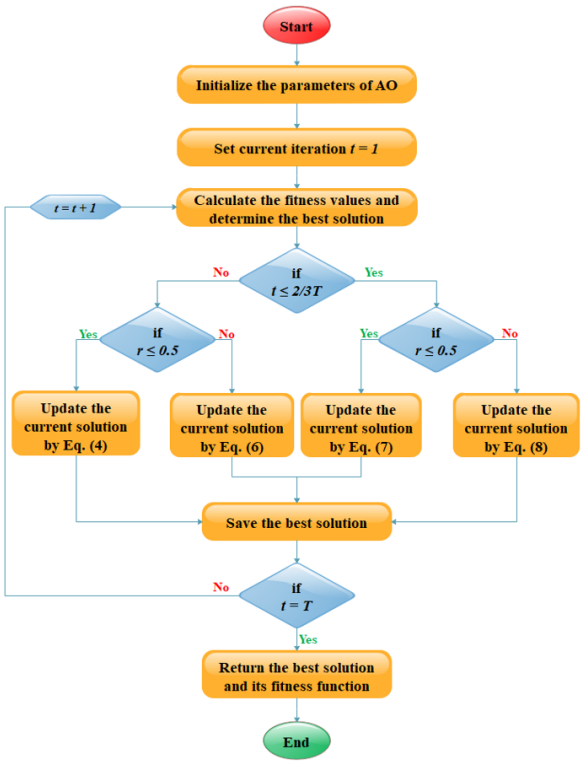

Fig. 4. The flowchart of AO

nature [42]. Each dandelion seed expressed a candidate solution assumed in the presented DO algorithm.

Dandelion seeds must meet a certain height before swimming away from their parents during the rising phase. Dandelion seeds rise to various heights under humidity, wind speed influence, etc. Here, the weather is separated into the following two positions.

The wind speed on a sunny day can be considered a log-normal distribution. The random numbers are distributed more along the Y axis with this distribution and making it more likely that dandelion seeds will migrate to distant regions. The wind blows dandelion seeds randomly to different locations in the exploration room. The wind's speed defines the dandelion's height seed rise. The higher the dandelion, the farther the seeds fly with stronger wind. The vortex above the dandelion seeds is always adapted to swirl upwards under the influence of wind speed. The corresponding formula in this situation can be determined as follows:

$$\begin{aligned} X(t+1) &= X(t) \\ &+ a \times l_x \times l_y \times \ln D \times (X(r) - X(t)) \end{aligned} \tag{9}$$

$$X(r) = r(1, D) \times (ub - lb) + lb \tag{10}$$

Here $X(t)$ is the dandelion's position during t iteration, $X(r)$ is the randomly chosen location in the explore distance during t iteration, $\ln D$ is a log-normal distribution subject to $\mu = 0$ and $\sigma^2 = 1$. In addition, a shows an adjusted parameter utilized for adapting the explore step length and can be defined as:

$$a = r \times \left(\frac{1}{T^2} t^2 - \frac{2}{T} t + 1 \right) \tag{11}$$

In early and later phases, such instabilities cause the algorithm to pay more attention to global exploration and switch to local searches. This is advantageous for ensuring exact convergence after a full global investigation. l_x and l_y are the coefficients of the dandelion lift component because of the separation vortex movement. Eq. (12) is used to compute variable dimension's forces:

$$\begin{aligned} l_x &= s \times \cos\theta \\ l_y &= s \times \sin\theta \\ s &= \frac{1}{e^{\theta}} \end{aligned} \tag{12}$$

Here θ shows a random number between $[-\pi, \pi]$. Due to the effects of air resistance and humidity on rainy days, the dandelion seeds are not blown up by the wind. In this case, the dandelion seeds are utilized in the neighborhood and using Eq. (13):

$$X(t+1) = X(t) \times v \tag{13}$$

Here v is utilized to adjust the Dandelion local search domain and can be determined as:

$$\begin{aligned} v &= 1 - r \times e \\ e &= \frac{1}{T^2 - 2T + 1} t^2 \\ &- \frac{1}{T^2 - 2T + 1} t + 1 \\ &+ \frac{1}{T^2 - 2T + 1} \end{aligned} \tag{14}$$

Where v shows "downward convex" oscillation that favors the algorithm's local exploitation, with significant steps in the early stages and shorter cross lengths in the later stages. At the end of the iterations, v uses stepwise method 1, ensuring that the population eventually converges on the optimal explore agent.

At the Descending stage, the DO algorithm focuses on search. Dandelion seeds will come down when they reach a certain distance. Brown movement is employed in DO to simulate dandelion trails. It is easy for an individual to traverse more explored communities during iterative updates since the Brownian move follows a normal distribution with each change. The average positional information after the ascent phase reflects the dandelion descent's stability.

This encourages the development of the entire population toward future-oriented communities. The equation of the related stage can be defined as:

$$X(t+1) = X(t) + a \times m_t \times (X(t_{\text{mean}}) - a \times m_t \times X(t)) \tag{15}$$

Here m_t is the motion of Brownian and is a random number from the distribution of standard normal, $X(t_{\text{mean}})$ is the population's mean position in the $i^{\text{th}}_{m\,\text{th}}$ iteration and can be given as:

$$X(t_{\text{mean}}) = \frac{1}{pop} \sum_{i=1}^{pop} X_i \tag{16}$$

In the landing stage, the DO focuses on exploits. According to the first two phases, dandelion seeds randomly choose where to land. Hopefully, the algorithm will converge to a global optimum as the iterations proceed. Therefore, The optimal solution is the approximate location where the dandelion seeds will most likely survive. Explore agents borrow the best information from the current elite and use it locally for accurately converging on the global optimum. Ultimately, a global optimum is found as the population evolves. This behavior is shown in Eq. (17):

$$X(t+1) = X_{\text{elite}} + L \times a \times (X_{\text{elite}} - X(t) \times \gamma) \tag{17}$$

Here X_{elite} is the dandelion seed's optimal position in the i^{th}_m iteration, γ is a linear improving function between [0,2], and L is the Levy flight function, which is determined as:

$$L = c \times \frac{w \times \sigma}{|r|^{\frac{1}{\beta}}} \tag{18}$$

Where β shows a random number between [0,2], c shows a fixed constant of 0.01 , and r and w are the random number between $[0, 1]$. In addition, the flowchart of DO is demonstrated in Fig. 5.

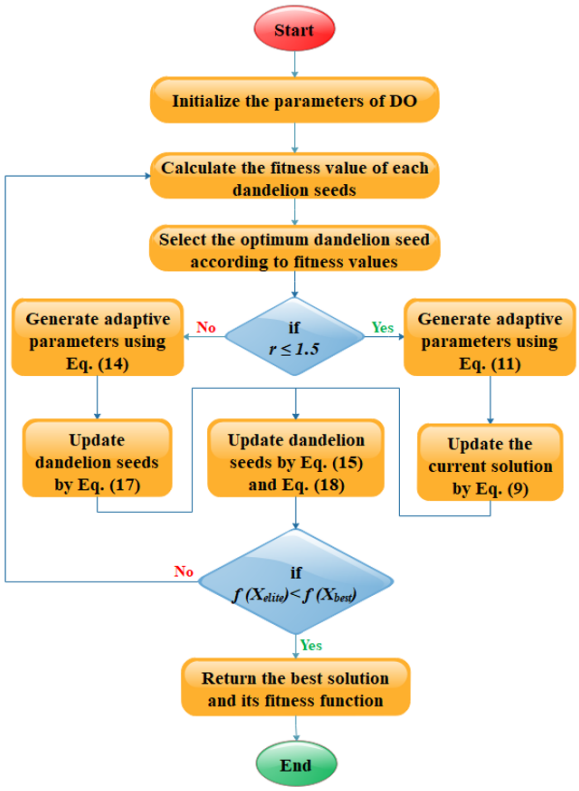

Fig. 5. The flowchart of the DO

2.5. Sooty Tern Optimization Algorithm (STOA)

Dhiman and Kaur proposed the STOA for engineering problems in 2019 [43]. The sooty tern utilized the flapping flight method while attacking in the air. This behavior can be formed to be assigned to the objective function to be optimized. Terns must meet three conditions during migration as in the following:

2.5.1. *Collision Avoidance*

E is employed to compute new search agent positions for avoiding collision avoidance between neighboring explore agents.

$$C = E \times P(x) \tag{19}$$

Here C shows the explore agent's position, which does not collide with other explore agents, P is the explore agent's current position, x is the current iteration, and E is the motion of explore agent in a represented explore agent.

$$E = V - \left(x \times \left(\frac{V}{\text{Max}_{iter}}\right)\right), \quad x = 0, 1, 2, \ldots, \text{Max}_{iter} \tag{20}$$

Here V shows the control variable for adapting the E, which is linearly reduced from V to 0.

2.5.2. *Connect towards best neighbors*

The explore agent converges on the best neighbors after collision avoidance.

$$O = M \times (F(x) - P(x)) \tag{21}$$

$$M = 0.5 \times r \tag{22}$$

Here O is the different position of explore agent towards the exploring agent of best fitness (F), M shows a random variable responsible for suitable searching, and r is a random number between the range of $[0, 1]$.

2.5.3. *Updated according to the best explore agents*

Ultimately, explore agents or terns can update their location relative to the best explore agent.

$$G = E + O \tag{23}$$

Here G is the gap between the exploring agent and explore agent of the best fitness. While moving, terns can change their speed and attack angle. They use their wings to grow taller. Therefore, the updated position of the exploring agent is calculated by Eq. (24):

$$P(x) = \left(G \times \left(x' + y' + z'\right)\right) \times F(x) \tag{24}$$

$$x' = R \times \sin(i) \tag{25}$$

$$y' = R \times \cos(i) \tag{26}$$

$$z' = R \times i \tag{27}$$

Here R is each turn's radius of the spiral, and i is the lies of the variable. The flowchart of STO is shown in Fig. 6.

2.6. Performance evaluation methods

Some metrics have been used to evaluate the hybrid models in the article to select the most suitable prediction model. Metrics include mean absolute percentage error (MAPE), coefficient correlation (R^2), mean absolute error (MAE), root mean square error (RMSE), and scatter index (SI).

- R^2 (Coefficient of Determination): Measures how well a regression model's predictions match the variability in actual data. It ranges from 0 to 1 , with higher values indicating a better fit.
- RMSE (Root Mean Square Error): Quantifies the average magnitude of differences between predicted and actual values. A lower RMSE signifies better predictive accuracy.
- MAPE (Mean Absolute Percentage Error): Calculates the average percentage difference between predicted and actual values. Helpful in understanding the proportional accuracy of predictions.
- MAE (Mean Absolute Error): Measures the average magnitude of differences between predicted and actual values. Lower values represent better prediction accuracy.
- Scatter Index (SI): Assesses the dispersion of data points around the line of best fit in a scatter plot. Provides insight into the consistency of data distribution.

Table 2. The hyperparameters for the developed models.

Layer	Hyperparameter	Model		
		MLAO	*MLST*	*MLDO*
Layer 1	*Neuron*	19	13	9
Layer 2	*Neuron1*	20	18	10
	Neuron2	7	18	17
Layer 3	*Neuron1*	8	15	5
	Neuron2	8	9	20
	Neuron3	6	20	4

The mathematical equations are as follows:

$$R^2 = \left(\frac{\sum_{i=1}^{N} (e_i - \bar{e})(o_i - \bar{o})}{\sqrt{\left[\sum_{i=1}^{N} (e_i - \bar{e})^2\right]\left[\sum_{i=1}^{N} (o_i - \bar{o})^2\right]}} \right)^2 \tag{28}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (o_i - e_i)^2} \tag{29}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |e_i - o_i| \tag{30}$$

$$MAPE = \frac{100}{N}\sum_{i}^{N} \frac{|o_i|}{|e_i|} \tag{31}$$

$$SI = \frac{RMSE}{\text{mean}\,(e_i)} \tag{32}$$

Where e_i and o_i determine the estimative and observed values, $\bar{e}$ and $\bar{o}$ define mean values of estimative and observed, respectively, N determines the sample number.

3. Results and discussion

The hyperparameters of MLP play a crucial role in determining its performance and behavior. Selecting the optimal hyperparameters involves experimentation and tuning. Techniques like grid search, random search, and more advanced methods like Bayesian optimization can be used to find the best combination of hyperparameters for a given problem. Table 2 indicates the hyperparameters of developed hybrid models.

This section has examined the current model using the metrics described in the previous section. When evaluating the model, the highest values of R^2 and the lowest values of MAPE, SI, MAE, and RMSE. Table 3 indicates

the model values obtained by the evaluator during the two training and testing phases, in which 70% and 30% of the samples are alternatively assigned to the training and testing section. Based on Table 1 , the highest R^2 value, equal to 99.39%, was obtained with MLAO3 $_{\text{Train,}}$ and the lowest was equal to 89.22%, obtained with MLDO1 $1_{\text{Test.}}$ In RMSE, the lowest was $\text{MLAO}_{\text{Train}} = 1.265\text{MPa}$, and the maximum was $\text{MLDO}_{\text{Train}} = 3.54\text{MPa}$. In MAPE, the best values came from MLST3 $3_{\text{rain}} = 1.42\text{MPa}$, while the worst values belonged to MLDO2 $2_{\text{Test}} = 3.165\text{MPa}$. For MAE, MLST3 $3_{\text{rain}} = 0.860\text{MPa}$ gave the best performance, and $\text{MLDO2}_{\text{Train}} = 2.36\text{MPa}$ was the worst. What is worth noting when evaluating the models is that all the models in the training part achieved the highest scores and performed poorly in the testing part. This determined that the model in the training part is not well trained. In the end, models should have the lowest of the best performance criteria in SI and other errors. The most satisfactory value was obtained MLAO3 $3_{\text{Test}} = 0.020\ (\text{MPa})$, and $\text{MLDO2}_{\text{Train}} = 0.050(\text{MPa})$ gives the lowest Generally, MLAO, MLST, and MLDO scored from strongest to weakest performers in each layer. Furthermore, it was indicated that combining AO and MLP can provide a highly accurate model acceptable for estimation. In addition, Table 4 compares the present model with other published models.

Fig. 7 indicates the scatterplots of measured and predicted CS in the two sections of testing and training. The related figure is defined according to two metrics, RMSE and R^2, which determined that they are on a straight line and have alternative variance and density. Moreover, the center line component based on the X = Y coordinates and the linear fit in two phases (training and testing) has been drawn in Fig. 7. The difference in the angles of these two lines shows whether the model performs well or poorly. In MLAOs, MLAO3 had higher values of R^2 and lower RMSE, resulting in less variability and a more negligible difference between angles and centerlines in linear fit. On the other hand, for MLDOs and MLSTs, MLDO3 and MLST3 models show a more satisfactory performance compared to different layers. In general, MLDO1 had the most desperation because of the lowest R^2 and the highest RMSE, whereas MLAO3 had the most density and was near to center line.

Fig. 8 compares the measured and predicted CS in the training and testing phases. The ideal situation is such that the predicted has similar behavior to the measured. In the first layer, comparing the models, it can be seen that MLAO1 recorded the least difference compared to the other two models, and MLDO1 had the most considerable difference in both phases. DO has the highest difference in the two different layers compared to other models, and AO

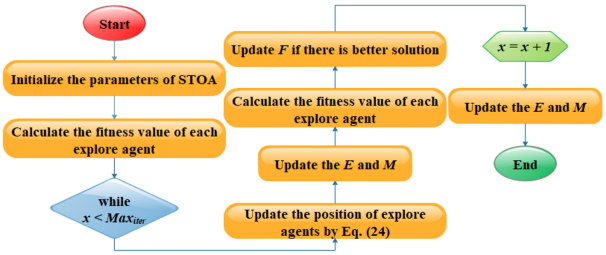

Fig. 6. The flowchart of STO

has been able to have a better combination in the second and third layers. All three models have performed their best in the third layer, and MLAO3 has achieved the most satisfactory results.

Fig. 9 determines the percentage of errors during the training and testing sections. As mentioned, 70% of the samples are related to the training and 30% to the testing phase. In the comparison of MLAOs in three layers, it can be seen that the lowest error percentage in the third layer is almost equal to 10% in the training phase, which by improving its performance, has brought the error to 5% in the testing phase. In MLDOs, in the training phase, the lowest value was obtained by MLDO2, which was equal to 19%, which is only 0.5% less than MLDO3, and in the test phase, MLDO3 obtained the lowest value equal to 7% compared to the other two models. In MLSTs, the lowest error, equivalent to 16% in the training phase, belonged to MLST1, which performed better in the test phase of MLST2, so it obtained the highest error equal to 8%. In general, it can be concluded that MLAO3 has obtained the most appropriate performance compared to other hybrid models.

Fig. 10 shows the histogram and distribution plot for developed models based on all data. In the first layer, it can be seen that MLDO1 had the highest dispersion and a flatter normal distribution. MLST1 and MLAO1 had a high density in the range close to zero percent, which had equal normal distribution, but MLST1 showed a slight right skew, which shows the superiority of MLAO1. MLDO2 had the weakest performance in the second layer, and MLAO2 had the most robust results with a sharper normal distribution and less error dispersion than the other two models. In the third layer, the normal distribution of MLDO3 is flatter and has more error dispersion, which compares the two models, MLST3 and MLAO3. Although the sharper normal distribution belongs to MLST3, the density of MLAO3 is higher in the range close to zero percent. In this case, it can be concluded that MLAO3 has performed more satisfactorily than other models.

Fig. 11 shows the Taylor diagram based on standard

Table 3. The result of presented models.

Hybrid model	Section	Statistic evaluator										
		R^2	Rank	RMSE	Rank	MAPE	Rank	MAE	Rank	SI	Rank	Total Rank
MLAO1	Train	97.86	4	2.5507	4	2.8598	5	1.7359	5	0.0394	4	6
	Test	91.64	7	2.3559	7	2.8063	6	1.8155	6	0.0378	7	
MLST1	Train	96.68	7	3.2307	7	3.3944	7	2.1540	7	0.0499	7	7
	Test	90.46	8	2.5660	8	2.9201	7	1.9318	8	0.0412	8	
MLDO1	Train	95.89	9	3.5469	8	3.6506	8	2.2716	8	0.0548	9	9
	Test	89.22	9	2.7108	9	3.2086	9	2.0034	9	0.0435	9	
MLAO2	Train	98.75	2	1.9828	2	1.4387	2	0.9302	2	0.0306	2	3
	Test	93.11	5	2.2625	4	1.8007	3	1.2113	3	0.0363	4	
MLST2	Train	97.78	6	2.6173	6	2.2375	4	1.3424	4	0.0404	6	5
	Test	91.71	6	2.3422	5	2.5723	5	1.6691	5	0.0376	5	
MLDO2	Train	96.52	8	3.2732	9	3.9228	9	2.3603	9	0.0505	8	8
	Test	94.15	3	2.3478	6	2.9169	8	1.8413	7	0.0377	6	
MLAO3	Train	99.39	1	1.370	1	1.6293	3	0.9576	3	0.0212	1	1
	Test	97.89	1	1.265	1	1.5904	2	1.0054	1	0.0203	1	
MLST3	Train	98.70	3	2.020	3	1.4281	1	0.8603	1	0.0312	3	2
	Test	94.95	2	1.8528	2	1.5031	1	1.0174	2	0.0297	2	
MLDO3	Train	97.86	5	2.5953	5	3.1098	6	1.8920	6	0.0401	5	4
	Test	93.68	4	2.0502	3	2.5332	4	1.6066	4	0.0329	3	

Table 4. Statistical properties of inputs and CS.

PAPER	MODEL	Compressive Strength	
		R^2	RMSE
[20]	MARS-PSO	0.9102	4.8756
[21]	MARS-PSOBBO	0.9422	4.3169
[44]	DMLP-I	0.9897	1.7634
[45]	AMLP-I	0.977	2.2271
Present work	SVPPM	0.984	9.50

deviation and correlation coefficient. As seen in the figure, the reference point specifies the measured value, and the closest predicted point indicates the proper performance of the model. In all three layers, the MLAO hybrid model has obtained the closest value, which is close to the MLAO of the MLST hybrid model. It had the weakest performance compared to the other two MLDO models, as it had the most significant difference in correlation coefficient and standard deviation.

4. Conclusion

This study introduced a novel approach by integrating machine learning (ML) for the prediction of Compressive Strength (CS) in High-Performance Concrete (HPC). The research focuses on Multi-layer Perceptron (MLP), a subset of artificial neural networks (ANNs), and delves into its application across three layers. Adding to this innovation, the study incorporates optimization algorithms to enhance accuracy and minimize errors. Specifically, three meta-heuristic algorithms-Dandelion Optimization (DO), Aquila Optimizer (AO), and Sooty Tern Optimization Algorithm (STOA) - are harnessed to fine-tune the model's performance. This amalgamation of optimizers with the corresponding model gives rise to a robust hybrid operating under MLDO, MLAO, and MLST. Evaluating the hybrid models is a critical facet of this research. The study employs several evaluators to rigorously assess the hybrid models, aiming to pinpoint the most suitable model. The evaluation process is grounded in well-defined metrics outlined in the previous sections.

The article's findings underscore the importance of achieving the highest values of R^2 (Coefficient of Determination) alongside the lowest values of MAPE (Mean Absolute Percentage Error), SI (Scatter Index), MAE (Mean Absolute Error), and RMSE (Root Mean Square Error). The presented results offer valuable insights into the performance of the models during both the training and testing phases. The diversity of outcomes, such as R^2 values ranging from 89.22% to 99.39%, underscores the complexity of the modeling process and the potential challenges. The distinctions between training and testing outcomes signal the need for further refinement, emphasizing the importance of a well-trained model for accurate predictions. A noteworthy observation is the model's affinity to achieve the lowest performance criteria errors, especially SI, which

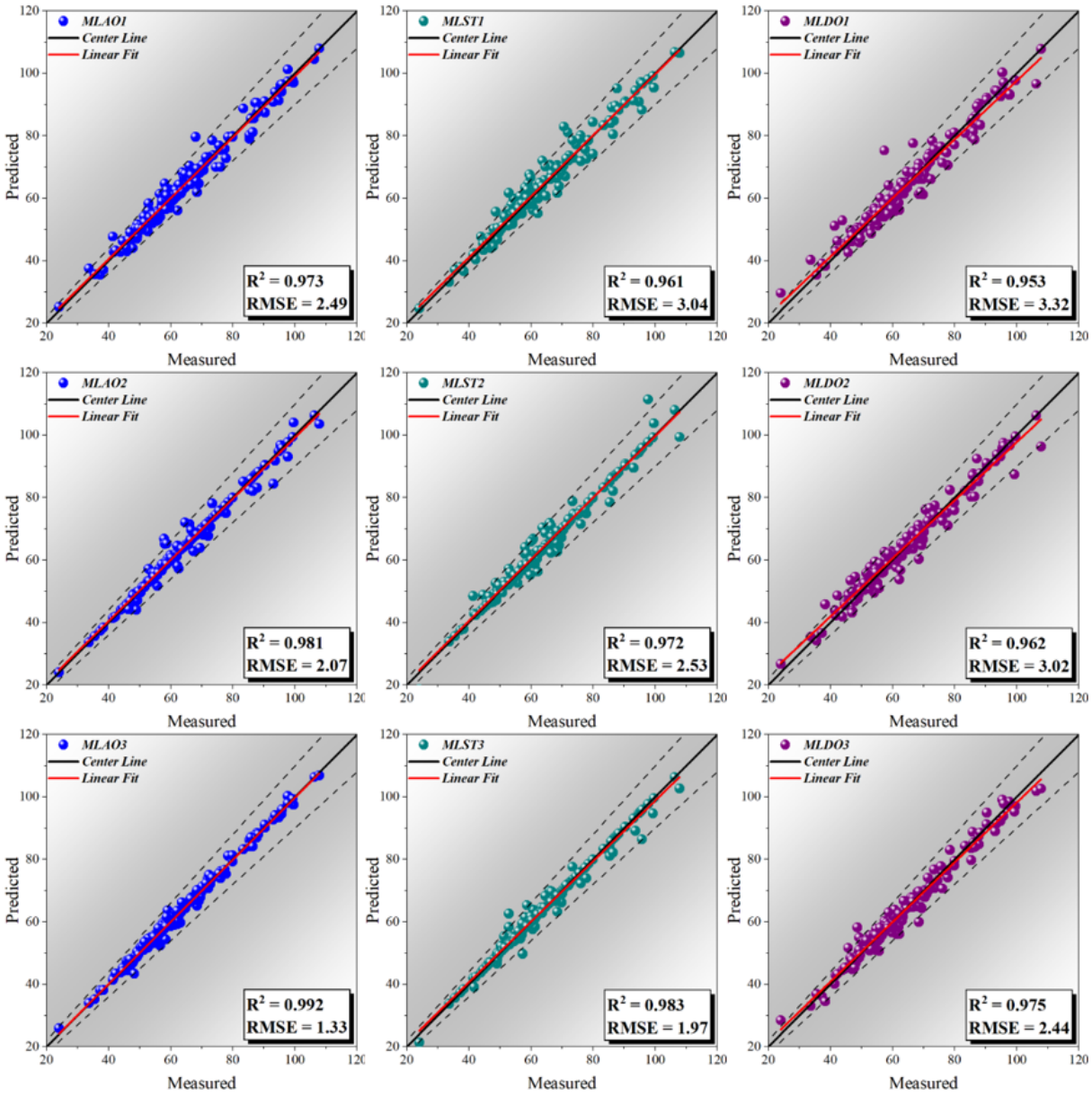

Fig. 7. The scatter plot for developed models

offers a comprehensive understanding of data distribution. The hierarchical performance ranking-MLAO, MLST, and MLDO from strongest to weakest-indicates the potential superiority of certain optimization strategies in specific contexts. The synergistic coupling of Aquila Optimizer (AO) and Multi-layer Perceptron (MLP) emerges as a powerful strategy, producing a highly accurate model wellsuited for prediction tasks.

The article's findings underscore the importance of achieving the highest values of R^2 (Coefficient of Determination) alongside the lowest values of MAPE (Mean Absolute Percentage Error), SI (Scatter Index), MAE (Mean Absolute Error), and RMSE (Root Mean Square Error). The presented results offer valuable insights into the performance of the models during both the training and testing phases. The diversity of outcomes, such as R^2 values ranging from 89.22% to 99.39%, underscores the complexity of the modeling process and the potential challenges. The distinctions between training and testing outcomes signal the need for further refinement, emphasizing the importance of a well-trained model for accurate predictions. A noteworthy observation is the model's affinity to achieve the lowest performance criteria errors, especially SI, which offers a comprehensive understanding of data distribution.

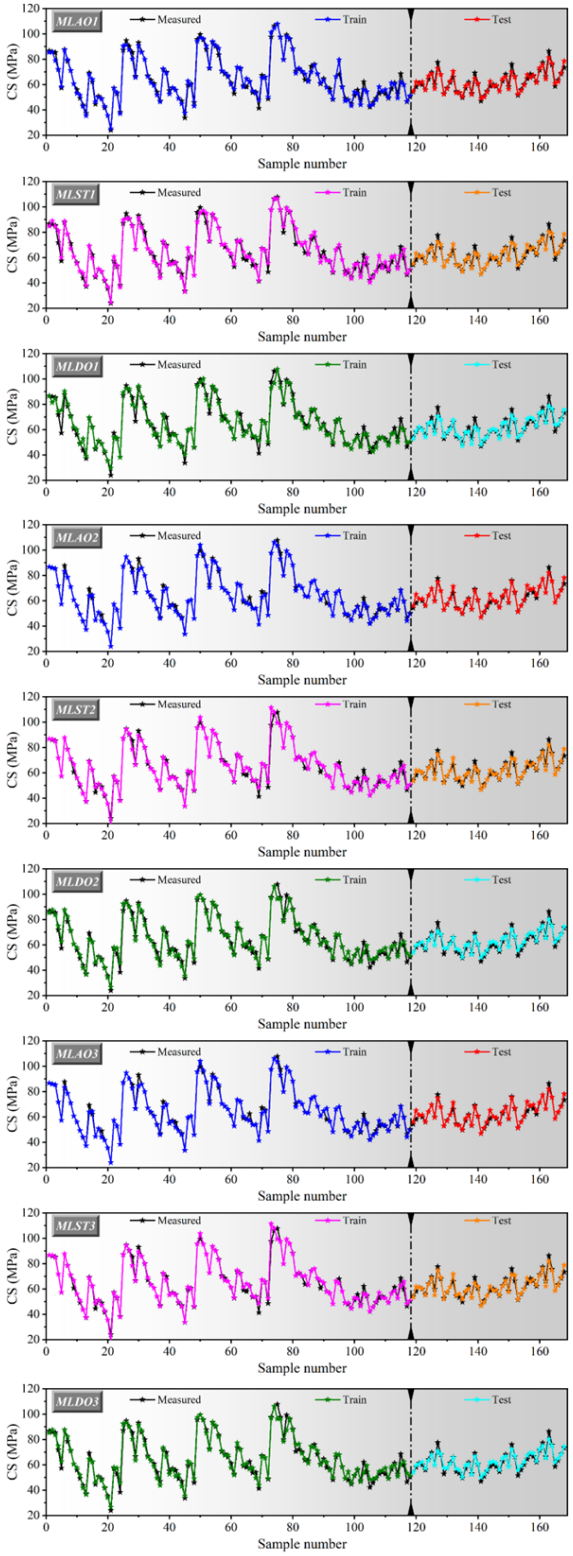

Fig. 8. The comparison between predicted and measured CS of developed models

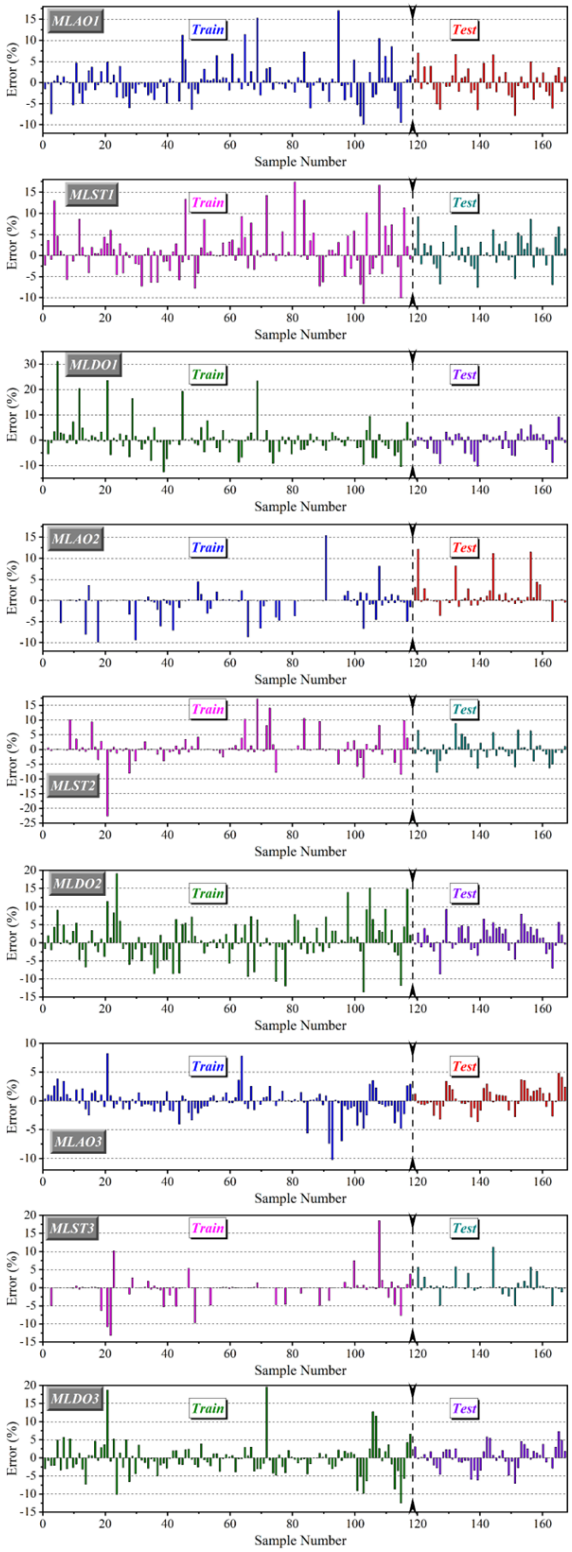

Fig. 9. The comparison between predicted and measured CS of developed models

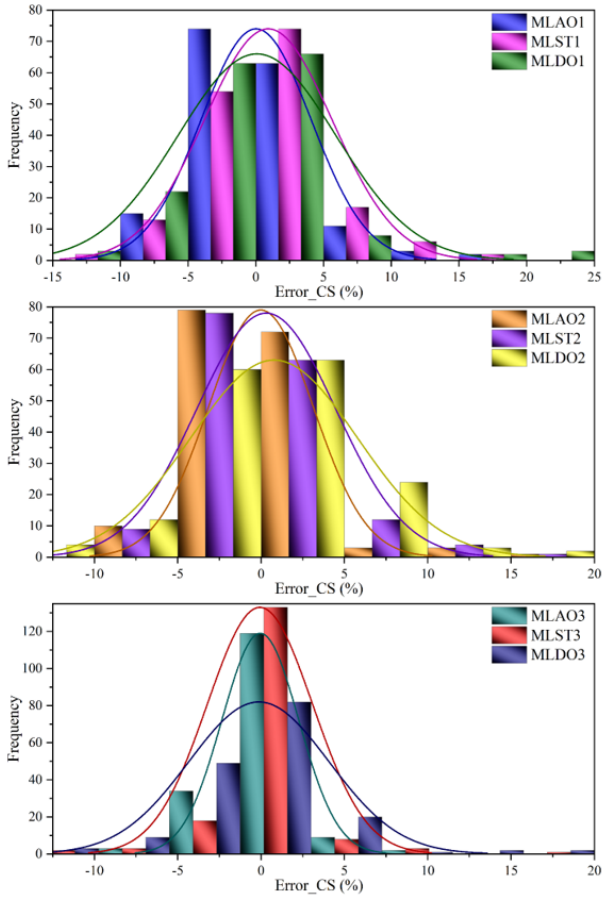

Fig. 10. The histogram and distribution plot for developed models

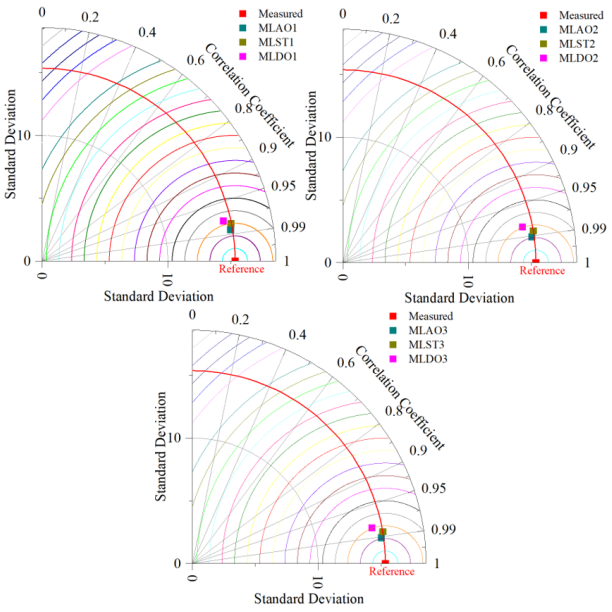

Fig. 11. The Taylor diagram for the developed models.

The hierarchical performance ranking-MLAO, MLST, and MLDO from strongest to weakest-indicates the potential superiority of certain optimization strategies in specific contexts. The synergistic coupling of Aquila Optimizer (AO) and Multi-layer Perceptron (MLP) emerges as a powerful strategy, producing a highly accurate model wellsuited for prediction tasks.

Funding

This work was supported by the Development and Application of an Intelligent Park Operation Management System based on BIM Technology (No. JC2021173)

Competing of interests: The authors declare no competing interests.

Authorship Contribution Statement. Xu Wu: Methodology, Software, Validation, Formal analysis.

Guifeng Yan: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.

Wei Zhang: Methodology, Writing-Original draft preparation, Software, Language review.

Yuping Bao: Formal analysis, Methodology, Software, Language review.

Appendix

References

[1] S. W. Forster, (1994) *"High-performance concrete: stretching the paradigm"* **Concrete International** ***16***(10): 33–34.

[2] V. M. Malhotra, (2006) *"Reducing CO2 emissions"* **Concrete international** ***28***(9): 42–45.

[3] P. K. Mehta, (2002) *"Greening of the concrete industry for sustainable development"* **Concrete international** ***24***(7): 23–28.

[4] E. G. Nawy. *Concrete construction engineering handbook*. CRC press, 1997.

[5] P. K. Mehta and P. J. Monteiro. *Concrete: microstructure, properties, and materials*. McGraw-Hill Education, 2014.

[6] J. Bai, B. Sabir, S. Wild, and J. Kinuthia, (2000) *"Strength development in concrete incorporating PFA and metakaolin"* **Magazine of concrete research** ***52***(3): 153–162.

[7] G. Menéndez, V. Bonavetti, and E. Irassar, (2003) *"Strength development of ternary blended cement with limestone filler and blast-furnace slag"* **Cement and Concrete Composites** ***25***(1): 61–67.

Table 5. The first layer weight of MLAO

	Hidden neuron 1	Hidden neuron 2	Hidden neuron 3	Hidden neuron 4	Hidden neuron 5	Hidden neuron 6	Hidden neuron 7	Hidden neuron 8
	W_{ini}	W_{iei}	W_{ii}	$W_{i,i}$	W_{ii}	$W_{i,i}$	W_{ii}	W_{ii}
Input 1	−0.764	0.867	−0.377	0.132	−0.902	−0.103	−0.344	−0.896
Input 2	−0.546	0.406	0.239	1.637	0.692	−0.241	1.144	0.056
Input 3	1.177	−0.265	0.689	−0.701	−0.191	−0.838	−0.075	0.639
Input 4	−0.901	−1.235	−0.899	−0.572	0.218	−0.924	−0.477	0.271
Input 5	−0.415	0.893	0.595	0.182	0.327	−0.005	−1.313	0.469
Input 6	−1.573	−0.004	−0.453	1.492	0.332	0.236	0.117	0.003
Input 7	−1.391	1.170	0.185	0.078	−0.401	−0.543	0.812	0.513
Input 8	−0.004	0.731	1.122	1.257	0.486	0.625	−0.292	−1.020

Table 6. The second layer weight of MLAO

	Hidden neuron 1	Hidden neuron 2	Hidden neuron 3	Hidden neuron 4	Hidden neuron 5	Hidden neuron 6	Hidden neuron 7	Hidden neuron 8
	w_{iii}	$W_{i,i}$	$W_{i,i}$	W_{iii}	$w_{i,i}$	$w_{i,i}$	$w_{i,i}$	$w_{i,i}$
Input 1	0.394	0.817	−0.339	1.297	0.486	0.078	−0.159	−0.365
Input 2	0.206	−0.982	−0.475	−0.170	−0.945	0.045	0.067	0.293
Input 3	−0.827	0.649	0.323	−0.363	−0.768	0.580	−1.319	0.420
Input 4	−0.383	0.298	−1.659	−0.204	0.205	0.054	−1.137	0.167
Input 5	−0.064	1.079	0.277	−0.851	−0.739	0.195	−1.116	−0.603
Input 6	0.994	−0.140	−0.297	0.045	−0.672	−0.692	−0.978	0.415

Table 7. The third layer weight of MLAO

	Hidden neuron 1	Hidden neuron 2	Hidden neuron 3	Hidden neuron 4	Hidden neuron 5	Hidden neuron 6	Hidden neuron 7	Hidden neuron 8
	$W_{i,i}$	$W_{i,i}$	$W_{i,i}$	W_{ii}	$W_{i,i}$	$W_{i,i}$	W_{ii}	$W_{i,i}$
Input 1	0.307	−0.355	−0.965	0.102	0.497	−0.342	0.307	−0.355

*Here i and j present the row and column, respectively, and w is the weight of the layer.

[8] W. Sun, Y. Zhang, S. Liu, and Y. Zhang, (2004) *"The influence of mineral admixtures on resistance to corrosion of steel bars in green high-performance concrete"* **Cement and Concrete Research *34***(10): 1781–1785.

[9] N. Farzadnia, A. A. A. Ali, and R. Demirboga, (2011) *"Incorporation of mineral admixtures in sustainable high performance concrete"* **International Journal of Sustainable Construction Engineering and Technology *2***(1):

[10] A. Ahmad, W. Ahmad, F. Aslam, and P. Joyklad, (2022) *"Compressive strength prediction of fly ash-based geopolymer concrete via advanced machine learning techniques"* **Case Studies in Construction Materials *16***: e00840.

[11] K. T. Nguyen, Q. D. Nguyen, T. A. Le, J. Shin, and K. Lee, (2020) *"Analyzing the compressive strength of green fly ash based geopolymer concrete using experiment and machine learning approaches"* **Construction and Building Materials *247***: 118581.

[12] S. S. Raza, M. T. Amir, M. Azab, B. Ali, M. Abdallah, M. H. El Ouni, and A. B. Elhag, (2022) *"Effect of micro-silica on the physical, tensile, and load-deflection characteristics of micro fiber-reinforced high-performance concrete (HPC)"* **Case Studies in Construction Materials *17***: e01380.

[13] W. Zhu, L. Huang, and Z. Zhang, (2022) *"Novel hybrid AOA and ALO optimized supervised machine learning approaches to predict the compressive strength of admixed concrete containing fly ash and micro-silica"* **Multiscale and Multidisciplinary Modeling, Experiments and Design *5***(4): 391–402.

[14] T. Chen, X. Gao, and M. Ren, (2018) *"Effects of autoclave curing and fly ash on mechanical properties of ultra-high performance concrete"* **Construction and Building Materials *158***: 864–872.

[15] P. Rossi, (2013) *"Influence of fibre geometry and matrix maturity on the mechanical performance of ultra high-performance cement-based composites"* **Cement and Concrete Composites *37***: 246–248.

[16] H. Yazıcı, H. Yiğiter, A. Ş. Karabulut, and B. Baradan, (2008) *"Utilization of fly ash and ground granulated blast furnace slag as an alternative silica source in reactive powder concrete"* **Fuel *87***(12): 2401–2407.

[17] L. Urbonas, D. Heinz, and T. Gerlicher, (2013) *"Ultra-high performance concrete mixes with reduced portland cement content"* **Journal of Sustainable Architecture and Civil Engineering *3***(4): 47–51.

[18] F. De Larrard and T. Sedran, (2002) *"Mixture-proportioning of high-performance concrete"* **Cement and concrete research *32***(11): 1699–1704.

[19] I.-C. Yeh, (1999) *"Design of high-performance concrete mixture using neural networks and nonlinear programming"* **Journal of Computing in Civil Engineering *13***(1): 36–42.

[20] H. Yin, S. Liu, S. Lu, W. Nie, and B. Jia, (2021) *"Prediction of the compressive and tensile strength of HPC concrete with fly ash and micro-silica using hybrid algorithms"* **Advances in Concrete Construction *12***(4): 339.

[21] L. Huang, W. Jiang, Y. Wang, Y. Zhu, and M. Afzal, (2022) *"Prediction of long-term compressive strength of concrete with admixtures using hybrid swarm-based algorithms"* **Smart Structures and Systems, An International Journal *29***(3): 433–444.

[22] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*. **4**. 4. Springer, 2006.

[23] J.-S. Jang and J.-J. Chen. "Neuro-fuzzy and soft computing for speaker recognition". In: *Proceedings of 6th international fuzzy systems conference*. **2**. IEEE. 1997, 663–668.

[24] Z. Waszczyszyn and M. Slonski, (2010) *"Some problems of artificial neural networks design"* **Advances of soft computing in engineering *512***: 237–316.

[25] Z. Waszczyszyn, M. Slonski, B. Miller, and G. Piatkowski. "Bayesian neural networks in the analysis of structural mechanics problems". In: *8th World congress on computational mechanics (WCCM8), Venice, Italy, EU*. 2008.

[26] A. H. Gandomi, A. H. Alavi, D. Mohammadzadeh Shadmehri, and M. Sahab, (2013) *"An empirical model for shear capacity of RC deep beams using genetic-simulated annealing"* **Archives of Civil and Mechanical Engineering *13***: 354–369.

[27] S. Afzal, B. M. Ziapour, A. Shokri, H. Shakibi, and B. Sobhani, (2023) *"Building energy consumption prediction using multilayer perceptron neural network-assisted models; comparison of different optimization algorithms"* **Energy *282***: 128446.

[28] F. Masoumi, S. Najjar-Ghabel, A. Safarzadeh, and B. Sadaghat, (2020) *"Automatic calibration of the groundwater simulation model with high parameter dimensionality using sequential uncertainty fitting approach"* **Water Supply *20***(8): 3487–3501.

[29] Z. Nurlan, (2022) *"A novel hybrid radial basis function method for predicting the fresh and hardened properties of self-compacting concrete"* **Advances in Engineering and Intelligence Systems *1***(01):

[30] H. Cheng, S. Kitchen, and G. Daniels, (2022) *"Novel hybrid radial based neural network model on predicting the compressive strength of long-term HPC concrete"* **Advances in Engineering and Intelligence Systems *1***(02):

[31] S. K. Babanajad, A. H. Gandomi, D. Mohammadzadeh, and A. H. Alavi, (2013) *"Numerical modeling of concrete strength under multiaxial confinement pressures using linear genetic programming"* **Automation in construction *36***: 136–144. DOI: 10.1016/j.autcon.2013.08.016.

[32] D.-C. Feng, Z.-T. Liu, X.-D. Wang, Y. Chen, J.-Q. Chang, D.-F. Wei, and Z.-M. Jiang, (2020) *"Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach"* **Construction and Building Materials *230***: 117000. DOI: 10.1016/j.conbuildmat.2019.117000.

[33] H. S. Ullah, R. A. Khushnood, F. Farooq, J. Ahmad, N. I. Vatin, and D. Y. Z. Ewais, (2022) *"Prediction of compressive strength of sustainable foam concrete using individual and ensemble machine learning approaches"* **Materials *15***(9): 3166. DOI: 10.3390/ma15093166.

[34] J. Kasperkiewicz, J. Racz, and A. Dubrawski, (1995) *"HPC strength prediction using artificial neural network"* **Journal of Computing in Civil Engineering *9***(4): 279–284. DOI: 10.1061/(ASCE)0887-3801(1995)9:4(279).

[35] M. Jakubek and Z. Waszczyszyn. "Neural analysis of concrete fatigue durability by the neuro-fuzzy FWNN". In: *International Conference on Artificial Intelligence and Soft Computing*. Springer. 2004, 1075–1080.

[36] H. Naderpour, A. H. Rafiean, and P. Fakharian, (2018) *"Compressive strength prediction of environmentally friendly concrete using artificial neural networks"* **Journal of Building Engineering *16***: 213–219.

[37] H. Mashhadban, S. S. Kutanaei, and M. A. Sayarinejad, (2016) *"Prediction and modeling of mechanical properties in fiber reinforced self-compacting concrete using particle swarm optimization algorithm and artificial neural network"* **Construction and Building Materials *119***: 277–287. DOI: 10.1016/j.conbuildmat.2016.05.034.

[38] S. Jueyendah, M. Lezgy-Nazargah, H. Eskandari-Naddaf, and S. Emamian, (2021) *"Predicting the mechanical properties of cement mortar using the support vector machine approach"* **Construction and Building Materials *291***: 123396.

[39] L. Lam, Y. Wong, and C.-S. Poon, (1998) *"Effect of fly ash and silica fume on compressive and fracture behaviors of concrete"* **Cement and Concrete research *28***(2): 271–283.

[40] B. T. Pham, D. T. Bui, I. Prakash, and M. Dholakia, (2017) *"Hybrid integration of Multilayer Perceptron Neural Networks and machine learning ensembles for landslide susceptibility assessment at Himalayan area (India) using GIS"* **Catena *149***: 52–63.

[41] N. Karballaeezadeh, F. Zaremotekhases, S. Shamshirband, A. Mosavi, N. Nabipour, P. Csiba, and A. R. Várkonyi-Kóczy, (2020) *"Intelligent road inspection with advanced machine learning; hybrid prediction models for smart mobility and transportation maintenance systems"* **Energies *13***(7): 1718.

[42] S. Zhao, T. Zhang, S. Ma, and M. Chen, (2022) *"Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications"* **Engineering Applications of Artificial Intelligence *114***: 105075.

[43] G. Dhiman and A. Kaur, (2019) *"STOA: a bio-inspired based optimization algorithm for industrial engineering problems"* **Engineering Applications of Artificial Intelligence *82***: 148–174.