Smart Structures and Systems, *Vol. 33, No. 1 (2024) 65-91*
https://doi.org/10.12989/sss.2024.33.1.065

Slime mold and four other nature-inspired optimization algorithms in analyzing the concrete compressive strength

Yinghao Zhao [1a], Hossein Moayedi[*2,3], Loke Kok Foong [2,3c] and Quynh T. Thi [2,3d]

[1] *School of Civil Engineering and Engineering Management, Guangzhou Maritime University, Guangzhou, 510725, China*
[2] *Institute of Research and Development, Duy Tan University, Da Nang, Vietnam*
[3] *School of Engineering & Technology, Duy Tan University, Da Nang, Vietnam*

(*Received September 9, 2020, Revised November 25, 2023, Accepted December 28, 2023*)

Abstract. The use of five optimization techniques for the prediction of a strength-based concrete mixture's best-fit model is examined in this work. Five optimization techniques are utilized for this purpose: Slime Mold Algorithm (SMA), Black Hole Algorithm (BHA), Multi-Verse Optimizer (MVO), Vortex Search (VS), and Whale Optimization Algorithm (WOA). MATLAB employs a hybrid learning strategy to train an artificial neural network that combines least square estimation with back-propagation. Thus, 72 samples are utilized as training datasets and 31 as testing datasets, totaling 103. The multi-layer perceptron (MLP) is used to analyze all data, and results are verified by comparison. For training datasets in the best-fit models of SMA-MLP, BHA-MLP, MVO-MLP, VS-MLP, and WOA-MLP, the statistical indices of coefficient of determination (R^2) in training phase are 0.9603, 0.9679, 0.9827, 0.9841 and 0.9770, and in testing phase are 0.9567, 0.9552, 0.9594, 0.9888 and 0.9695 respectively. In addition, the best-fit structures for training for SMA, BHA, MVO, VS, and WOA (all combined with multilayer perceptron, MLP) are achieved when the term population size was modified to 450, 500, 250, 150, and 500, respectively. Among all the suggested options, VS could offer a stronger prediction network for training MLP.

Keywords: fly ash; high strength concrete; neural-evolutionary; optimization

1. Introduction

Concrete is a necessary component utilized in nearly all civil engineering-related activities. It is a widely utilized artificial concoction of fundamental elements like gravel, cement, water, and other minerals (Huang *et al*. 2021a, b, Han *et al*. 2023). Compressive strength, in addition to suitable concrete workability, is a determining element that helps workers mold concrete into any desired shape (Moayedi *et al*. 2019). Compared to ordinary concrete mixtures, high-strength concrete is intended to have noticeably higher compressive strength (Chen *et al*. 2023; Han *et al*. 2023). High-strength concrete can have compressive values ranging from 40 MPa to 100 MPa, whereas normal concrete generally has compressive strengths of between 20 and 40 MPa. High-strength concrete is stronger because of several characteristics, such as the use of certain cement blends, a lower water-to-cement ratio, and careful mix percentage management (Zhu *et al*. 2018, Wang *et al*. 2022). These elements reduce voids in the concrete matrix and optimize cement particle packing, producing a denser and stronger material (Jiang *et al*. 2022, Shi *et al*. 2022). Different materials and techniques may be used by different varieties of high-strength concrete to increase its strength (Huang *et al*. 2021b, Wang *et al*. 2022). Certain popular additions that can enhance the overall durability and performance of the concrete include ground granulated blast furnace slag, fly ash, and silica fume (Cao *et al*. 2020, Zhou *et al*. 2021, Guo *et al*. 2022, Moayedi *et al*. 2023). Using high-strength concrete benefits from increased weight-bearing capacity, less material needed, and better resistance to impact, abrasion, and adverse environmental conditions (Zhou *et al*. 2023b). High-rise buildings, bridges, dams, tunnels, and other constructions needing greater structural strength are frequently built using it (Liu *et al*. 2021, Zhang *et al*. 2021, Huang *et al*. 2022c). Li *et al*. (2022) tackled the time-lag effect in examining the soil structure, a significant concern in the building industry. Furthermore, Zhou *et al*. (2021) developed a low-alkali, high-strength geopolymer based on the BH-1 lunar soil simulator. Additionally, Liu *et al*. (2023) used finite element modeling to investigate the formation of cracks and damage in tunnel lining under seismic loading.

Numerous methods have been developed for evaluating structural materials like concrete (Huang *et al*. 2022c, Singh *et al*. 2023). Within this context, intelligent models—like Artificial Neural Networks (ANNs)—have shown promise among the several methods for assessing and forecasting particular attributes (Sun *et al*. 2022, Li *et al*. 2023a, Ren *et al*. 2023). Artificial Neural Networks (ANNs) are effective forecasting tools that mimic organic neural networks (McCulloch and Pitts 1943, Moayedi and Jahed Armaghani

*Corresponding author, Ph.D.,
E-mail: hosseinmoayedi@duytan.edu.vn
[a] Ph.D., E-mail: zhaoyinghao@gzmtu.edu.cn
[b] Ph.D., E-mail: hosseinmoayedi@duytan.edu.vn
[c] Ph.D., E-mail: lokekokfoong@duytan.edu.vn
[d] Ph.D., E-mail: quynhtrng@duytan.edu.vn

ISSN: 1738-1584 (Print), 1738-1991 (Online)

2018, Moayedi *et al*. 2021a). Put another way, ANNs are a class of machine-learning models that draw inspiration from the composition and capabilities of the human brain (Li *et al*. 2023b). They are made up of layered networks of linked nodes, sometimes referred to as artificial neurons or units. Every neuron receives inputs, transforms them mathematically, and outputs something that is then sent to the layer below. ANNs learn from data by modifying the weights and biases associated with the connections between neurons to maximize their performance on a particular job. Because of its primary advantage over conventional predictive models like linear regression—namely, its capacity to examine the non-linear relationship between independent and dependent variables, or, in the case of this study, the strength and efficient factors—many researchers have begun to apply ANNs in a variety of engineering-related fields (Moayedi *et al*. 2011, Moayedi and Hayati 2018a, b). In several domains, and including PV-battery storage systems (Huang and Wang 2020), Landslide Susceptibility (Bui *et al*. 2019, Mehrabi and Moayedi 2021), power system efficiency, pile uplift load capacity (Nazir *et al*. 2015), ANN-based methodologies have demonstrated promising results (Huang and Wang 2022). ANNs may be used to estimate the strength of concrete as one of its practical applications (Zhou *et al*. 2023a). Predicting concrete strength is a crucial part of the construction industry since it allows engineers and designers to ensure that concrete constructions are durable and structurally sound.

More computational intelligent base research studies can be mentioned here such as employing deep-reinforcement learning in the safety evaluation of traffic systems (Dai 2021, 2022) and metro traffic improvement plan (Dai 2023), bearing capacity of shallow footing (Wenjun *et al*. 2023), model transient vibration responses of GPLs reinforced doubly curved concrete panel under instantaneous heating (Zhao *et al*. 2023), Application in robot selection (Zhang *et al*. 2024), Single Phase PM Brushless DC Motor using deep neural network (Zhang *et al*. 2023), and enhancing robot path planning (Zhang and Zhang 2023).

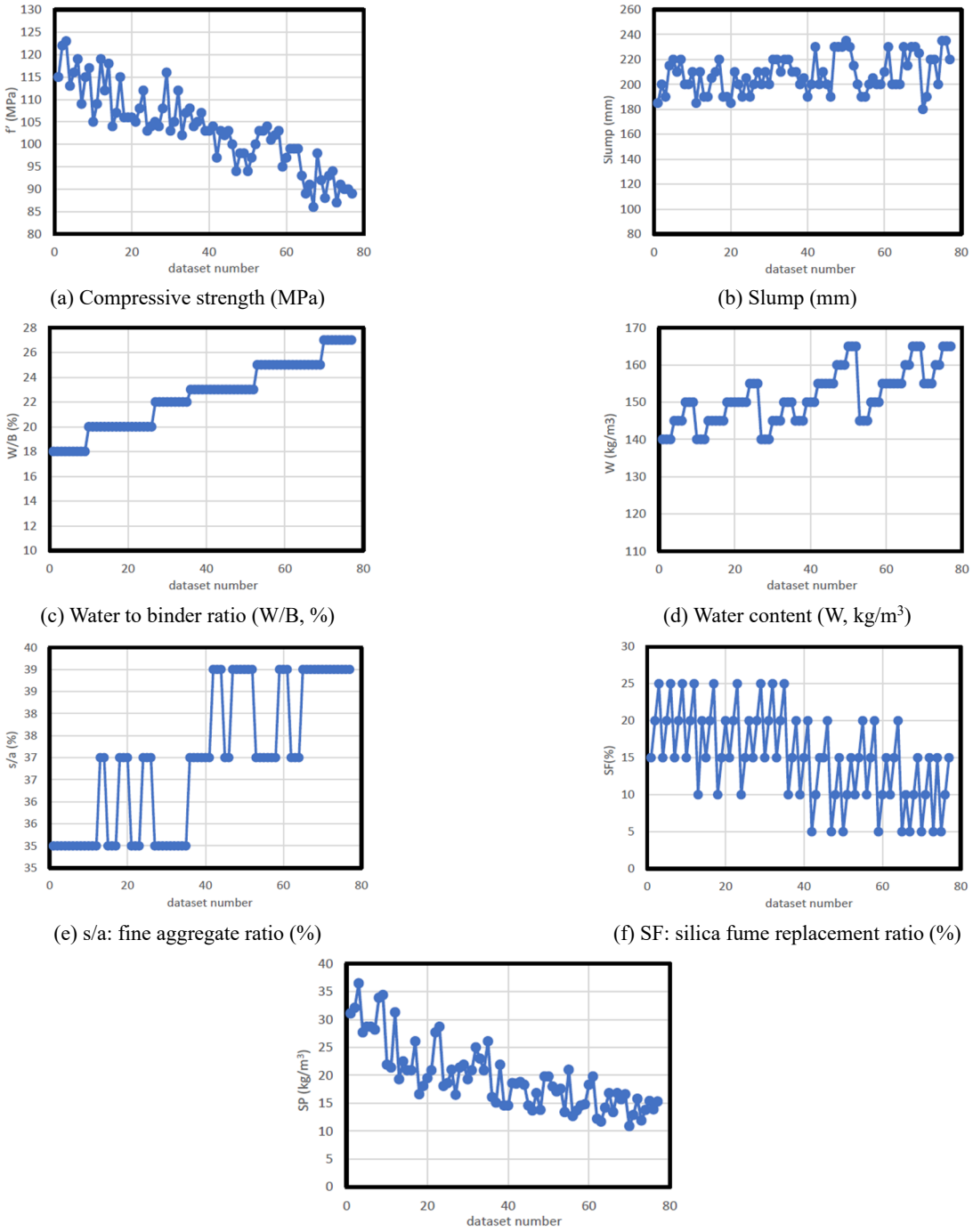
There are several benefits to predicting concrete strength utilizing artificial intelligence-based predictive networks. They can represent interactions and nonlinear connections between different input components, which can be difficult to represent using conventional mathematical equations. They are ideal for managing intricate and dynamic systems like concrete strength as they can also adjust to various data patterns and generalize effectively to data that has not yet been seen. Cemalgil *et al*. (2022) looked into a model that predicted the durability characteristics of concrete. They used a hybridized generalized extreme learning machine and added steel fiber and silica fume to the concrete. The layers of balanced laminated composite plates were optimized using the SMA, BHA, MVO, VS and WOA. These algorithms represent innovative optimization approaches, drawing inspiration from diverse natural phenomena. Each method introduces unique strategies to navigate complex solution spaces, making them suitable for optimization problems in various fields, including engineering, finance, and machine learning. Researchers and practitioners often explore these algorithms to find efficient and effective solutions to real-world challenges. The robustness of the approach was demonstrated by the results and the number of function evaluations (Kaveh and Bakhshpoori 2013). Cemalgil *et al*. (2023) investigated using a standalone and hybrid generalized extreme learning machine to estimate the engineering parameters of changed concrete. This paper aims to suggest a novel optimization framework for concrete strength. In the current study, five optimization techniques were used to forecast the strength of concrete (including fly ash and superplasticizer) and select the best-fit model. He *et al*. (2023) used a machine learning technique to study the problem of controlled synthesis of carbon dots-based corrosion inhibitors. Huang *et al*. 2022a, Hu *et al*. 2023, Shi *et al*. 2023) are a few other studies that used intelligence methodologies to achieve positive outcomes in civil business.

2. Established database

The representativeness and quality of the training data determine the predictive networks' accuracy and dependability. Accurate prediction is also contingent upon the network design and the suitable selection of input parameters. Therefore, the key to applying machine learning theories for concrete strength prediction efficiently is rigorous model construction and training, together with appropriate data collection and preprocessing. Accordingly, the database utilized for this study included 77 admixtures that were taken from Lim *et al*. (2004). Compaction strength is determined by several factors, including the water-to-binder ratio (W/B, %), water content (W, kg/m^3), fine aggregate ratio (s/a, %), fly ash replacement ratio (FA, %), silica fume replacement ratio (SF, %), air-entraining agent content (AE, %), superplasticizer content (kg/m^3), and the content of the air-entraining agent (AE, %). The quantity of input layer parameters is shown in Fig. 1.

3. Methodology

Artificial neural networks (ANNs) may be trained using historical data, including details on the composition of concrete mixtures and the accompanying strength test results, to forecast the strength of concrete. The water-cement ratio, aggregate characteristics, curing conditions, type and quantity of cement, and other variables that determine concrete strength are all complicated interactions that the artificial neural network (ANN) learns to understand. Based on fresh input data, the ANN may be trained to estimate the strength of concrete. By providing the ANN with information on the composition and curing conditions of a concrete mixture, the network can determine the strength of the concrete. This prediction can help engineers ensure that the required strength requirements are satisfied, optimize mix designs, and evaluate the quality of concrete before construction. Metaheuristic methods, on the other hand, are optimization algorithms that draw inspiration from natural events or approaches to addressing



(a) Compressive strength (MPa)

(b) Slump (mm)

(c) Water to binder ratio (W/B, %)

(d) Water content (W, kg/m^3)

(e) s/a: fine aggregate ratio (%)

(f) SF: silica fume replacement ratio (%)

(g) SP: superplasticizer content (kg/m3)

Fig. 1 Graphical description of the input and output layers

problems. In situations when standard approaches may be ineffective or impracticable, they address complicated optimization issues. Metaheuristics offers a broad framework for examining and utilizing a problem's search space to identify good or nearly perfect solutions.

The overall operation of metaheuristic methods is summarized as follows:

- Initialization: To begin, the algorithm generates a starting population or collection of solutions. The following is a collection of possible fixes for the issue. Evaluation: An objective function or fitness function is used to assess each solution in the population. The objective function gauges the performance or quality of a solution.
- Iterative Improvement: This method is the foundation of the metaheuristic algorithm. It entails coming up with fresh ideas and honing them again.
- Solution Generation: Depending on the particular metaheuristic algorithm, new solutions are produced by utilizing a variety of operators or approaches, such as crossover, perturbation, or mutation. These operators imitate swarm behavior, local search, or

evolution as natural processes.

- Solution Evaluation: The objective function assesses the newly created solutions to establish their quality or fitness.
- Solution Selection: The algorithm chooses a portion of the population's solutions for the following iteration. To prefer superior solutions, the selection process might be based on various factors, including fitness rating or probabilistic selection.
- Termination Criteria: Until a certain termination condition is satisfied, the algorithm keeps up the iterative improvement process. This requirement might be a time constraint, a maximum number of iterations, or a desirable level of solution quality.
- Output: The best solution discovered throughout the iterations is chosen as the final answer or as a close approximation to the ideal solution of the issue after the algorithm concludes.

It's crucial to remember that many metaheuristic approaches have unique qualities and differences in the particular procedures. Popular metaheuristic algorithms include Tabu Search (TS), Simulated Annealing (SA), Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Ant Colony Optimization (ACO). Every algorithm has a unique method for scouring the search space and striking an effective balance between exploitation and exploration to locate good answers. In general, metaheuristic approaches, which draw inspiration from nature and problem-solving methodologies, offer a versatile and resilient method of tackling optimization issues. They are very helpful when dealing with complicated, multi-dimensional, or combinatorial optimization issues, where standard techniques would find it difficult to discover appropriate answers. The five most recently developed metaheuristic optimization strategies are integrated with multilayer perceptron (MLP) because metaheuristic algorithms function appropriately while optimizing neural network models. Here, a basic discussion of MLP and combination approaches is provided.

3.1 Multi-layer perceptron

An artificial neural network (ANN) comprised of several layers of linked nodes, sometimes called artificial neurons or perceptrons, is called a multilayer perceptron (MLP). Machine learning applications such as pattern recognition, regression analysis, and classification tasks extensively use MLPs.

This is a detailed description of how an MLP operates:

Input Layer: The input layer of the MLP is where the input data is first received. Each neuron represents a characteristic or aspect of the input in the input layer.

Hidden Layers: One or more hidden layers come behind the input layer. Several neurons make up each hidden layer, which uses the input data to perform nonlinear transformations. The MLP can learn intricate representations and extract significant information from the input using these hidden layers.

Weights and Biases: Every neuronal connection between neighboring layers has a weight attached. Consider the weights as the parameters that the machine learning program (MLP) learns during training; they establish the strength of the link. A bias term is present in every neuron inside a hidden layer to further enhance the model's adaptability.

Activation Function: The outcome is subjected to an activation function following the weighted sum of the inputs and biases. The MLP can recognize and depict nonlinear connections in the data thanks to the activation function, which adds nonlinearity to the model. Examples of frequently utilized activation are the sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU) functions.

Forward Propagation: Starting at the input layer, the input data is sent layer by layer through the network. Every neuron in a layer takes in information from every other layer's neuron, applies bias and weighted sum, and then sends the outcome through the activation function. Up until the output layer is reached, this process keeps on.

Output Layer: The neurons of the output layer, the last layer of the MLP, generate the outputs or predictions of the network. The output layer's neuron count varies according to the particular job. One neuron would normally represent each class in a binary classification issue; however, in a regression job, one neuron may be used to create a continuous result.

Loss Function: A loss function is used to quantify the difference between the expected and actual outputs of the MLP and compare it to the desired output. The job at hand determines the loss function to be used. For instance, cross-entropy loss is frequently employed in classification tasks, but in regression assignments, mean squared error is frequently utilized.

Backpropagation and Training: The MLP is trained by an optimization method that repeatedly modifies the weights and biases. Backpropagation is the most widely used method; it computes the gradient of the loss function concerning the weights and biases. The MLP can thus minimize the loss and continuously improve its predictions by updating the weights and biases in the opposite direction of the gradient.

Iterative Training: For a predetermined number of iterations or until a convergence requirement is satisfied, steps 5 through 8 are repeated. The MLP gains the ability to adjust its weights and biases to minimize loss and retrieve pertinent features from the input data during training.

Inference: The MLP can forecast fresh, unseen data once trained. The output layer generates the predictions based on the learned weights and biases after the input is passed through the network.

MLPs may be tailored to various tasks and achieve high performance in many real-world applications by modifying the architecture (number of hidden layers and neurons) and training parameters (learning rate, batch size, etc.), as a similar approach illustrated in Table 1. It comprises layers of interconnected neurons and nodes that function as a single entity. During training, the ANN modifies its structure in response to external or internal data passing through the network. Although it contains several layers, the input and output layers comprise at least two. Hidden layers

Table 1 Optimization of the used MLP in the current study

Number of neurons	Network results			Scoring			Total score	Rank
	MSE_{total}	$RMSE_{train}$	$RMSE_{test}$	MSE_{total}	$RMSE_{train}$	$RMSE_{test}$		
1	0.174	0.209	0.185	2	5	2	9	8
2	0.151	0.245	0.184	3	1	3	7	9
3	0.115	0.179	0.137	9	7	9	25	2
4	0.116	0.217	0.153	8	3	6	17	5
5	0.080	0.146	0.104	10	10	10	30	1
6	0.124	0.166	0.138	7	9	8	24	3
7	0.131	0.171	0.144	5	8	7	20	4
8	0.129	0.223	0.163	6	2	4	12	7
9	0.140	0.187	0.156	4	6	5	15	6
10	0.177	0.211	0.188	1	4	1	6	10

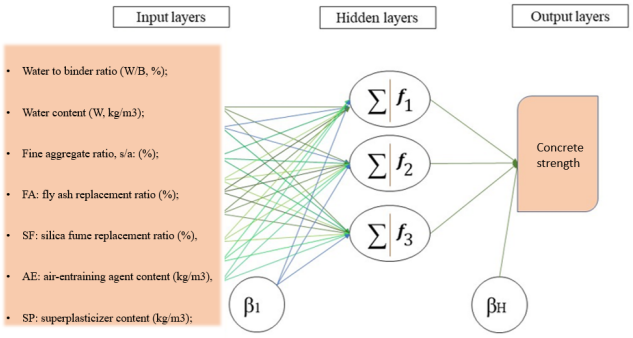


Fig. 2 Structure of the used multi-layer perceptron (MLP) in this study

comprise the neurons that come before the output layer and after the input layer. The output layer of the network is established by the most recent neurons after signals are received through the input layer and modified on the various layers. An activation function is assigned to a neuron, defining the node's output in response to an input. An MLP is a feed-forward network with at least one hidden layer (Gardner and Dorling 1998, Chandwani *et al*. 2015, Greco *et al*. 2021). Multi-layered perceptrons, or MLPs, have a structure corresponding to supervised learning data. The nodes in the final layer represent every class label included in the training dataset. It validates the relationship between the data and is consistent with the data entered (Chatterjee *et al*. 2022). Every layer is connected to both the levels above and below. Fig. 2 depicts the MLP structure that was employed in this investigation. The first MLP optimization procedure demonstrates that an MLP may produce the best-fit MLP basis prediction network in the given case with one hidden layer and five neurons. This implies that we may start the MLP network with this topology for the next hybridization before launching the hybrid algorithms.

3.2 Development of hybrid metaheuristic algorithms (training MLP)

The general technique employed to achieve the study's goal is shown in Fig. 3. Given that metaheuristic algorithms are effective in optimizing traditional predictive models, five recently developed algorithms are employed for MLP. As alternate search strategies to find the best answer to a given problem, the Slime Mold Algorithm (SMA), Black

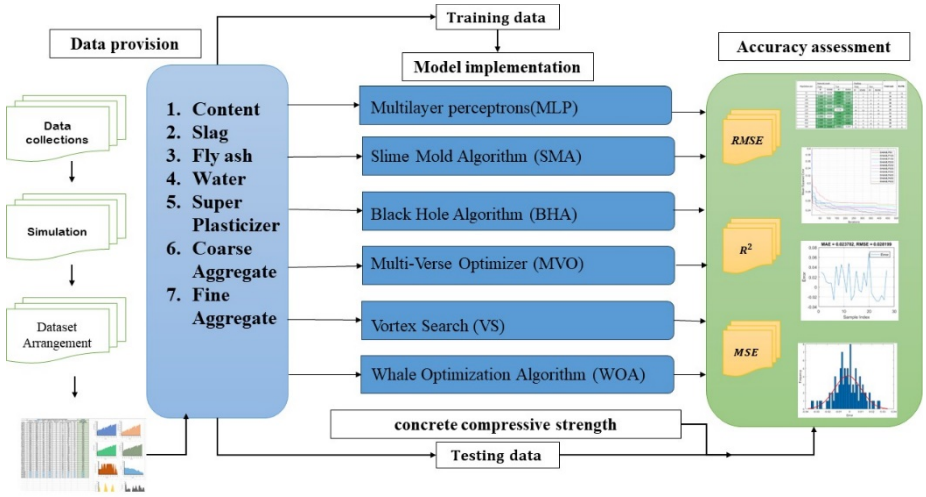


Fig. 3 The graphical methodology of the present study

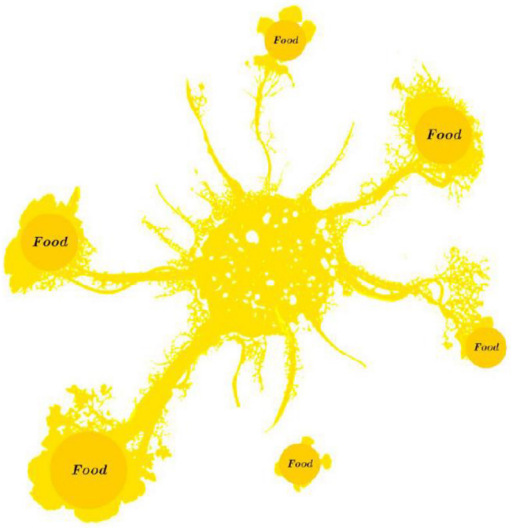


Fig. 4 The foraging morphology of the SM (after Li *et al*. (2020))

Hole Algorithm (BHA), Multi-Verse Optimizer (MVO), Vortex Search (VS), and Whale Optimization Algorithm (WOA) are discussed. Given a typical multilayer parity issue, the methods aim to determine the best weights and biases for the network in terms of a cost function. The mean square error, R^2, mean absolute error, and standard deviations are the loss functions used to evaluate the accuracy of predictions.

3.2.1 Slime Mould Algorithm (SMA)

The Slime Mould Algorithm (SMA) was inspired by the natural behavior of slime molds, particularly Physarum polycephalum, a simple yet fascinating organism. The development of the algorithm draws upon observations of how slime molds create efficient networks to find optimal paths for nutrient absorption. The algorithm was proposed to solve optimization problems by mimicking slime molds' adaptive and decentralized behavior (Li *et al*. 2020). The dynamic nutritional behavior of the SM known as Plasmodium serves as the primary source of inspiration for the SMA algorithm. To find food, surround it, and secrete enzymes to break it down, an organic substance goes through three stages in this stage. The foraging morphology that uses several food items at once to create a linked venous network is seen in Fig. 4.

As a unique optimization technique, Li *et al*. (2020) created the SMA motivated by the mentioned foraging behavior. The SM's goal is to find the best route to the highest concentration of nutrients. Latty and Beekman (2010). While the SM views this as the most promising food source, it must consider two critical aspects of foraging: danger and speed. Another issue for the SM is deciding when to leave the searched area and move on to a new one. The algorithm makes use of empirical or heuristic rules to determine it. However, as previously said, the algorithm can concurrently take advantage of several sources (Beekman and Latty 2015). An adaptive search approach is used to choose the optimum food source when several options with varying attributes are available.

The SMA is composed of four main mathematical steps, which are explained as follows:

a) Approaching food

Regarding the odor in the air, the SM approaches food based on the below equation:

$$\overrightarrow{X(t+1)} = \begin{cases} \overrightarrow{X_b(t)} + \overrightarrow{vb} \cdot \left(\overrightarrow{W} \cdot \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}\right), r < p \\ \overrightarrow{vc} \cdot \overrightarrow{X(t)}, r \geq p \end{cases} \quad (1)$$

where t signifies the current iteration, $\overrightarrow{vc}$ follows a linear decrease from 1 to 0, $\overrightarrow{vb}$ is a parameter ranging in $[-a, a]$ where $a = arctanh\left(-\left(\frac{t}{\max_t}\right)+1\right)$, and $\overrightarrow{W}$ stands for the weight of the SM. Also, the locations belonging to the SM, the individual with the largest odor concentration so far, and two randomly selected individuals are represented by $\vec{X}$, $\overrightarrow{X_b}$, $\overrightarrow{X_A}$ and $\overrightarrow{X_B}$, respectively.

Moreover, given (i) as the fitness of $\vec{X}$, and DF as the best fitness obtained ever, the term p can be formulated as follows

$$p = \tanh|S(i) - DF| \quad (2)$$

Eq. (3) gives the $\overrightarrow{W}$.

$$\overrightarrow{W(SmellIndex(i))} = \begin{cases} 1 + r \cdot log\left(\frac{bF - S(i)}{bF - wF} + 1\right), condition \\ 1 - r \cdot log\left(\frac{bF - S(i)}{bF - wF} + 1\right), \quad others \end{cases} \quad (3)$$

$$SmellIndex = sort(S) \quad (4)$$

where $condition$ refers to ranking the first half of the population concerning the $S(i)$. The term r is a random number in [0,1], bF and wF stand for the optimal and worst fitnesses grasped in the current repetitions, respectively, $SmellIndex$ gives the ascending sequence of sorted $S(i)$s.

b) Wrapping the food

This step simulates the contraction of the venous tissue structure of the SM during the search. In this sense, the concentration level of the food that the vein contacts is directly correlated with three parameters: the strength of the waves that the bio-oscillator releases, the vein's thickness, and the cytoplasm's flow rate. As mentioned, the SMA ranks various dietary blocks according to their concentrations. Greater weights are assigned to the locations with higher concentrations, and vice versa. As a result, the SM's location is adjusted to better regions. Eq. (5) formulates this procedure.

$$\overrightarrow{X^*} = \begin{cases} rand \cdot (UB - LB) + LB, & rand < z \\ \overrightarrow{X_b(t)} + \overrightarrow{vb} \cdot \left(W \cdot \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}\right), & r < p \\ \overrightarrow{vc} \cdot \overrightarrow{X(t)}, & r \geq p \end{cases} \quad (5)$$

in which LB and UB are the lower and upper bounds, and $rand$ and r stand for the random value between 0 and 1.

c) Grabbing the food

The cytoplasmic flow in the veins is affected by the waves released by the biological oscillator. To simulate the variations of the SM's venous width, three vectors of $\overrightarrow{vc}$, $\overrightarrow{vb}$, and $\overrightarrow{W}$ are considered. $\overrightarrow{W}$ gives a mathematical presentation of the SM's oscillation frequency at different food concentrations. This parameter helps the SM to achieve a better food source by accelerating its movement toward high-quality ones and vice versa. The $\overrightarrow{vb}$ value randomly ranges in $[-a, a]$ and it heads to 0 as the number of iterations increases. The $\overrightarrow{vc}$ value ranges in [-1, 1] and finally reaches 0. During this stage, some organic members are assigned to explore the remaining areas even if the SM has reached a more potential source than earlier attempts. It enables the algorithm to seek a better food block all over the area. Notably, the SM decides whether to select the proposed food source or look for another one concerning the oscillation of $\overrightarrow{vb}$.

The Pseudo-code of the SMA is presented below:

Algorithm 1: Pseudo-code of SMA (Li *et al*. 2020).

Initialize the parameters *pop size*, $Max\_iteraition$;
Initialize the positions of slime mould (i = 1, 2, …, n);
While ($t \leq Max\_iteraition$)
Calculate the fitness of all slime mold; $Update\ bestFitness, Xb$.
Calculate the W by **Eq.** (3);
For $each\ search\ portion$
$Update\ p,vb,vc$;
$Update\ positions\ by$ **Eq**. (5);
End For
$t = t + 1$;
End While
Return $bestFitness, Xb$;

d) Computational complexity analysis

Considering different steps of the SMA (i.e., initialization, assessing the fitness, sorting, updating the weights, and updating the locations), the algorithm's complexity is explained in this section. Let N and T be the number of the SM's cells and the maximum number of iterations, respectively, in D-dimensional space. Then, (N), $(N + N\ log\ N)$, $O(N \times D)$, and $O(N \times D)$ are the computational complexity of initialization, fitness evaluation and sorting, weight update, and location update, respectively. Hence, the overall complexity of the SMA can be expressed as $O(N * (1 + T * N * (1 + log\ N + 2 * D)))$ (Li *et al*. 2020).

Benchmark trainers: Henry's law inspired one of the most recent metaheuristic algorithms: the HGSO. It was created by Hashim *et al*. (2019). A modified version was also created by Hashim *et al*. (2020). Cao *et al*. (2020) also optimized a regression SVM model's parameters using the HGSO. Each gas in this algorithm is originally given a place and a so-called characteristic "partial pressure." After that, the gasses are grouped into many clusters. The next step is to identify which gases are optimal. The location and solubility of each gas are modified to improve the solution's quality based on the algorithm's particular criteria. It's important to remember that upgrading the worst particles is seen as a way to get out of the local optimal region. Relevant works like (Shehabeldeen *et al*. 2020, Yıldız *et al*. 2020) provide mathematical details of the method. The HHO, created by Heidari *et al*. (2019), is the second benchmark algorithm. High applicability of the HHO has been demonstrated for a number of complicated situations, such as slope stability analysis (Moayedi *et al*. 2021b) and landslide susceptibility analysis (Bui *et al*. 2019). The coordinated behavior of Harris' hawks during a startling hunt that involves tracking, surrounding, approaching, and attacking is the basis of this algorithm. Three main phases go into the creation of these steps. The first is called exploration, which aims to locate and track the prey. The energy of the prey determines the following two steps. Attacking strategies are implemented during the exploitation stage, following the transition from exploration to exploitation. Please refer to (Chen *et al*. 2020a, b) for more details regarding the HHO.

3.2.2 Black Hole Algorithm (BHA)

A reliable stochastic optimization method for explaining a black hole's behavior in space is the black hole optimization algorithm. The techniques listed below describe how to simulate the black hole anomaly (BHA) (Kumar *et al*. 2015):

Step 1: There are both known and undiscovered stars in the universe. Black holes are formed when individual stars collide. Therefore, BHA starts with a population of stars randomly distributed around the universe. Every star in BHA has its fitness value assessed and maximized using a fitness function. The black hole is the best star with the highest fitness value. In Fig. 5, the BHA schema is shown. The green circles are stars, while the black circle is a black hole. They were scattered over the search region. It is referred to as "black" because it completely absorbs light and does not reflect it.

Step 2: In the actual world, a black hole is an object with a great density and a strong gravitational attraction. The surrounding stars exert a powerful gravitational pull as a consequence. BHA has behaved similarly. Every star began to gravitate toward the black hole following Eq. (6).

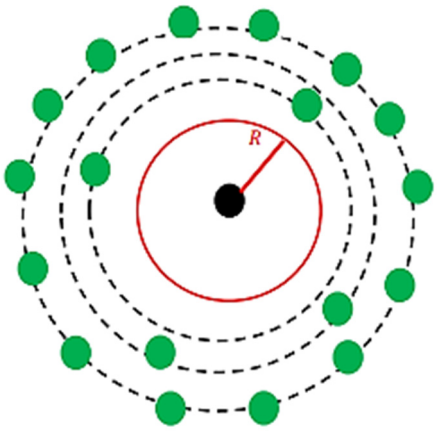


Fig. 5 Black Hole schema (after Kumar *et al*. (2015)

Step 3: The event horizon, which resembles a spherical, is the outer edge of a black hole in space. The name of the Schwarzschild radius knows the event horizon. The black hole's event horizon is depicted as a red circle in Fig. 5. Eqs. (7) and (8) may be used to determine the Schwarzschild radius in real space and BHA, respectively.

Step 4: A black hole will swallow a star due to its extreme gravity and density, and it will disappear when it moves over its event horizon. Since the escapee's speed equals light, nothing can leave the event horizon. The Euclidean distance between a black hole and a star is computed via BHA. If the distance exceeds the Schwarzschild radius, replace the current star with a new one at a random point inside the search area.

Step 5: If a star in BHA travels to a location less expensive than a black hole, its position should be modified.

$$X_i(t+1) = X_i(t) + rand \times \left(X_{BH} - X_i(t)\right) \quad i = 1, 2, \ldots, N \tag{6}$$

$$R = 2GM/C^2 \tag{7}$$

$$R = \frac{f_{BH}}{\sum_{i=1}^{N} f_i} \tag{8}$$

The coordinates of the *i*th star at iterations *t* and *t* + 1 are represented by $X_i(t)$ and $X_i(t+1)$, respectively, and Rand denotes a uniform distribution with a 0 to 1 scale. The letter N stands for the number of stars. The symbol indicates the black hole's position in the exploration space. X_{BH}. The letters M, G, and C stand for the gravitational constant, the mass of the black hole, and the speed of light, respectively. The fitness value of the black hole is represented by f_{BH}, and that of the ith star by f_i.

3.2.3 Multi-Verse Optimizer (MVO)

Based on multiverse physics theory, Mirjalili *et al*. (2016) presented the multiverse optimizer (MVO) approach. The multiverse theory states that the universe was created by several big bangs (Mirjalili *et al*. 2016, Moayedi and Mosavi 2021). It builds a mathematical model of the wheel mechanism for transferring universe products and the wormhole (white/black) tunnels that connect two universes. The inflation rates of the universes are rated after each iteration, and a roulette wheel selects the universe with the greatest inflation rate to have a white hole in it. The MVO model is described mathematically as follows:

Think about the following

$$u = \begin{bmatrix} x_1^1 & x_1^2 & \ldots & x_1^d \\ x_2^1 & x_2^2 & \ldots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^1 & \cdots & x_n^d \end{bmatrix} \tag{9}$$

If *n* is the total number of universes (possible solutions) and *d* is the total number of parameters (variables), then

$$x_k^j = \begin{cases} x_k^j & r_1 < NI(u_i) \\ x_i^j & r_1 \geq NI(u_i) \end{cases} \tag{10}$$

Where r_1 is an integer between 0 and 1, u_i is the *i*-th universe, $NI(u_i)$ is the normalized inflation rate of the i^{th} universe, and x_i^j is the j^{th} parameter of the i^{th} world.

The method of transportation is as follows, assuming a wormhole tunnel connects one world to the best universe

$$x_i^j = \begin{cases} \begin{cases} x_j + TDR + \left(\left(ub_j - lb_j\right) * r_4 + lb_j\right) & if\ r_3 < 0.5 \\ x_j - TDR + \left(\left(ub_j - lb_j\right) * r_4 + lb_j\right) & if\ r_3 \geq 0.5 \end{cases} & r_2 < WEP \\ x_i^j & r_2 \geq WEP \end{cases} \tag{11}$$

Where lb_j and ub_j are the lower and upper bounds of the *j*th variable, x_j is the *j*-th parameter of the best universe, WEP and TDR are the worm existence probability and traveling distance rate, respectively, and r_2, r_3, and r_4 are random values between [0, 1]. The WEP and TDR formulas are as follows

$$WEP = MIN + l \times \left(\frac{max - min}{L}\right) \tag{12}$$

$$TDR = 1 - \frac{l^{1/p}}{L^{1/p}} \tag{13}$$

L indicates the maximum number of iterations that can be carried out, *p* indicates the exploitation accuracy over iterations, set at 6, and min and max indicate the minimum and maximum values, respectively, set at 0.2 and 1. In this case, l represents the iteration that is currently being carried out. The maximum number of iterations for this investigation, utilizing 30 universes, is 500.

3.2.4 Vortex Search (VS)

Doğan and Ölmez (2015) developed VS based on a single solution. The variable interval (step) size phenomenon, which greatly enhances the search mechanism's efficiency, sets the VS algorithm apart. To get the best answer, the VS algorithm software considers both neighborhood weak and strong locality. Moreover, the algorithm nearly reaches the optimal point when it reacts to the revised solution in an exploitative (strong locality) way to get the best outcome. Thus, the desired radius decreases as the number of iterations increases. The VS approach deterministically produces a solution that converges to the global optimization point within the given lower and higher bounds. The best design for the analog filter group delay and an analog active filter component has been chosen using the VS technique to assess performance (Doğan and Yüksel 2015).

When applying the Vortex Search Optimization (VSO) technique, strong and weak localities significantly influence the effectiveness and usefulness of optimal solutions. Weak and strong locales represent minor and major modifications

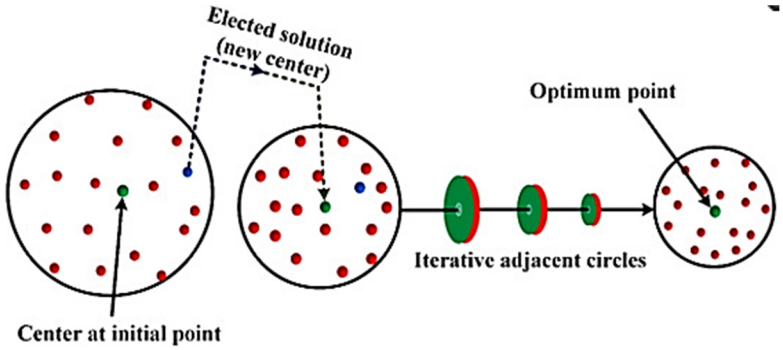


Fig. 6 Employing a two-dimensional nested circle model of the VS to represent the working search process (after Doğan and Ölmez (2015))

to the existing solution. When the search process starts, a weak locality is needed; on the other hand, a strong locality is needed when the optimization strategy effectively converges near the most optimum answer. The main estimation (VS initialization), candidate solutions, current solution substitution, and radius reduction technique are the important phases of the VS algorithm search.

3.2.4.1 Primary estimation

The search strategy for the nested vortex pattern is described for the given technique. A two-dimensional nested circle representation of the VS approach is shown in Fig. 6. Given the initial circumstances, the outermost circle's diameter determines the search's pivot point. From Eq. (14), the rivet or starting center (μ_0) may be determined as follows

$$\mu_0 = \frac{upper\ bound + lower\ bound}{2} \tag{14}$$

3.2.4.2 Candidate solutions

Once the initial response has been assessed, neighbor solutions $C_i(X) = \{x_1, x_2, x_3, \dots, x_k\}, k = 1, 2, \dots, n$ are determined as follows using a Gaussian distribution, as indicated by Eq. (15).

$$p(\xi|\mu, v) = \frac{1}{\sqrt{(2\pi)^d |v|}} \exp\left(-\frac{1}{2}\frac{(\xi-\mu)^T(\xi-\mu)}{v}\right) \tag{15}$$

where ξ and μ signify the sample mean (selected as the center) and the vector for a randomly generated variable, respectively, and n stands for the number of candidates' local optimal points, i for the count, and d for the dimension number. Moreover, v denotes the covariance matrix given by Eq. (16) in the following way

$$v = s^2[I], \text{dimension: d} \times \text{d} \tag{16}$$

where s^2 represents the variance distribution, and I stands for the identity matrix. The standard deviation s_0 under the starting circumstances given by Eq. (17) as follows

$$s_0(= r_0) = \frac{\max(upper\ bound) - \min(lower\ bound)}{2} \tag{17}$$

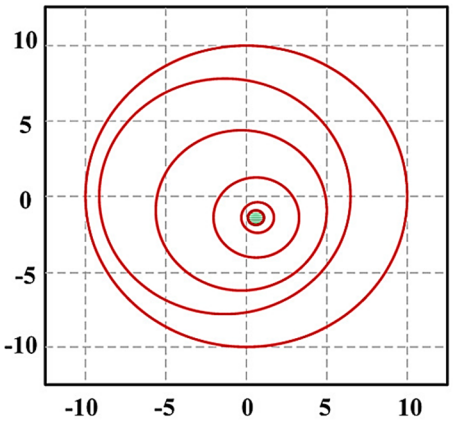


Fig. 7 The VS algorithm's search area for vortex patterns (after Doğan and Ölmez (2015)

To completely cover the weak proximity in the neighborhood search region, s sub 0 is considered the initial radius (r sub 0 for a weak locality at the beginning stage.

3.2.4.3 Current results substitution

The nearest candidate result is replaced with the current solution in the replacement phase when the solution. $X' \in C_0(X), (i = 0)$ from $C_0(X)$, where the current circle center μ_0 fits inside the search space bounds. If the new solutions are beyond the search space boundaries, the candidate solutions are moved within the specified borders as indicated by Eq. (18) as follows

$$(lower\ bound)^d \le s_k^d \le (upper\ bound)^d \tag{18}$$

where d is the bound boundary dimension, and k is an integer between 1 and n. In the subsequent iteration, the circle's center is indicated by the obtained optimal solution, X'. The circle's active radius (r_1) decreases in the second step of the coeval phase, and a new set of vectors $C_1(X)$ is produced over the new center. The new solution set $C_1(X)$ is evaluated in the second stage of the selection process via $X'C_1(X)$. It is preserved if the chosen answer progresses to the more complex one.

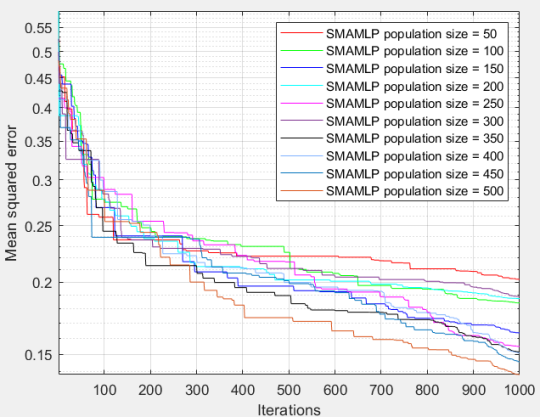
Likewise, the artificial maintenance of the third-step allotted center in Fig. 7 as the new advanced/optimal solution is evident. Until the completion requirements are

met, the phenomena keep happening.

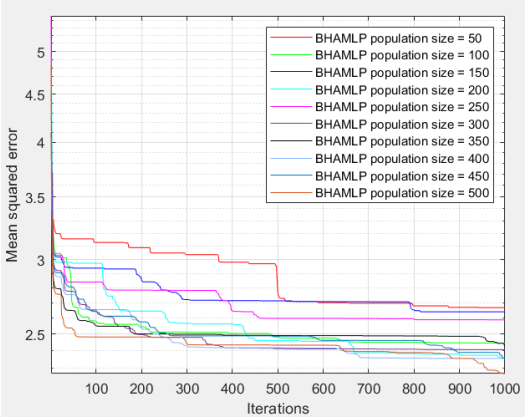
3.2.5 Whale Optimization Algorithm (WOA)

The whale optimization algorithm is a revolutionary meta-heuristic system that mimics the social behavior of humpback whales (Nasiri and Khiyabani 2018, Tien Bui *et al*. 2021). Mirjalili and Lewis (2016) created this meta-heuristic technique. The spiral bubble-net feeding maneuver theoretically represents this optimization strategy. The fact that humpback whales are the only known animal that feeds using bubble nets is noteworthy (Mirjalili and Lewis 2016, Trivedi *et al*. 2018). Two behaviors—the spiral bubble-net feeding maneuver and the decreasing encircling mechanism—have been shown to update the whales' location during optimization. Since the exact position of the optimal design in the search space is uncertain, the basic WOA technique assumes that the best candidate solution at the moment is the optimum or extremely near to it (Mirjalili and Lewis 2016). It also presumes that when one search agent outperforms the others, they will move to match. The following is a quick explanation of the WOA method from Mirjalili and Lewis (2016).

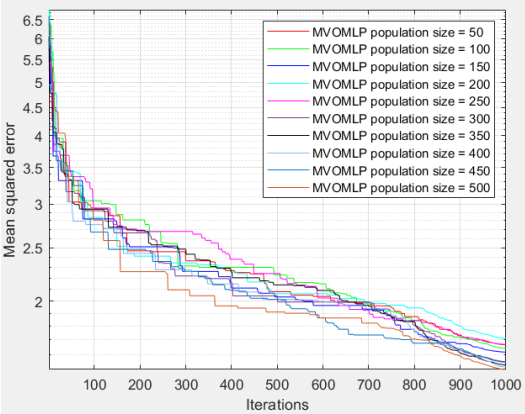
A collection of randomly generated populations is the starting point for the WOA. At every iteration, search



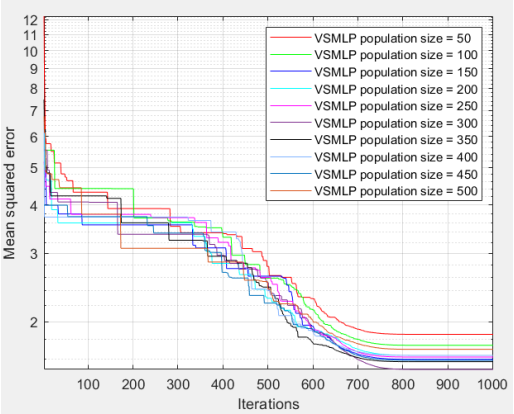
(a) SMA-MLP



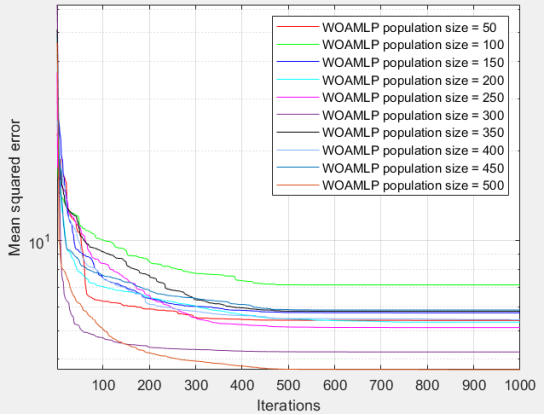
(b) BHA-MLP



(c) MVO-MLP

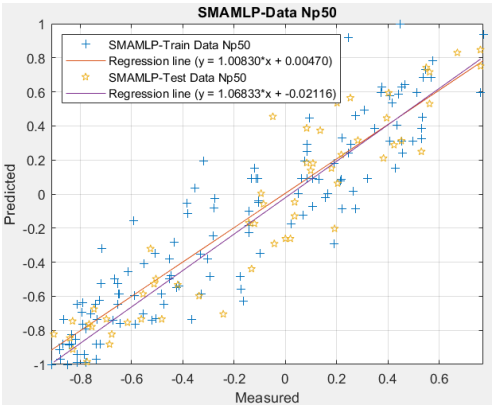


(d) VS-MLP

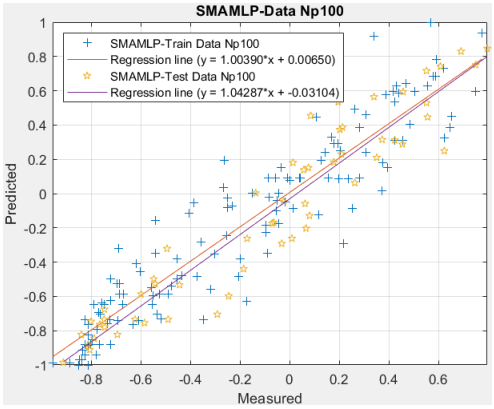


(e) WOA-MLP

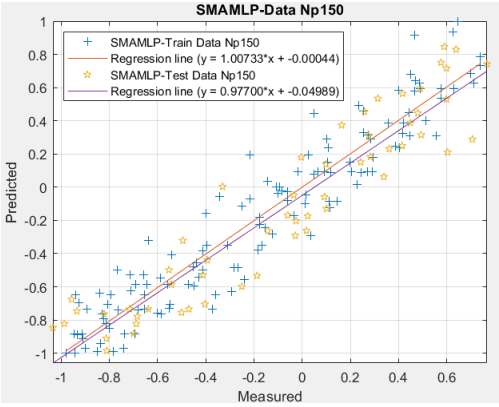
Fig. 8 Iterations include SMA-MLP, BHA-MLP, MVO-MLP, VS-MLP, and WOA-MLP



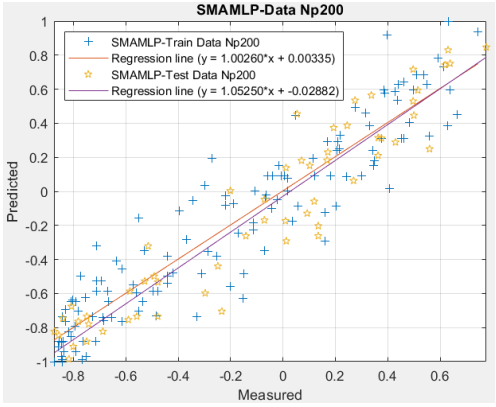
(a) SMA-MLP - N_p50



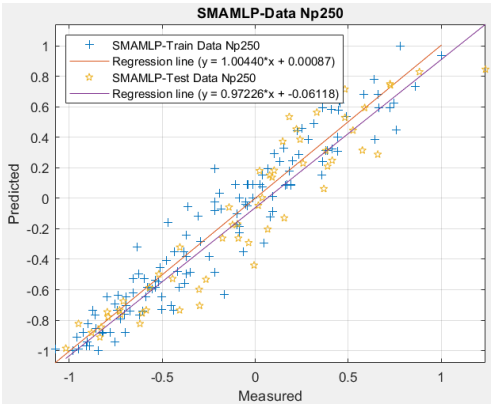
(b) SMA-MLP - N_p100



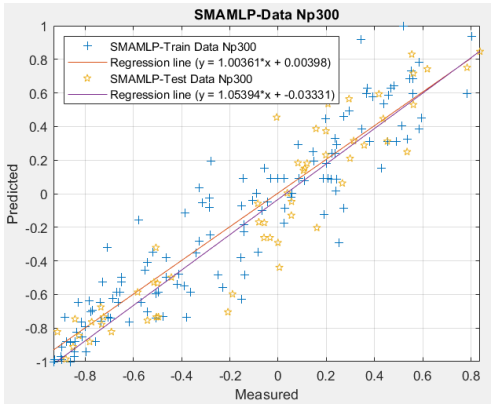
(c) SMA-MLP - N_p150



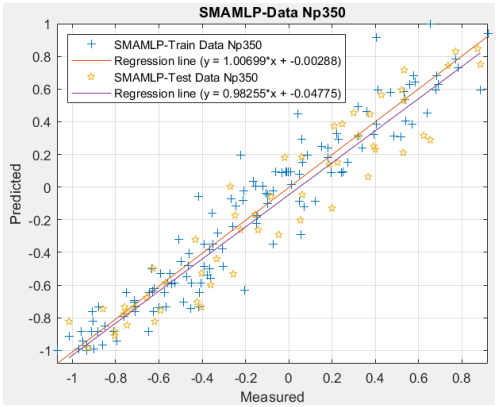
(d) SMA-MLP - N_p 200



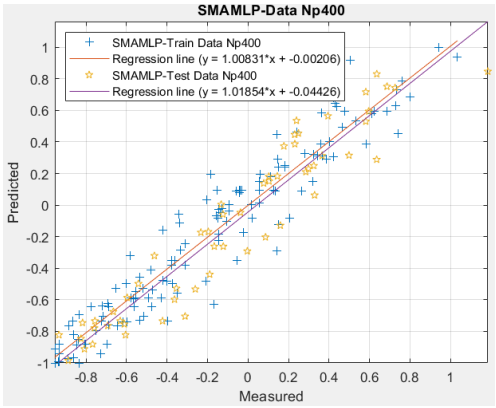
(e) SMA-MLP - N_p250



(f) SMA-MLP - N_p300

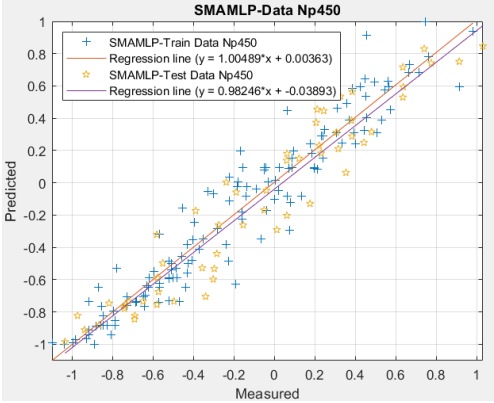


(g) SMA-MLP - N_p350

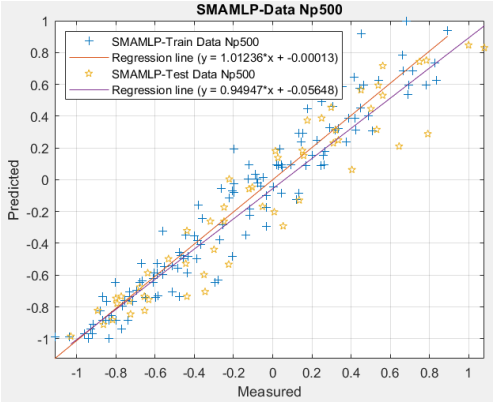


(h) SMA-MLP - N_p400

Fig. 9 Continued



(i) SMA-MLP - N_p450



(j) SMA-MLP - N_p500

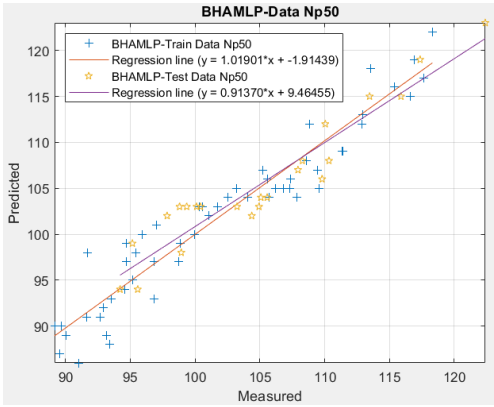
Fig. 9 Accuracy results of training and testing dataset for different proposed SMA-MLP structure

agents adjust their locations based on the value of the A vector. The updating procedure is covered in the sections that follow. This process is continued until the termination condition is met.

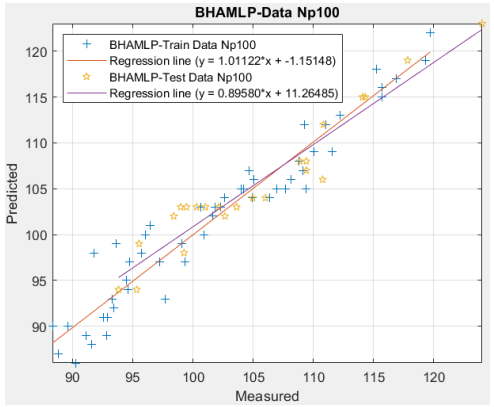
The WOA algorithm consists of two steps: exploration and exploitation. This method seamlessly transitions from the phase of discovery to that of exploitation. The alteration in the value of the A vector brings about the transition. The value of a vector decreases with time; while |A| ≥ 1, half of the iterations are utilized for exploration, and when |A| < 1, the other half is used for exploitation. The character jj represents the absolute value. The calculation of vector A is displayed below

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \tag{19}$$

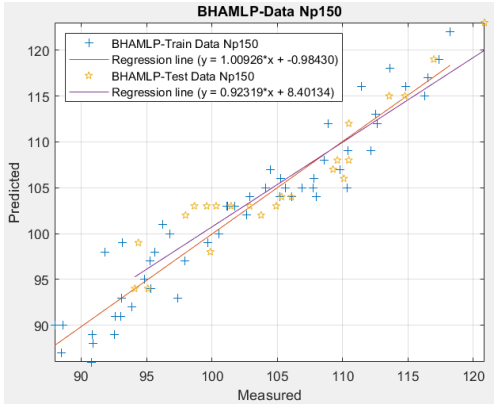
where $\vec{a}$ A decrease linearly over iterations from 2 to 0, and r is a random vector with a range of [0, 1].



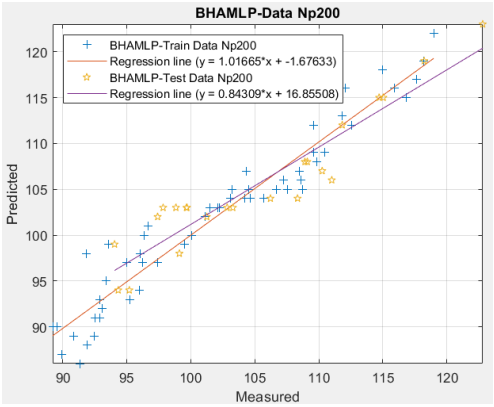
(a) BHA-MLP- N_p50



(b) BHA-MLP- N_p100

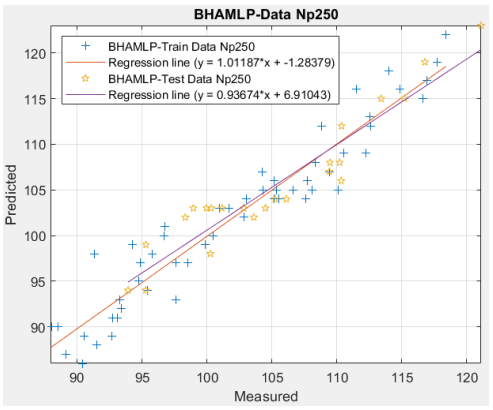


(c) BHA-MLP- N_p150

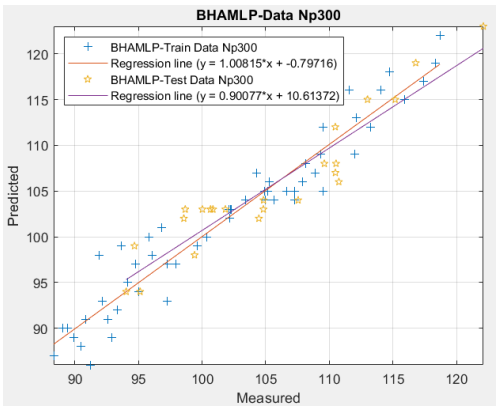


(d) BHA-MLP- N_p 200

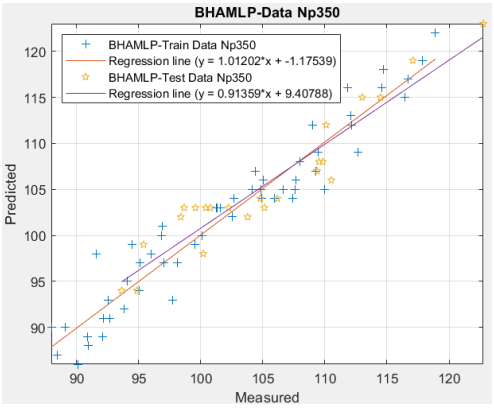
Fig. 10 Continued



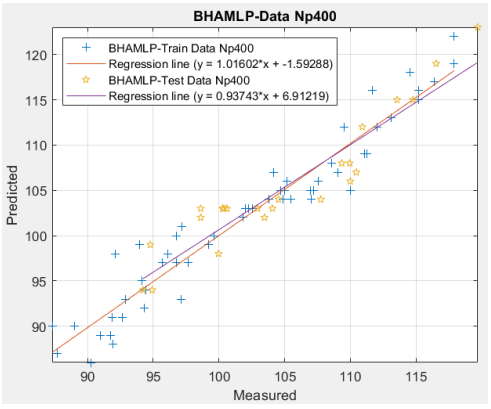
(e) BHA-MLP- N_p250



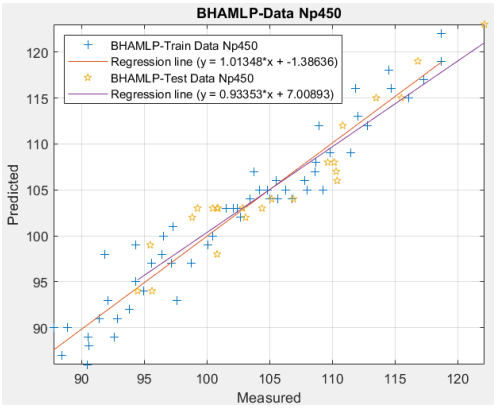
(f) BHA-MLP- N_p300



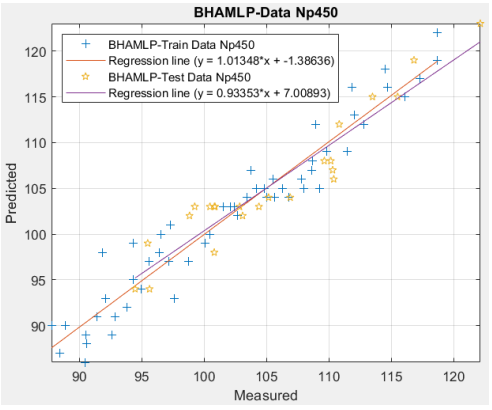
(g) BHA-MLP- N_p350



(h) BHA-MLP- N_p400



(i) BHA-MLP- N_p450



(j) BHA-MLP- N_p500

Fig. 10 Accuracy results of training and testing datasets for different proposed BHA-MLP structure

4. Model evaluation and presentation

The prediction network generated the artificial neural network (ANN), and each model's output was assessed.

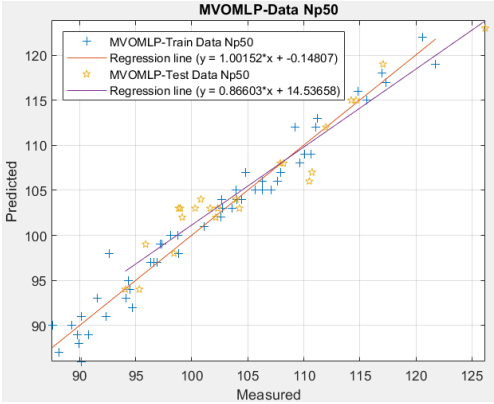
A development of the most significant optimization method parameters used is presented in this section. To evaluate each technique's effectiveness, various statistical indicators (SMA-MLP, BHA-MLP, MVO-MLP, VS-MLP, and WOA-MLP) were employed to rank the outcomes. Mean square error (MSE) and the R^2 score are several statistical indicators that form the foundation of the total ranking systems (TRS)

One of the most often used methods for evaluating accuracy when compared to one is the R^2 score, which may be calculated as follows

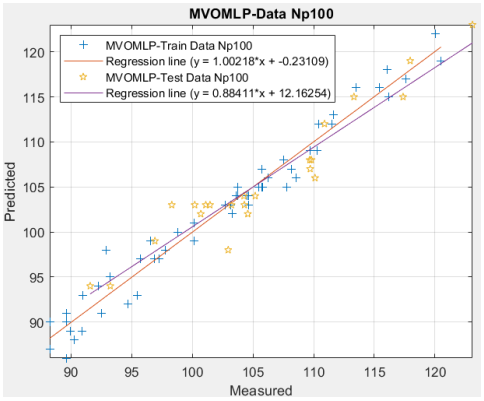
$$R^2 = 1 - \frac{\sum_{i=0}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=0}^{n}(y_i - \overline{y_i})^2} \tag{20}$$

$y_i, \widehat{y_i}, and\ \overline{y_i}$, where it consists of mean, projected, and actual values. The population size is denoted by n. Better precision is indicated by quantities that are not equal but closer to 1.

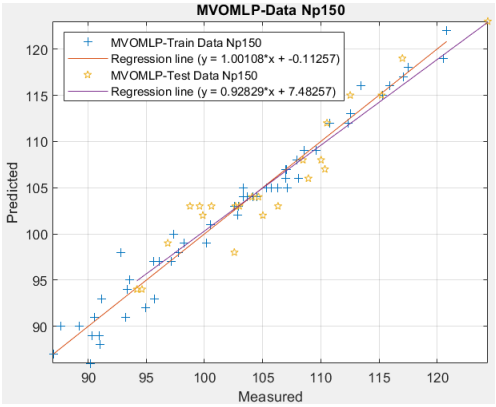
Mean square error is calculated based on the variation between actual and expected values. The more closely its



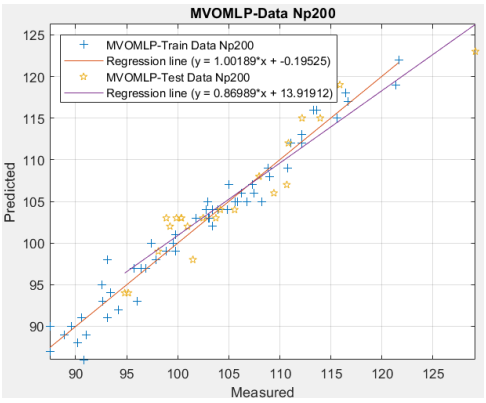
(a) MVO-MLP- N_p50



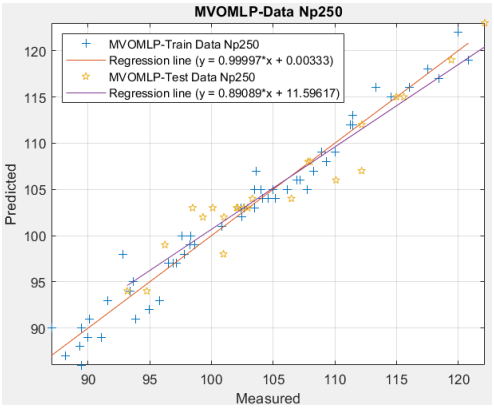
(b) MVO-MLP- N_p100



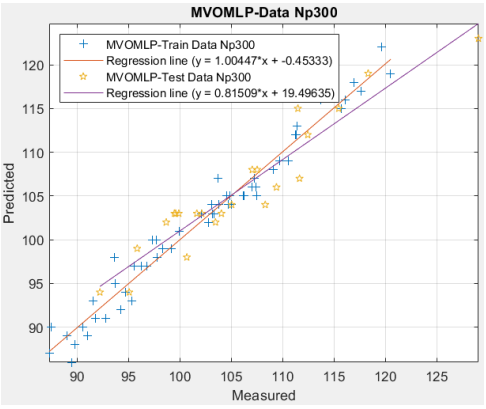
(c) MVO-MLP- N_p150



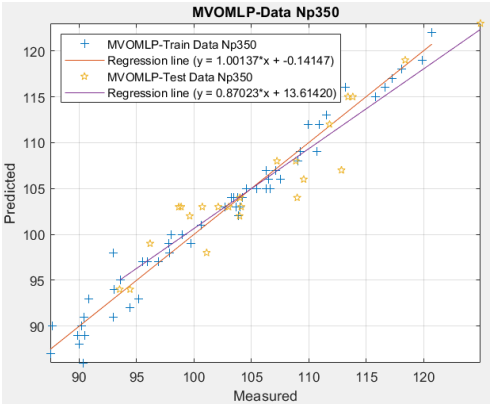
(d) MVO-MLP- N_p 200



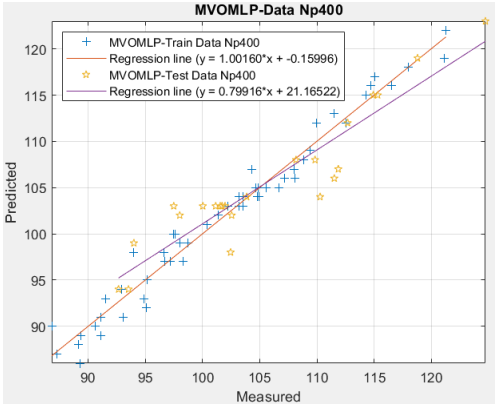
(e) MVO-MLP- N_p250



(f) MVO-MLP- N_p300

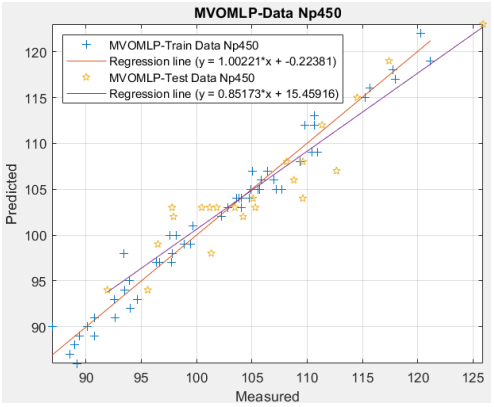


(g) MVO-MLP- N_p350

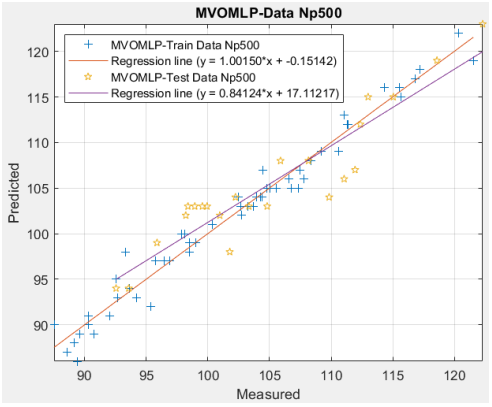


(h) MVO-MLP- N_p400

Fig. 11 Continued



(i) MVO-MLP- N_p450



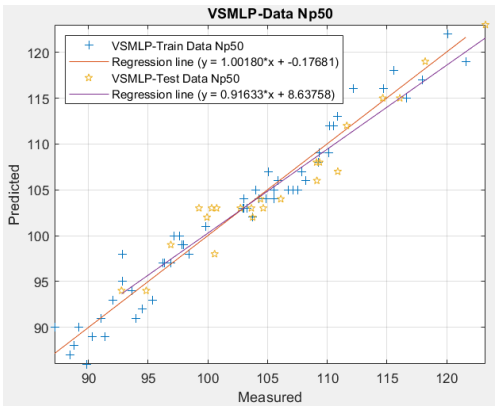
(j) MVO-MLP- N_p500

Fig. 11 Accuracy results of training and testing datasets for different proposed MVO-MLP structure

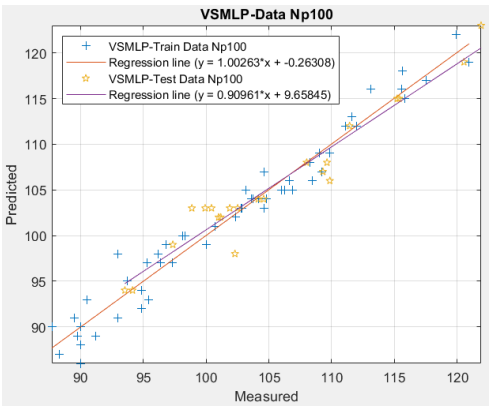
values resemble zero, the more accurate the model's forecast is. This parameter can be obtained using the function below

$$\text{MSE} = \frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2 \tag{21}$$

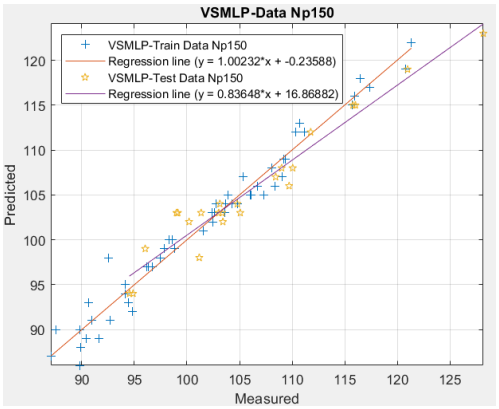
Fig. 8 shows the performance results for different acceleration constants and NPOPs (50, 100, 150, 200, 250, 300, 350, 400, 450, and 500). These results showed that the SMA-MLP algorithm (N_{POP} = 450) (Fig. 8(a)), the BHA-MLP algorithm (N_{POP} = 500) (Fig. 8(b)), the MVO-MLP algorithm (N_{POP} = 250) (Fig. 8(c)), the VS-MLP (N_{POP} = 150) (Fig. 8(d)), and the WOAMLP (N_{POP} = 500) (Fig. 8(e)) are the algorithms that most accurately predict the output. The results were gathered from 77 extensive laboratory studies that sought to determine ultimate carrying capacity and were also forecasted using the suggested artificial intelligence (AI) models. The hybrid VS-MLP model might be regarded as an extraordinary prediction network (with better accuracy than the conventional ANN model) in forecasting strength, even though all of the suggested



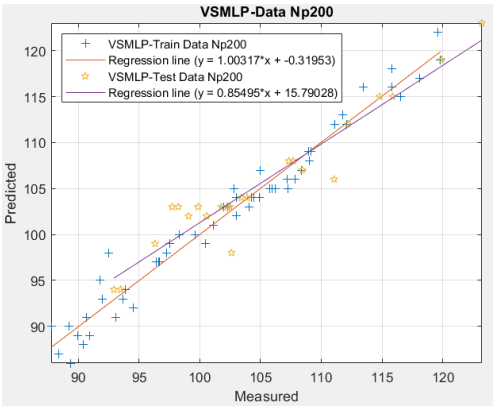
(a) VS-MLP- N_p50



(b) VS-MLP- N_p100

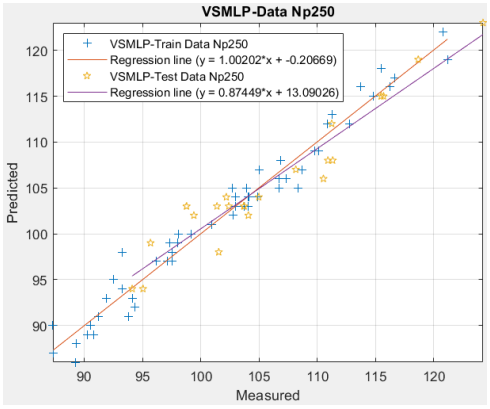


(c) VS-MLP- N_p150

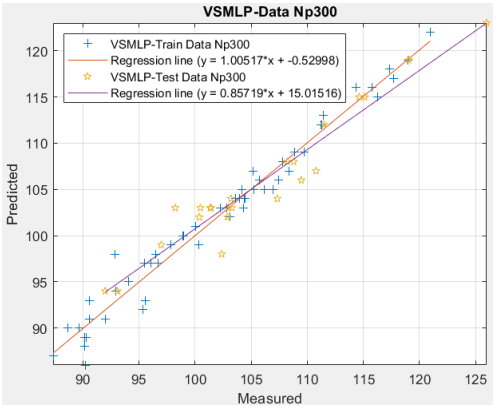


(d) VS-MLP- N_p 200

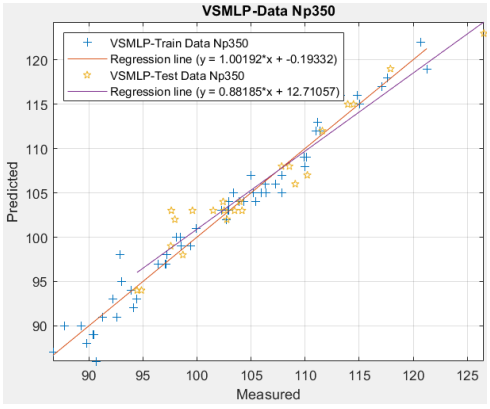
Fig. 12 Continued



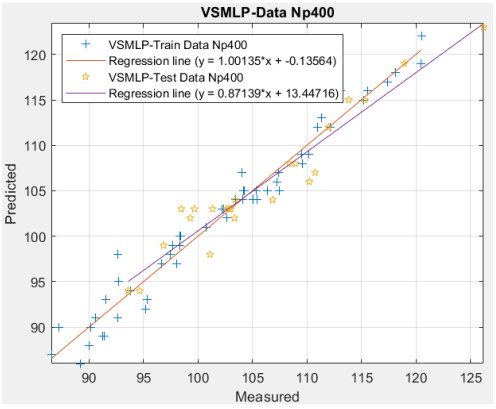
(e) VS-MLP- N_p250



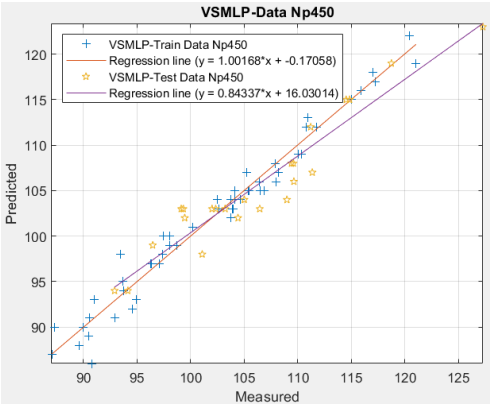
(f) VS-MLP- N_p300



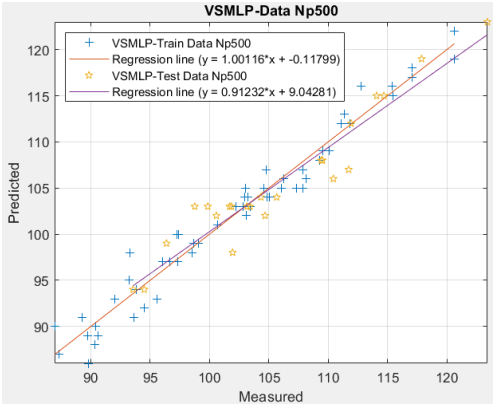
(g) VS-MLP- N_p350



(h) VS-MLP- N_p400



(i) VS-MLP- N_p450



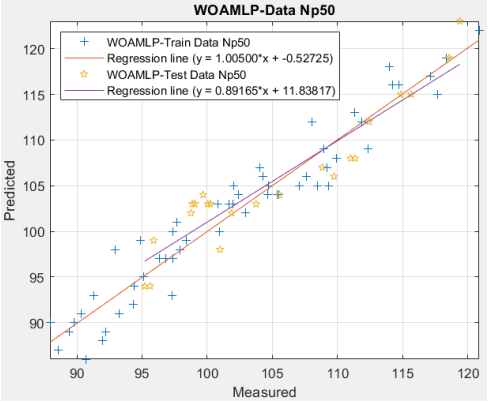
(j) VS-MLP- N_p500

Fig. 12 Accuracy results of training and testing datasets for different proposed VS-MLP structure

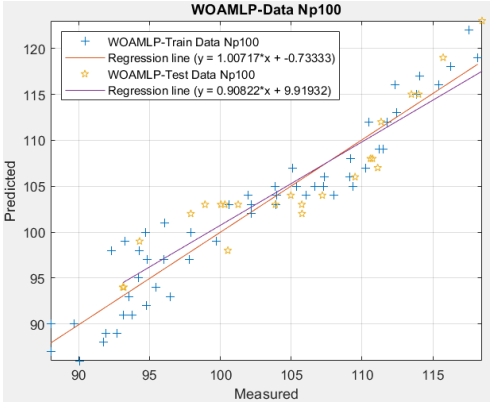
models produced respectable estimate results in predicting S. Emphasizing the appropriateness of the learning technique across all examined prediction models—such as those with high R^2 or low MSE rates—is crucial. Furthermore, the VS-MLP predictive networks might perform better across the board regarding statistical metrics (MSE, R^2).

In contrast to previous methods, this remained accurate during the training and testing. The training and testing networks' optimal performance outcomes make this clear. For training and testing the enhanced VS-MLP prediction models, the MSE and R^2 values were 1.59934, 0.98412 in training phase and 1.75627, 0.96882 in testing phase. Moreover, in-network results are quite accurate, particularly for datasets used for testing. The optimization technique of VS-MLP may accurately predict the final S.

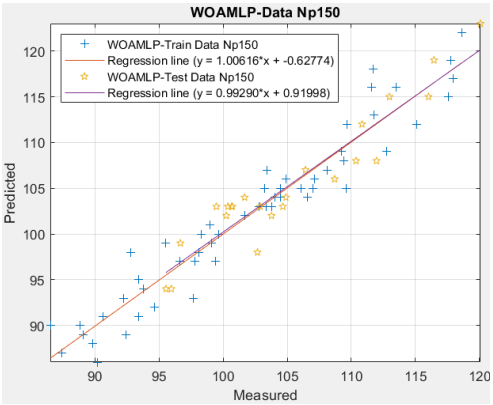
All of the network results with R^2 and RMSE in the upcoming tables must be considered for testing and training. The suggested method and the effect of population size on their search optimization are evaluated based on total rank. It should be mentioned that accuracy increases with R^2's proximity to 1. Values around 0 for the MSE signify increased accuracy. The SMA-MLP optimization algorithm yielded the following R^2 values for population sizes: testing (0.93956, 0.95224, 0.93817, 0.94492, 0.94527, 0.93917, 0.95203, 0.95244, 0.95676, and 0.95165)



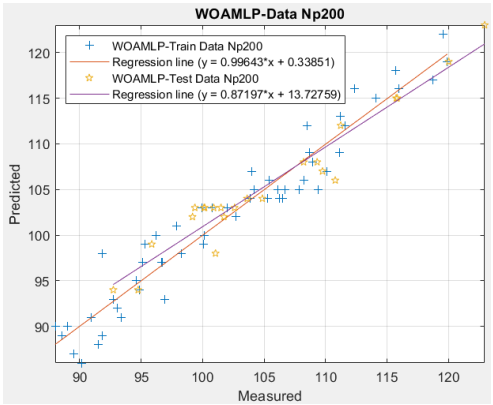
(a) WOA-MLP- Np50



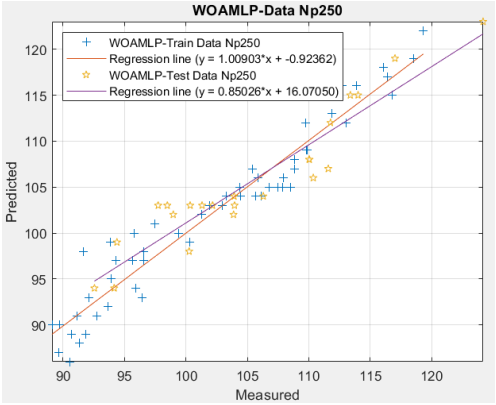
(b) WOA-MLP- N_p100



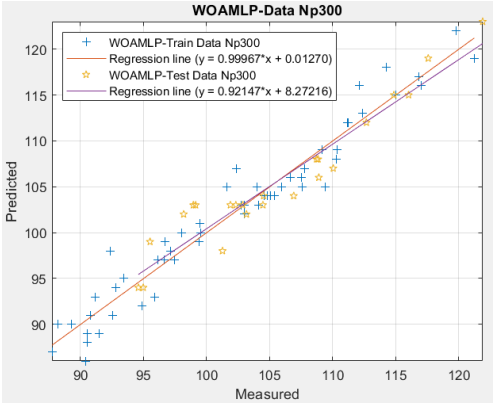
(c) WOA-MLP- N_p150



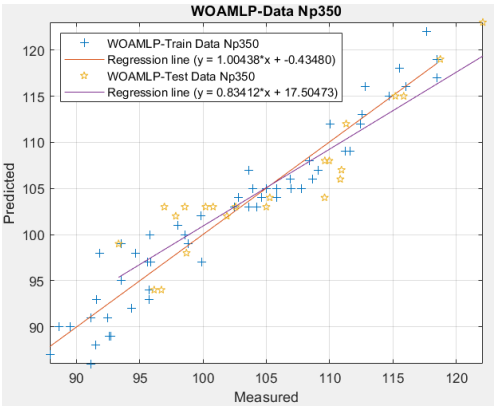
(d) WOA-MLP- N_p 200



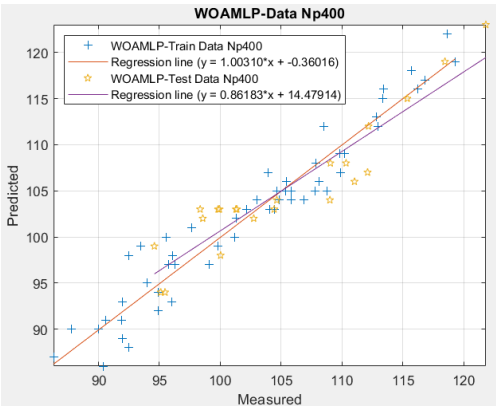
(e) WOA-MLP- N_p250



(f) WOA-MLP- N_p300

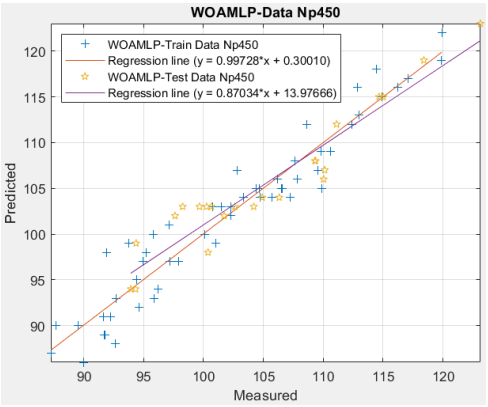


(g) WOA-MLP- N_p350

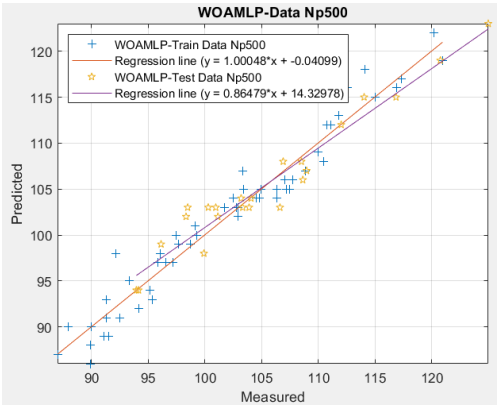


(h) WOA-MLP- N_p400

Fig. 13 Continued



(i) WOA-MLP- N_p450



(j) WOA-MLP- N_p500

Fig. 13 Accuracy results of training and testing datasets for different proposed WOA-MLP structure

Table 2 Hybrid network results for the SMA-MLP predictive model

Population size	Network result				Scoring				Total score	Rank
	Train		Test		Train		Test			
	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2		
50	0.20207	0.92200	0.18504	0.93956	1	1	3	3	8	9
100	0.18394	0.93583	0.16502	0.95224	4	4	8	8	24	5
150	0.16341	0.94972	0.18707	0.93817	5	5	1	1	12	8
200	0.18721	0.93345	0.17688	0.94492	3	3	4	4	14	7
250	0.15482	0.95499	0.17634	0.94527	6	6	5	5	22	6
300	0.18863	0.93240	0.18561	0.93917	2	2	2	2	8	9
350	0.15142	0.95699	0.16537	0.95203	7	7	7	7	28	4
400	0.15077	0.95736	0.16468	0.95244	8	8	9	9	34	2
450	0.14554	0.96033	0.15720	0.95676	9	9	10	10	38	1
500	0.13831	0.96424	0.16602	0.95165	10	10	6	6	32	3

Table 3 Hybrid network results for the BHA-MLP predictive model

Population size	Network result				Scoring				Total score	Rank
	Train		Test		Train		Test			
	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2		
50	2.66237	0.95535	2.21066	0.95013	1	1	8	8	18	7
100	2.44372	0.96252	2.03098	0.95807	4	4	10	10	28	2
150	2.63851	0.95617	2.37138	0.94238	2	2	4	4	12	10
200	2.34976	0.96540	2.45999	0.93785	9	9	1	1	20	6
250	2.58588	0.95793	2.28742	0.94650	3	3	5	5	16	8
300	2.40417	0.96375	2.45036	0.93835	6	6	2	2	16	8
350	2.42851	0.96299	2.27714	0.94700	5	5	6	6	22	4
400	2.35055	0.96537	2.44874	0.93843	8	7	3	3	21	5
450	2.35080	0.96537	2.25936	0.94784	7	7	7	7	28	2
500	2.26560	0.96787	2.09686	0.95525	10	10	9	9	38	1

and training (0.92200, 0.93583, 0.94972, 0.93345, 0.95499, 0.93240, 0.95699, 0.95736, 0.96033, and 0.96424). Additionally, 0.20207, 0.18394, 0.16341, 0.18721, 0.15482, 0.18863, 0.15142, 0.15077, 0.14554, and 0.13831 were the MSE results obtained from training and testing this method, respectively, as well as (0.18504, 0.16502, 0.18707, 0.17688, 0.17634, 0.18561, 0.16537, 0.16468, 0.15720, and 0.16602). Clearly, the recommended population size in the SMA-MLP optimization methods strongly affects the training and testing network accuracies, with a population

Table 4 Hybrid network results for the MVO-MLP predictive model

Population size	Network result				Scoring				Total score	Rank
	Train		Test		Train		Test			
	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2		
50	1.66487	0.9828	2.01017	0.95895	3	3	9	9	24	1
100	1.64146	0.9833	2.14362	0.95318	4	4	8	8	24	1
150	1.61499	0.9838	2.26937	0.94737	5	5	7	7	24	1
200	1.70988	0.9818	2.34949	0.94347	1	1	5	5	12	10
250	1.66827	0.9827	1.99923	0.9594	2	2	10	10	24	1
300	1.55124	0.9851	2.31183	0.94532	6	6	6	6	24	1
350	1.53055	0.9855	2.36332	0.94278	8	8	4	4	24	1
400	1.51758	0.9857	2.52781	0.93425	9	9	3	3	24	1
450	1.54829	0.9851	2.55914	0.93255	7	7	2	2	18	9
500	1.50182	0.9860	2.64308	0.92788	10	10	1	1	22	8

Table 5 Hybrid network results for the VS-MLP predictive model

Population size	Network result				Scoring				Total score	Rank
	Train		Test		Train		Test			
	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2		
50	1.85531	0.97857	1.80151	0.96717	1	1	9	9	20	4
100	1.73838	0.98121	1.80806	0.96692	2	2	8	8	20	4
150	1.59934	0.98412	1.75627	0.96882	7	7	10	10	34	1
200	1.63730	0.98335	2.04271	0.95758	5	5	4	4	18	6
250	1.62117	0.98368	2.09906	0.95515	6	6	3	3	18	6
300	1.50596	0.98593	1.91057	0.96299	10	10	6	6	32	2
350	1.57696	0.98457	1.88513	0.96399	9	9	7	7	32	2
400	1.63776	0.98334	1.94725	0.96153	4	4	5	5	18	6
450	1.59039	0.98430	2.18201	0.95144	8	8	1	1	18	6
500	1.69732	0.98210	2.12291	0.95410	3	3	2	2	10	10

Table 6 Hybrid network results for the WOA-MLP predictive model

Population size	Network result				Scoring				Total score	Rank
	Train		Test		Train		Test			
	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2		
50	2.32472	0.96614	2.49398	0.93606	6	6	4	4	20	5
100	2.66619	0.95522	2.82027	0.91744	1	1	1	1	4	10
150	2.39364	0.96407	2.36340	0.94278	4	4	6	6	20	5
200	2.30951	0.96659	1.83248	0.96601	7	7	9	9	32	3
250	2.26049	0.96802	2.37339	0.94228	8	8	5	5	26	4
300	2.05781	0.97357	2.08603	0.95572	9	9	8	8	34	2
350	2.40857	0.96361	2.77107	0.92042	3	3	2	2	10	9
400	2.33053	0.96597	2.52727	0.93428	5	5	3	3	16	8
450	2.42041	0.96325	2.14474	0.95313	2	2	7	7	18	7
500	1.92181	0.97699	1.73667	0.96952	10	10	10	10	40	1

size equal to 450 exhibiting the greatest accuracy.

The outcomes of the BHA-MLP prediction model's optimization are displayed in Table 3. The approach exhibited maximum accuracy when the population size was 500. Although the population size of 450 has a good test score, its rank is equivalent to 2. Therefore, reviewing the

Table 7 Total ranking of best-fitted model for the five employed hybrid methods

Used method	Population size	Training dataset R^2	Testing dataset R^2	Score		Total score	Rank
				Training	Testing		
SMAMLP	450	0.9603	0.95676	1	2	3	4
BHAMLP	500	0.9679	0.95525	2	1	3	4
MVOMLP	250	0.9827	0.9594	4	3	7	3
VSMLP	150	0.9841	0.96882	5	4	9	1
WOAMLP	500	0.9770	0.96952	3	5	8	2

overall R^2 and RMSE training and testing results is crucial. The outcome of the MVO-MLP prediction model is

displayed in Table 4. In this instance, the 250-person population has a rank of 1. You can view other scores. The testing R^2 values deviate excessively from 1 compared to the training values.

The VS-MLP prediction models for population sizes of 150 were determined to have the best model, as Table 5 illustrates. Notably, although the population size of 300 or 350 produced reasonable training prediction outputs, their performance in testing was deemed unacceptable. Therefore,

comparable to MVO-MLP, the suggested VS-MLP model with a population size of 500 offered the best predictive network estimate utilizing high-strength concrete strength.

The VS-MLP prediction models for population sizes of 150 were determined to have the best model, as Table 5 illustrates. Notably, although the population size of 300 or 350 produced reasonable training prediction outputs, their performance in testing was deemed unacceptable. Therefore, comparable to MVO-MLP, the suggested VS-MLP model with a population size of 150 offered the best predictive network estimate utilizing high-strength concrete strength.

Table 6 shows the accuracy obtained by the causal method. As shown in Table 6, WOA-MLP method is the best prediction method used after VS-MLP method.

Table 7 displays the overall ranking of the suggested best-fitted models for each approach. The VS-MLP approaches yield the best predictive networks that approximate high-strength concrete strength out of all the structures suggested in this study. In this regard, the MSE and R^2 for the training and testing datasets of the VS-MLP were, respectively, 1.59934 and 0.98412 and 1.75627 and 0.96882, respectively. After completing all of the computations in this article, we discovered that population size, or "swarm size," is one of the most important variables in the hybrid prediction approaches. For the swarm sizes of 500, 450, 500, 250 and 150, the best-fit prediction networks are found for the hybrid approaches of WOA-MLP, SMA-MLP, BHA-MLP, MVO-MLP, and VS-MLP, respectively.

5. Error analysis

The process known as error analysis is examining and analyzing the mistakes or faults committed in a certain job or system. It is frequently used to pinpoint and comprehend the causes of mistakes and enhance the functionality of systems or models in various disciplines, including linguistics, computer science, statistics, and engineering. Error analysis in machine learning refers to examining a model's mistakes while executing a particular job, such as speech recognition, machine translation, or text categorization. The objectives include understanding the many kinds of mistakes being made, their root causes, and viable tactics for enhancing system performance. Usually, error analysis entails the following actions:

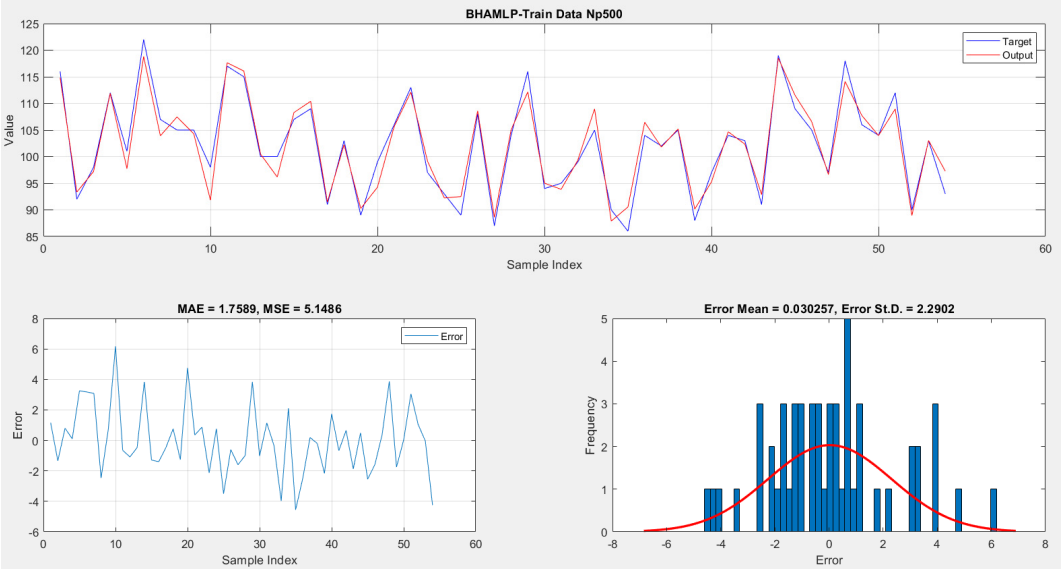
1. Data collection and annotation: After gathering or sampling a dataset, human annotators examine and annotate the information following the intended task. Documents may bear labels indicating their respective classifications, for instance, when it comes to text classification.

2. Training and testing the model: An independent test dataset is used to gauge the model's performance once trained on the annotated data. Other assessment measures, such as accuracy, precision, recall, or F1 score, may be utilized depending on the job.

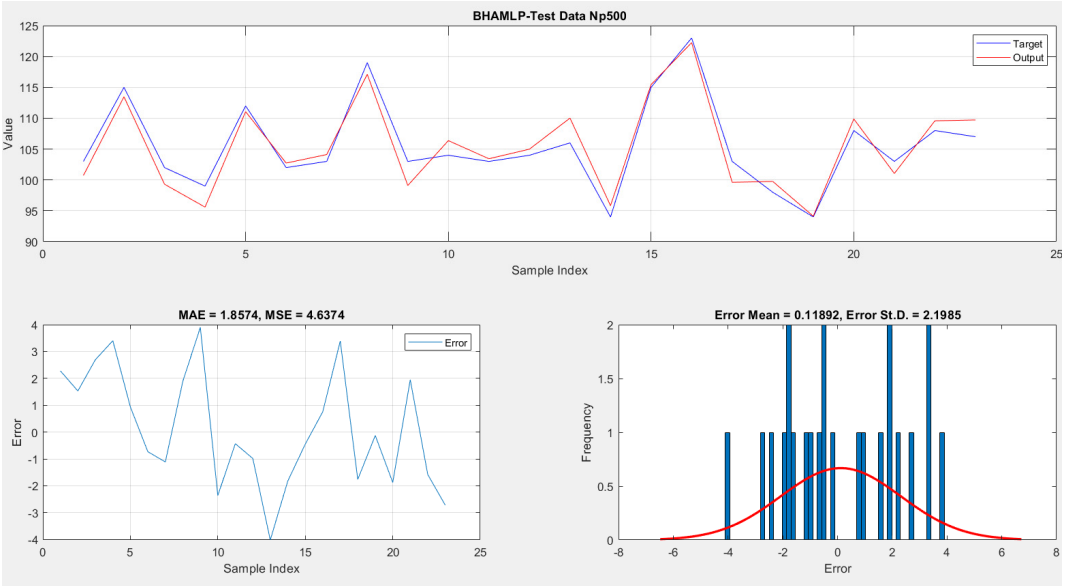
3. Error identification: By contrasting the model's predictions with the ground truth labels, flaws in the model are found. These mistakes may be misclassified, false positives, false negatives, or other task-specific mistakes of a particular kind.

4. Error classification and analysis: To find trends, recurring causes of errors, or any systemic biases in the model's operation, the errors are classified and examined. The aspects of the incorrectly categorized examples may be examined, the effects of certain features or input data may be investigated, or the model's design or training procedure may be considered.

5. Error mitigation strategies: These are plans created to deal with and lessen the errors found based on the knowledge gathered from the error analysis. This may entail enhancing the feature representation, gathering additional training data, fine-tuning the model architecture, or using post-processing methods to fix particular mistakes. Researchers and developers may make better judgments and



(a) BHA-MLP 500 train



(b) BHA-MLP 500 test

Fig. 15 Error frequency and MAE variance for the proposed best-fit structures of (a) BHA-MLP 500; train and (b) BHA-MLP 500 test

enhance the model's performance by conducting error analysis to obtain a deeper knowledge of the constraints and difficulties related to a certain activity or system.

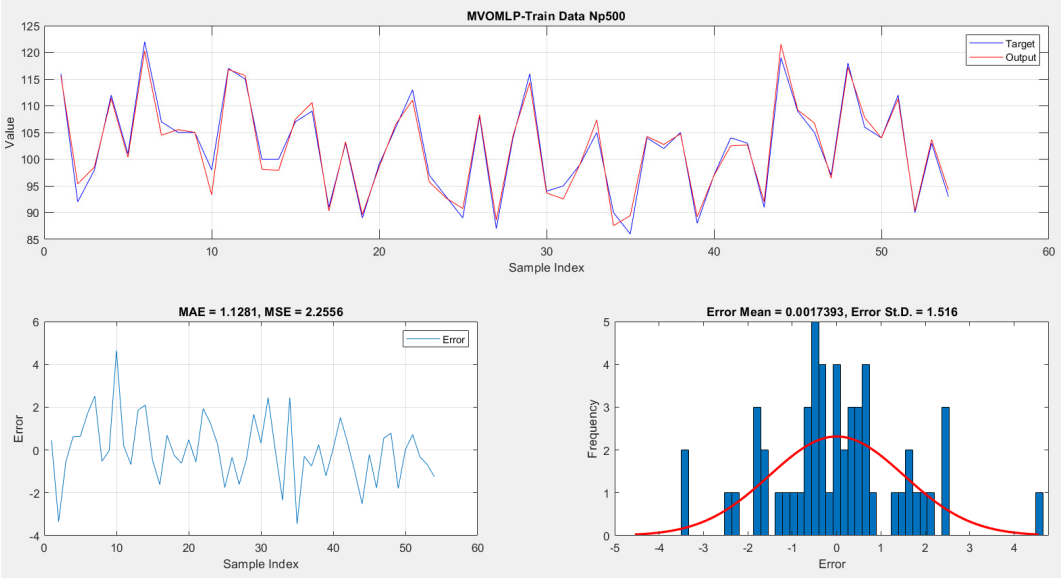
Two of the error functions used to evaluate the accuracy of each algorithm's best-fit models are used in this section. The standard deviations (Std. D.) and mean absolute error (MAE) are these functions. The definition of mean absolute error is

$$\text{MAE}=\frac{1}{n}\sum_{i=0}^{n}|y_i-\hat{y}_i| \tag{22}$$

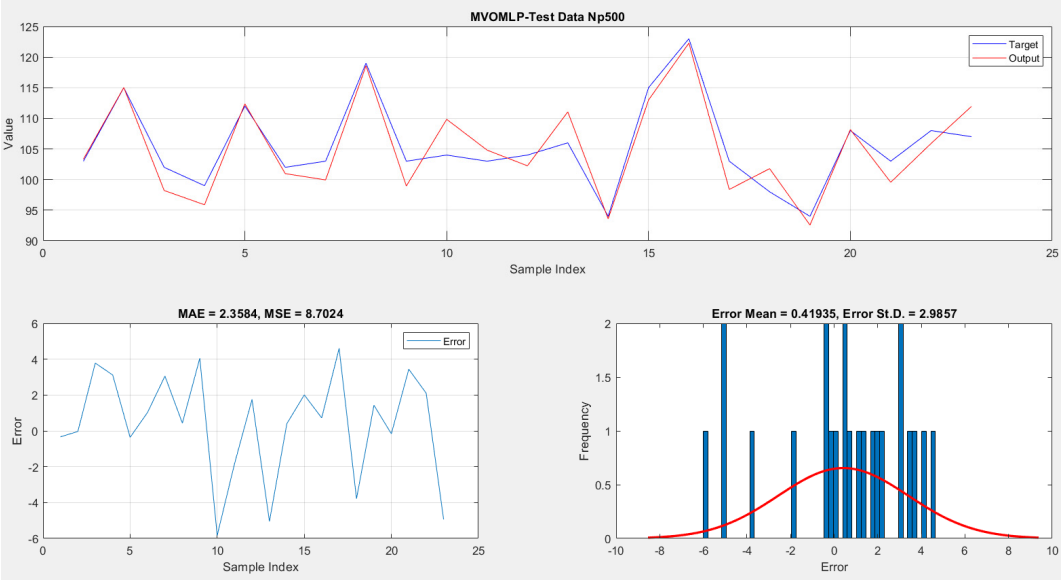
Where n, $\hat{y}_i$, and y_i Represent the population size, actual values, and anticipated values, respectively; the greater the difference between the two, the greater the MAE. Thus, accuracy increases as the MAE decreases. The following figures show that the MAE values produced are all less than one for each of the best-fit models, indicating correctness. What follows is the standard deviation

$$\sigma=\sqrt{\frac{\sum_{i=0}^{n}(y_i-\mu)^2}{n}} \tag{23}$$

The remaining parameters are the same as the top function, and μ is the population mean. This function looks at relationships between each data point and the mean of the dataset. Therefore, the graph's skewness is regulated to assess the models' accuracy using this criterion. The accuracy of the distribution chart increases with its degree of normalcy. This is true for all models, as demonstrated by the accompanying figures.



(a) MVO-MLP 500 train

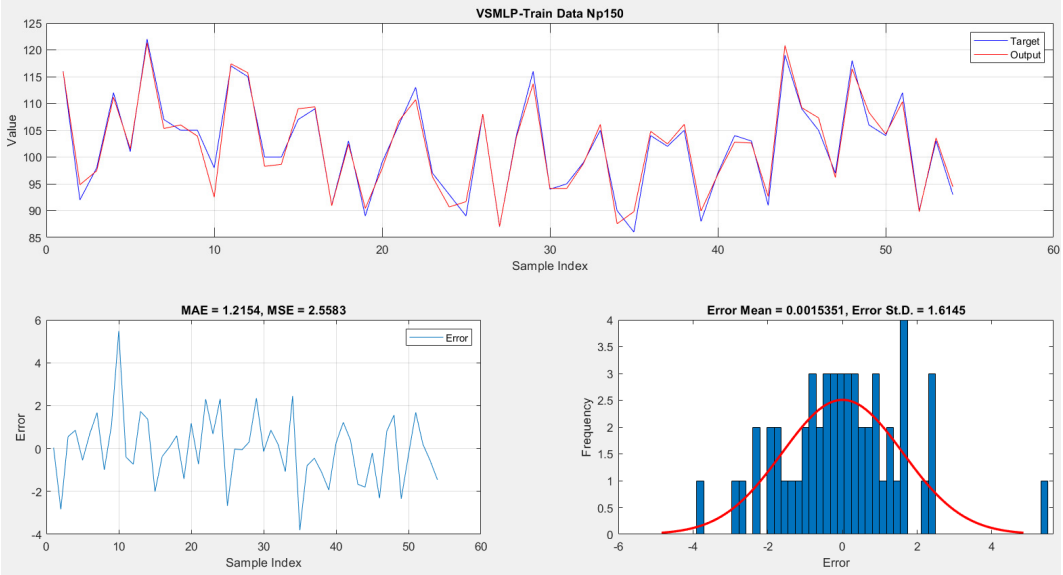


(b) MVO-MLP 500 test

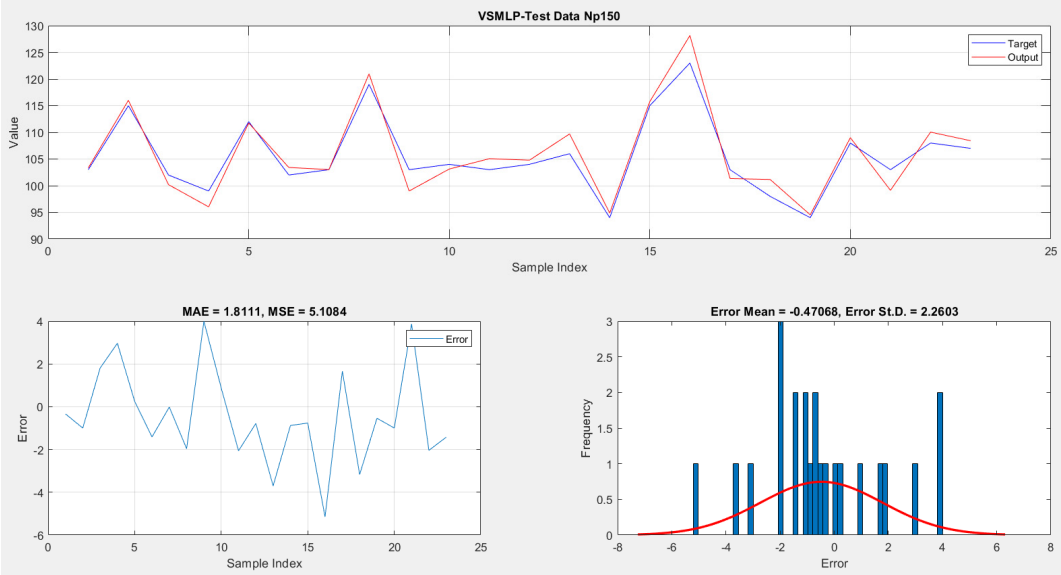
Fig. 16 Error frequency and MAE variance for the proposed best-fit structures of (a) MVO-MLP 500; train and (b) MVO-MLP 500 test

6. Limitations and future work

Metaheuristic algorithms have shown significant success in solving complex optimization problems, but like any approach, they come with limitations and areas for future research. Here are some common limitations and potential future research directions in metaheuristic analysis:
Many metaheuristic algorithms have parameters that need to be tuned for optimal performance, and their sensitivity to parameter settings can be a limitation. Research is needed to develop more robust algorithms that are less sensitive to parameter changes or to explore automated methods for parameter tuning. Metaheuristics may exhibit bias toward certain types of problems or problem structures. Methods to mitigate algorithmic bias and develop more versatile metaheuristics across a broad range of problem domains. Some metaheuristics may face challenges in scaling up to efficiently handle large-scale or high-dimensional optimization problems. Explore strategies to enhance the scalability of metaheuristic algorithms, potentially through parallelization, distributed computing, or problem-specific adaptations. Premature convergence to suboptimal solutions is a common issue in metaheuristics. Techniques to prevent premature convergence, including adaptive mechanisms for maintaining diversity, dynamic exploration strategies, and hybrid approaches. Some metaheuristic algorithms lack strong theoretical foundations, making it challenging to guarantee their convergence or optimality. Conduct more theoretical analyses of metaheuristic algorithms to understand their convergence properties and performance guarantees better. Handling constraints, especially in real-world optimization problems with complex constraints, is



(a) VS-MLP 150 train

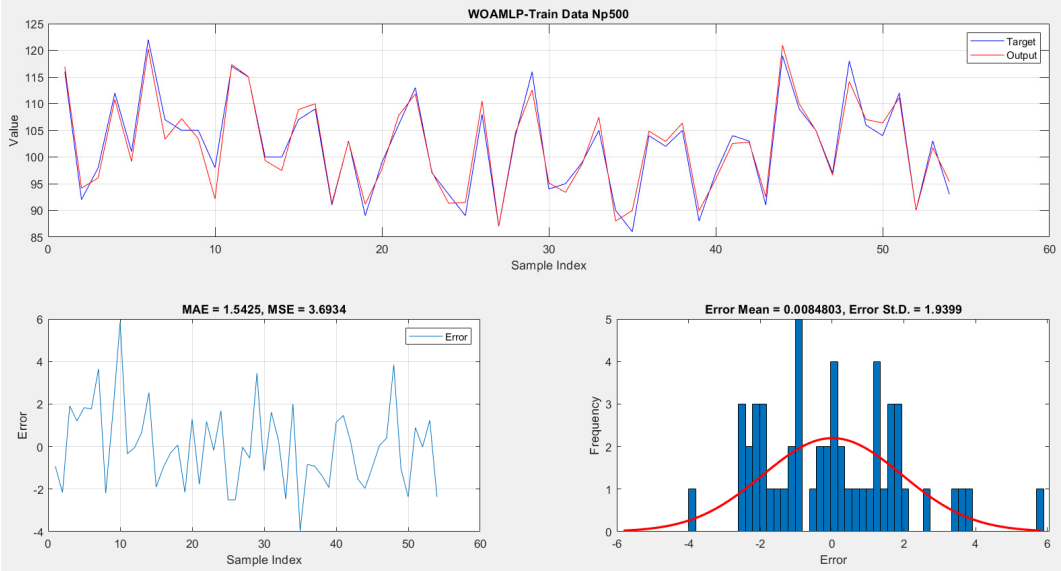


(b) VS-MLP 150 test

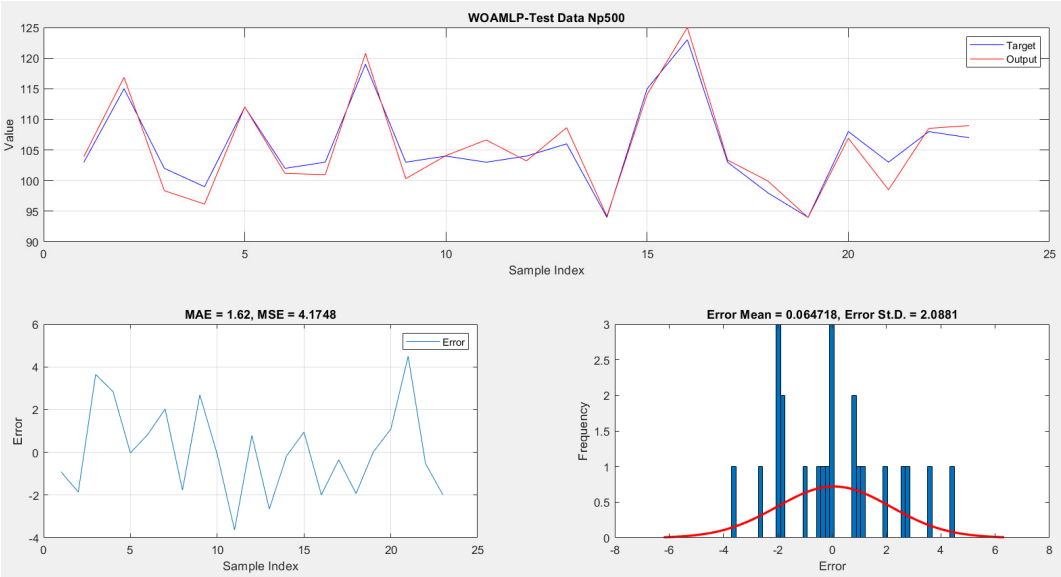
Fig. 17 Error frequency and MAE variance for the proposed best-fit structures of (a) VS-MLP 150 train; and (b) VS-MLP 150 test

a challenge for many metaheuristic algorithms. Research is needed to develop more effective approaches for handling constraints, possibly through the integration of constraint-handling techniques or hybridization with constraint-specific methods.

Further explore and develop metaheuristic algorithms specifically designed for multi-objective optimization problems, considering the challenges of Pareto dominance, diversityhpg preservation, and decision-maker preferences. Investigate the development of metaheuristic algorithms with adaptive mechanisms that enable self-learning and continuous improvement over time. Explore innovative ways to combine different metaheuristics or integrate metaheuristics with machine learning techniques to create hybrid approaches that leverage the strengths of each method. Develop metaheuristic algorithms capable of handling dynamic optimization problems where the characteristics of the problem change over time. Further investigation is required for the parallelization and distribution of metaheuristic algorithms to harness the power of modern computing architectures for solving large-scale problems efficiently. Research Need: Focus on enhancing the explainability and interpretability of metaheuristic algorithms, particularly in applications where decision-makers require a clear understanding of the optimization process. Explore methods to enhance the transferability of metaheuristic algorithms across different problem domains, reducing the need for problem-specific algorithm adjustments. Establish standardized benchmarks and evaluation criteria to facilitate fair comparisons between different metaheuristic algorithms and promote transparency in reporting experimental results. Addressing



(a) WOA-MLP 500 train



(b) WOA-MLP 500 test

Fig. 18 Error frequency and MAE variance for the proposed best-fit structures of (a) WOA-MLP 500 train; and (b) WOA-MLP 500 test

these limitations and exploring these future research directions can contribute to the ongoing development and improvement of metaheuristic algorithms, making them more robust, versatile, and effective in solving a wide range of optimization problems.

7. Conclusions

The current work investigated the viability of using hybrid neural network models to assess the strength of concretes containing fly ash and superplasticizers. Five artificial intelligence-based prediction models are put forth in this work to forecast concrete strength effectively. The data-driven model is evaluated using a dataset comprising 77 distinct high-strength concrete experiments. To validate the superiority of the combined techniques, the performance of conventional MLP is compared against five standard forecasting techniques (WOA-MLP, SMA-MLP, BHA-MLP, MVO-MLP, and VS-MLP). According to the findings, the VS-MLP techniques produced the greatest R^2 values, with 0.9841 in training and 0.96882 in testing, respectively, and the lowest RMSEs, 1.59934 in training and 1.75627 in testing. Adopting VS-MLP resulted in notable performance increases over previous approaches. However, the BHA-MLP and SMA-MLP techniques did not perform well throughout the training phase. The results showed that the VS-MLP predictive model, with populations of 150, respectively, had the highest accuracy in forecasting the strength of the concrete. Moreover, the suggested methodology might be utilized for additional data-driven issues in material science or civil engineering

undertakings. Despite its limitations, this study shows that the created model is preferable in terms of computing speed and error statistics analysis.

Acknowledgments

This study was supported by Guangdong Provincial Key Laboratory of Green Construction and Intelligent Operation & Maintenance for Off- shore Infrastructure, and supported by Special Projects in Key Fields of Higher Education in Guangdong Province (No. 2023ZDZX4044).

References

Beekman, M. and Latty, T. (2015), "Brainless but Multi-Headed: Decision Making by the Acellular Slime Mould Physarum polycephalum", *J. Molecular Biol*., **427**(23), 3734-3743. https://doi.org/10.1016/j.jmb.2015.07.007

Bui, D.T., Moayedi, H., Kalantar, B., Osouli, A., Pradhan, B., Nguyen, H. and Rashid, A.S.A. (2019), "Harris hawks optimization: A novel swarm intelligence technique for spatial assessment of landslide susceptibility", *Sensors*, **19**, 3590.

Cao, B., Zhao, J., Lv, Z., Gu, Y., Yang, P. and Halgamuge, S.K. (2020), "Multiobjective evolution of fuzzy rough neural network via distributed parallelism for stock prediction", *IEEE Transact. Fuzzy Syst*., **28**(5), 939-952. https://doi.org/10.1109/TFUZZ.2020.2972207

Cemalgil, S., Gül, E., Onat, O. and Aruntaş, H.Y. (2022), "A novel prediction model for durability properties of concrete modified with steel fiber and Silica Fume by using Hybridized GRELM", *Constr. Build. Mater*., **341**, 127856. https://doi.org/10.1016/j.conbuildmat.2022.127856

Cemalgil, S., Onat, O. and Aruntaş, H.Y. (2023), "An Estimation Proposal for Engineering Properties of Modified Concrete by using Standalone and Hybrid GRELM", *Iran. J. Sci. Technol. Transact. Civil Eng*., **47**(3), 1357-1377. https://doi.org/10.1007/s40996-022-01005-6

Chandwani, V., Agrawal, V. and Nagar, R. (2015), "Modeling slump of ready mix concrete using genetic algorithms assisted training of Artificial Neural Networks", *Expert Syst. Applicat*., **42**(2), 885-893. http://doi.org/10.14716/ijtech.v6i2.213

Chatterjee, A., Saha, J. and Mukherjee, J. (2022), "Clustering with multi-layered perceptron", *Pattern Recogn. Lett*., **155**, 92-99. https://doi.org/10.1016/j.patrec.2022.02.009

Chen, H., Heidari, A.A., Chen, H., Wang, M., Pan, Z. and Gandomi, A.H. (2020a), "Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies", *Future Generat. Comput. Syst*., **111**, 175-198. https://doi.org/10.1016/j.future.2020.04.008

Chen, H., Jiao, S., Wang, M., Heidari, A.A. and Zhao, X. (2020b), "Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts", *J. Cleaner Product*., **244**, 118778. https://doi.org/10.1016/j.jclepro.2019.118778

Chen, L., Chen, Z., Xie, Z., Wei, L., Hua, J., Huang, L. and Yap, P.-S. (2023), "Recent developments on natural fiber concrete: A review of properties, sustainability, applications, barriers, and opportunities", *Develop. Built Environ*., **16**, 100255. https://doi.org/10.1016/j.dibe.2023.100255

Dai, W. (2021), "Safety Evaluation of Traffic System with Historical Data Based on Markov Process and Deep-Reinforcement Learning", *J. Computat. Methods Eng. Applicat.*, 1-14. https://ojs.sgsci.org/journals/jcmea/article/view/141

Dai, W. (2022), "Evaluation and Improvement of Carrying Capacity of a Traffic System", *Innov. Appl. Eng. Technol.*, 1-9. https://doi.org/10.58195/iaet.v1i1.001

Dai, W. (2023), "Design of Traffic Improvement Plan for Line 1 Baijiahu Station of Nanjing Metro", *Innov. Appl. Eng. Technol.* https://doi.org/10.58195/iaet.v2i1.133

Doğan, B. and Ölmez, T. (2015), "A new metaheuristic for numerical function optimization: Vortex Search algorithm", *Infor. Sci*., **293**, 125-145. https://doi.org/10.1016/j.ins.2014.08.053

Doğan, B. and Yüksel, A. (2015), "Analog filter group delay optimization using the Vortex Search algorithm", *Proceedings of the 23nd Signal Processing and Communications Applications Conference* (*SIU*), pp. 288-291.

Gardner, M.W. and Dorling, S. (1998), "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences", *Atmosph. Environ*., **32**(14-15), 2627-2636. https://doi.org/10.1016/S1352-2310(97)00447-0

Greco, C., Pace, P., Basagni, S. and Fortino, G. (2021), "Jamming detection at the edge of drone networks using multi-layer perceptrons and decision trees", *Appl. Soft Comput*., **111**, 107806. https://doi.org/10.1016/j.asoc.2021.107806

Guo, M., Huang, H., Zhang, W., Xue, C. and Huang, M. (2022), "Assessment of RC frame capacity subjected to a loss of corner column", *J. Struct. Eng*., **148**(9), 04022122. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003423

Han, Y., Shao, S., Fang, B., Shi, T., Zhang, B., Wang, X. and Zhao, X. (2023), "Chloride ion penetration resistance of matrix and interfacial transition zone of multi-walled carbon nanotube-reinforced concrete", *J. Build. Eng*., **72**, 106587. https://doi.org/10.1016/j.jobe.2023.106587

Hashim, F.A., Houssein, E.H., Mabrouk, M.S., Al-Atabany, W. and Mirjalili, S. (2019), "Henry gas solubility optimization: A novel physics-based algorithm", *Future Generat. Comput. Syst*., **101**, 646-667. https://doi.org/10.1016/j.future.2019.07.015

Hashim, F.A., Houssein, E.H., Hussain, K., Mabrouk, M.S. and Al-Atabany, W. (2020), "A modified Henry gas solubility optimization for solving motif discovery problem", *Neural Comput. Applicat*., **32**(14), 10759-10771. https://doi.org/10.1007/s00521-019-04611-0

He, H., Shuang, E., Ai, L., Wang, X., Yao, J., He, C. and Cheng, B. (2023), "Exploiting machine learning for controlled synthesis of carbon dots-based corrosion inhibitors", *J. Cleaner Product*., **419**, 138210. https://doi.org/10.1016/j.jclepro.2023.138210

Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H. (2019), "Harris hawks optimization: Algorithm and applications", *Future Generat. Comput. Syst*., **97**, 849-872. https://doi.org/10.1016/j.future.2019.02.028

Hu, D., Li, Y., Yang, X., Liang, X., Zhang, K. and Liang, X. (2023), "Experiment and Application of NATM Tunnel Deformation Monitoring Based on 3D Laser Scanning", Structural Control and Health Monitoring, **2023**, 3341788. https://doi.org/10.1155/2023/3341788

Huang, B. and Wang, J. (2020), "Deep-reinforcement-learning-based capacity scheduling for PV-battery storage system", *IEEE Transact. Smart Grid*, **12**(3), 2272-2283. https://doi.org/10.1109/TSG.2020.3047890

Huang, B. and Wang, J. (2022), "Applications of physics-informed neural networks in power systems-a review", *IEEE Transact. Power Syst*., **38**(1), 572-588. https://doi.org/10.1109/TPWRS.2022.3162473

Huang, H., Yuan, Y., Zhang, W. and Li, M. (2021a), "Seismic behavior of a replaceable artificial controllable plastic hinge for precast concrete beam-column joint", *Eng. Struct*., **245**, 112848. https://doi.org/10.1016/j.engstruct.2021.112848

Huang, H., Yuan, Y., Zhang, W. and Zhu, L. (2021b), "Property assessment of high-performance concrete containing three types of fibers", *Int. J. Concrete Struct. Mater*., **15**(1), 39.

https://doi.org/10.1186/s40069-021-00476-7
Huang, H., Guo, M., Zhang, W. and Huang, M. (2022a), "Seismic Behavior of Strengthened RC Columns under Combined Loadings", *J. Bridge Eng.*, **27**(6), 05022005. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001871
Huang, H., Li, M., Yuan, Y. and Bai, H. (2022b), "Theoretical analysis on the lateral drift of precast concrete frame with replaceable artificial controllable plastic hinges", *J. Build. Eng.*, **62**, 105386. https://doi.org/10.1016/j.jobe.2022.105386
Huang, H., Li, M., Zhang, W. and Yuan, Y. (2022c), "Seismic behavior of a friction-type artificial plastic hinge for the precast beam–column connection", *Arch. Civil Mech. Eng.*, **22**(4), 201. https://doi.org/10.1007/s43452-022-00526-1
Jiang, S., Zhao, C., Zhu, Y., Wang, C. and Du, Y. (2022), "A practical and economical ultra-wideband base station placement approach for indoor autonomous driving systems", *J. Adv. Transport.* https://doi.org/10.1155/2022/3815306
Kaveh, A. and Bakhshpoori, T. (2013), "Optimum design of space trusses using cuckoo search algorithm with levy flights". http://doi.org/10.22099/IJSTC.2013.945
Kumar, S., Datta, D. and Singh, S.K. (2015), "Black hole algorithm and its applications", In: *Computational intelligence applications in modeling and control*, pp. 147-170. https://doi.org/10.1007/978-3-319-11017-2_7
Latty, T. and Beekman, M. (2010), "Food quality and the risk of light exposure affect patch-choice decisions in the slime mold Physarum polycephalum", *Ecology*, **91**(1), 22-27. https://doi.org/10.1890/09-0358.1
Li, S., Chen, H., Wang, M., Heidari, A.A. and Mirjalili, S. (2020), "Slime mould algorithm: A new method for stochastic optimization", *Future Generat. Comput. Syst.*, **111**, 300-323. https://doi.org/10.1016/j.future.2020.03.055
Li, J., Chen, M. and Li, Z. (2022), "Improved soil–structure interaction model considering time-lag effect", *Comput. Geotech.*, **148**, 104835. https://doi.org/10.1016/j.compgeo.2022.104835
Li, Z., Gao, M., Lei, Z., Tong, L., Sun, J., Wang, Y., Wang, X. and Jiang, X. (2023a), "Ternary cementless composite based on red mud, ultra-fine fly ash, and GGBS: Synergistic utilization and geopolymerization mechanism", *Case Stud. Constr. Mater.*, **19**, e02410. https://doi.org/10.1016/j.cscm.2023.e02410
Li, Z., Kong, Y. and Jiang, C. (2023b), "A Transfer Double Deep Q Network Based DDoS Detection Method for Internet of Vehicles", *IEEE Transactions on Vehicular Technology*. https://doi.org/10.1109/TVT.2022.3233880
Lim, C.-H., Yoon, Y.-S. and Kim, J.-H. (2004), "Genetic algorithm in mix proportioning of high-performance concrete", *Cement Concrete Res.*, **34**(3), 409-420. https://doi.org/10.1016/j.cemconres.2003.08.018
Liu, C., Cui, J., Zhang, Z., Liu, H., Huang, X. and Zhang, C. (2021), "The role of TBM asymmetric tail-grouting on surface settlement in coarse-grained soils of urban area: Field tests and FEA modelling", *Tunnell. Undergr. Space Technol.*, **111**, 103857. https://doi.org/10.1016/j.tust.2021.103857
Liu, C., Peng, Z., Cui, J., Huang, X., Li, Y. and Chen, W. (2023), "Development of crack and damage in shield tunnel lining under seismic loading: Refined 3D finite element modeling and analyses", *Thin-Wall. Struct.*, **185**, 110647. https://doi.org/10.1016/j.tws.2023.110647
McCulloch, W.S. and Pitts, W. (1943), "A logical calculus of the ideas immanent in nervous activity", *Bull. Mathe. Biophys.*, **5**(4), 115-133. https://doi.org/10.1007/BF02478259
Mehrabi, M. and Moayedi, H. (2021), "Landslide susceptibility mapping using artificial neural network tuned by metaheuristic algorithms", *Environ. Earth Sci.*, **80**, 1-20. https://doi.org/10.1007/s12665-021-10098-7
Mirjalili, S. and Lewis, A. (2016), "The whale optimization algorithm", *Adv. Eng. Software*, **95**, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008
Mirjalili, S., Mirjalili, S.M. and Hatamlou, A. (2016), "Multi-verse optimizer: a nature-inspired algorithm for global optimization", *Neural Comput. Applicat.*, **27**(2), 495-513. https://doi.org/10.1007/s00521-015-1870-7
Moayedi, H. and Hayati, S. (2018a), "Applicability of a CPT-Based Neural Network Solution in Predicting Load-Settlement Responses of Bored Pile", *Int. J. Geomech.*, **18**(6), 06018009. http://doi.org/10.1061/(ASCE)GM.1943-5622.0001125
Moayedi, H. and Hayati, S. (2018b), "Modelling and optimization of ultimate bearing capacity of strip footing near a slope by soft computing methods", *Appl. Soft Comput.*, **66**, 208-219. https://doi.org/10.1016/j.asoc.2018.02.027
Moayedi, H. and Jahed Armaghani, D. (2018), "Optimizing an ANN model with ICA for estimating bearing capacity of driven pile in cohesionless soil", *Eng. Comput.*, **34**(2), 347-356. http://doi.org/10.1007/s00366-017-0545-7
Moayedi, H. and Mosavi, A. (2021), "Hybridizing neural network with multi-verse, black hole, and shuffled complex evolution optimizer algorithms predicting the dissolved oxygen". https://doi.org/10.20944/preprints202101.0464.v1
Moayedi, H., Huat, B.B., Ali, T.A.M., Asadi, A., Moayedi, F. and Mokhberi, M. (2011), "Preventing landslides in times of rainfall: case study and FEM analyses", *Disaster Prevent. and Manage.: Int. J.*, **20**(2), 115-124. http://doi.org/10.1108/09653561111126067
Moayedi, H., Kalantar, B., Foong, L.K., Tien Bui, D. and Motevalli, A. (2019), "Application of three metaheuristic techniques in simulation of concrete slump", *Appl. Sci.*, **9**(20), 4340. http://doi.org/10.3390/app9204340
Moayedi, H., Abdullahi, M.a.M., Nguyen, H. and Rashid, A.S.A. (2021a), "Comparison of dragonfly algorithm and Harris hawks optimization evolutionary data mining techniques for the assessment of bearing capacity of footings over two-layer foundation soils", *Eng. Comput.*, **37**(1), 437-447. http://doi.org/10.1007/s00366-019-00834-w
Moayedi, H., Osouli, A., Nguyen, H. and Rashid, A.S.A. (2021b), "A novel Harris hawks' optimization and k-fold cross-validation predicting slope stability", *Eng. Comput.*, **37**(1), 369-379. http://doi.org/10.1007/s00366-019-00828-8
Moayedi, H., Foong, L.K. and Le, B.N. (2023), "Three intelligent computational models to predict the high-performance concrete mixture", *Neural Comput. Applicat.*, 1-20. https://doi.org/10.1007/s00521-023-09233-1
Nasiri, J. and Khiyabani, F.M. (2018), "A whale optimization algorithm (WOA) approach for clustering", *Cogent Mathe. Statist.*, **5**(1), 1483565. https://doi.org/10.1080/25742558.2018.1483565
Nazir, R., Moayedi, H., Pratikso, A. and Mosallanezhad, M. (2015), "The uplift load capacity of an enlarged base pier embedded in dry sand", *Arab. J. Geosci.*, **8**, 7285-7296. https://doi.org/10.1007/s12517-014-1721-3
Ren, Z., Zeng, H., Zeng, X., Chen, X. and Wang, X. (2023), "Effect of nanographite conductive concrete mixed with magnetite sand excited by different alkali activators and their combinations on the properties of conductive concrete", *Buildings*, **13**(7), 1630. https://doi.org/10.3390/buildings13071630
Shehabeldeen, T.A., Elaziz, M.A., Elsheikh, A.H., Hassan, O.F., Yin, Y., Ji, X., Shen, X. and Zhou, J. (2020), "A novel method for predicting tensile strength of friction stir welded AA6061 aluminium alloy joints based on hybrid random vector functional link and henry gas solubility optimization", *IEEE Access*, **8**, 79896-79907. https://doi.org/10.1109/ACCESS.2020.2990137
Shi, T., Liu, Y., Hu, Z., Cen, M., Zeng, C., Xu, J. and Zhao, Z.

(2022), "Deformation performance and fracture toughness of carbon nanofiber-modified cement-based materials", ACI Materials Journal, **119**(5), 119-128. https://doi.org/10.14359/51735976

Shi, M.-L., Lv, L. and Xu, L. (2023), "A multi-fidelity surrogate model based on extreme support vector regression: fusing different fidelity data for engineering design", *Eng. Computat.*, **40**(2), 473-493. https://doi.org/10.1108/EC-10-2021-0583

Singh, A., Wang, Y., Zhou, Y., Sun, J., Xu, X., Li, Y., Liu, Z., Chen, J. and Wang, X. (2023), "Utilization of antimony tailings in fiber-reinforced 3D printed concrete: A sustainable approach for construction materials", *Constr. Build. Mater.*, **408**, 133689. https://doi.org/10.1016/j.conbuildmat.2023.133689

Sun, L., Wang, C., Zhang, C., Yang, Z., Li, C. and Qiao, P. (2022), "Experimental investigation on the bond performance of sea sand coral concrete with FRP bar reinforcement for marine environments", *Adv. Struct. Eng.*, **26**(3), 533-546. https://doi.org/10.1177/13694332221131153

Tien Bui, D., Abdullahi, M.a.M., Ghareh, S., Moayedi, H. and Nguyen, H. (2021), "Fine-tuning of neural computing using whale optimization algorithm for predicting compressive strength of concrete", *Eng. Comput.*, **37**(1), 701-712. https://doi.org/10.1007/s00366-019-00850-w

Trivedi, I.N., Jangir, P., Kumar, A., Jangir, N. and Totlani, R. (2018), *Advances in computer and computational sciences*, Springer, pp. 53-60.

Wang, M., Yang, X. and Wang, W. (2022), "Establishing a 3D aggregates database from X-ray CT scans of bulk concrete", *Constr. Build. Mater.*, **315**, 125740. https://doi.org/10.1016/j.conbuildmat.2021.125740

Wenjun, D., Fatahizadeh, M., Touchaei, H.G., Moayedi, H. and Foong, L.K. (2023), "Application of six neural network-based solutions on bearing capacity of shallow footing on double-layer soils", *Steel Compos. Struct., Int. J.*, **49**(2), 231-244. https://doi.org/10.12989/scs.2023.49.2.231

Yıldız, B.S., Yıldız, A.R., Pholdee, N., Bureerat, S., Sait, S.M. and Patel, V. (2020), "The Henry gas solubility optimization algorithm for optimum structural design of automobile brake components", *Mater. Test.*, **62**(3), 261-264. https://doi.org/10.3139/120.111479

Zhang, Y. and Zhang, H. (2023), "Enhancing Robot Path Planning through a Twin-Reinforced Chimp Optimization Algorithm and Evolutionary Programming Algorithm", *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3337602

Zhang, Z., Li, W. and Yang, J. (2021), "Analysis of stochastic process to model safety risk in construction industry", *J. Civil Eng. Manage.*, **27**(2), 87-99. https://doi.org/10.3846/jcem.2021.14108

Zhang, Y., Gono, R. and Jasiński, M. (2023), "An Improvement in Dynamic Behavior of Single Phase PM Brushless DC Motor Using Deep Neural Network and Mixture of Experts", *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3289409

Zhang, Y., Abdullah, S., Ullah, I. and Ghani, F. (2024), "A new approach to neural network via double hierarchy linguistic information: Application in robot selection", *Eng. Applicat. Artif. Intell.*, **129**, 107581. https://doi.org/10.1016/j.engappai.2023.107581

Zhao, Y., Dai, W., Wang, Z. and Ragab, A.E. (2023), "Application of computer simulation to model transient vibration responses of GPLs reinforced doubly curved concrete panel under instantaneous heating", *Mater. Today Commun.*, 107949. https://doi.org/10.1016/j.mtcomm.2023.107949

Zhou, S., Lu, C., Zhu, X. and Li, F. (2021), "Preparation and Characterization of High-Strength Geopolymer Based on BH-1 Lunar Soil Simulant with Low Alkali Content", *Engineering*, **7**(11), 1631-1645. https://doi.org/10.1016/j.eng.2020.10.016

Zhou, F., Jiang, H., Huang, L., Hu, Y., Xie, Z., Zeng, Z., Liu, M., Wang, B. and Zhou, X. (2023a), "Early shrinkage modeling of complex internally confined concrete based on capillary tension theory", *Buildings*, **13**(9), 2201.

Zhou, F., Li, W., Hu, Y., Huang, L., Xie, Z., Yang, J., Wu, D. and Chen, Z. (2023b), "Moisture diffusion coefficient of concrete under different conditions", *Buildings*, **13**(10), 2421. https://doi.org/10.3390/buildings13102421

Zhu, X., Gao, Y., Dai, Z., Corr, D.J. and Shah, S.P. (2018), "Effect of interfacial transition zone on the Young's modulus of carbon nanofiber reinforced cement concrete", *Cement Concrete Res.*, **107**, 49-63. https://doi.org/10.1016/j.jobe.2022.105248