Smart Structures and Systems, *Vol. 26, No. 3 (2020) 263-275*
DOI: https://doi.org/10.12989/sss.2020.26.3.263

Metaheuristic-hybridized multilayer perceptron in slope stability analysis

Xinyu Ye [1a], Hossein Moayedi [2,3b], Mahdy Khari [4c] and Loke Kok Foong [*5,6]

[1] *School of civil engineering, Central South University, Changsha 410075, China*
[2] *Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam*
[3] *Faculty of Civil Engineering, Duy Tan University, Da Nang 550000, Vietnam*
[4] *Department of Civil Engineering, East Tehran Branch, Islamic Azad University, Tehran, Iran*
[5] *Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam*
[6] *Faculty of Civil Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam*

(*Received December 13, 2019, Revised May 26, 2020, Accepted May 30, 2020*)

Abstract. This research is dedicated to slope stability analysis using novel intelligent models. By coupling a neural network with spotted hyena optimizer (SHO), salp swarm algorithm (SSA), shuffled frog leaping algorithm (SFLA), and league champion optimization algorithm (LCA) metaheuristic algorithms, four predictive ensembles are built for predicting the factor of safety (FOS) of a single-layer cohesive soil slope. The data used to develop the ensembles are provided from a vast finite element analysis. After creating the proposed models, it was observed that the best population size for the SHO, SSA, SFLA, and LCA is 300, 400, 400, and 200, respectively. Evaluation of the results showed that the combination of metaheuristic and neural approaches offers capable tools for estimating the FOS. However, the SSA (error = 0.3532 and correlation = 0.9937), emerged as the most reliable optimizer, followed by LCA (error = 0.5430 and correlation = 0.9843), SFLA (error = 0.8176 and correlation = 0.9645), and SHO (error = 2.0887 and correlation = 0.8614). Due to the high accuracy of the SSA in properly adjusting the computational parameters of the neural network, the corresponding FOS predictive formula is presented to be used as a fast yet accurate substitution for traditional methods.

Keywords: geotechnical engineering; slope stability analysis; neural computing; metaheuristic optimizer

1. Introduction

Accurate assessment of the stability of slopes is a vital step in many civil/geotechnical engineering projects. Up to now, various analytical and numerical methods have been suggested for this task (Ng and Shi 1998, Cai and Ugai 2004, Zhang and Zhou 2018, Wang *et al*. 2019). Dai *et al*. (2008) conducted a numerical approach (by employing a finite element and a fast Lagrangian finite difference method) for analyzing the soil slope stability by taking into equation the tension and shear failures. Pradatta *et al*. (2018) presented an analytical technique for stability analysis of cohesionless soil slopes under combined horizontal and vertical seismic loads. Camargo *et al*. (2016) employed an effective numerical solution, namely numerical limit analysis to a 3D problem. They applied the proposed methodology to a real-world catchment affected by intense rainfall. Lim *et al*. (2016) used finite element limit analysis methods (i.e., the upper and lower bound solutions) to investigate the stability of soil and rock slopes. (Leshchinsky and Ambauen 2015) stated the applicability of upper bound limit analysis associated with discontinuity layout optimization. This method can be effectively used for stability evaluation and analyzing the failure mechanism, regardless of the constraints and assumptions needed in limit equilibrium methods.

Soft computing methods have been successfully used in various research projects in different fields of geotechnical engineering (Bagheri Sereshki and Derakhshani 2018, Qiao *et al*. 2020, Zhou *et al*. 2020). Different examples of soft computing are utilized for soil compression coefficient (Moayedi *et al*. 2019b, 2020a), landslide susceptibility assessment (Bui *et al*. 2019c, Nguyen *et al*. 2019), slope stability (Bui *et al*. 2019a, Moayedi *et al*. 2019c). Some studies have specifically addressed the shallow foundation problems such as predicting the ultimate bearing capacity and settlement (Barari *et al*. 2015, Kohestani *et al*. 2017, Mosallanezhad and Moayedi 2017, Moayedi and Hayati 2018). A review of the soft computing research programs regarding the ultimate bearing capacity is provided in the next section. In recent years, machine learning methods such as artificial neural network (ANN), genetic programming, bacterial foraging optimization (Xu and Chen 2014, Chen *et al*. 2020), improved ant colony optimization (Zhao *et al*. 2014), fruit fly optimization (Shen *et al*. 2016), chaotic moth-flame optimization (Wang *et al*. 2017), Moth-flame optimizer (Xu *et al*. 2019), grey wolf optimization (Zhao *et al*. 2019), multi-swarm whale optimizer (Wang and Chen 2020), etc. have become popular among the scholars because they can make predictions with

∗Corresponding author, Ph.D.,
E-mail: lokekokfoong@tdtu.edu.vn
[a] Ph.D., E-mail: yexinyu113@csu.edu.cn
[b] Ph.D., E-mail: hosseinmoayedi@duytan.edu.vn
[c] Ph.D., E-mail: mehdikhari@gmail.com

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

optimal accuracy when modeling complicated phenomena. The invention of soft computing has suggested efficient predictors (like fuzzy-based tools) which have been useful in many engineering and medical fields. ANNs are known as potent predictive methods that mimic the relationships between the components of a real neural system (McCulloch and Pitts 1943). The most notable option of this tool is the non-linear perception of every complex phenomenon including plenty of engineering parameters (Zounemat-Kermani *et al*. 2016, Ghiasi and Ghasemi 2018, Keshavarz and Torkian 2018, Li *et al*. 2018, Onat and Gul 2018). Basically, the ANNs benefit the backpropagation (BP) (Hecht-Nielsen 1992) scheme to map the relationship between a group of input-output data. In fact, a number of mathematical equations are established between so-called processors "neurons" by using connecting weights. The neural equation is then completed by adding a threshold term as well as applying an activation function. Many studies have investigated the efficacy of artificial intelligence models for slope stability analysis (Sakellariou and Ferentinou 2005, Choobbasti *et al*. 2009, Li *et al*. 2016, Akin and Sahin 2017, Chahnasir *et al*. 2018, Fallahian *et al*. 2018).

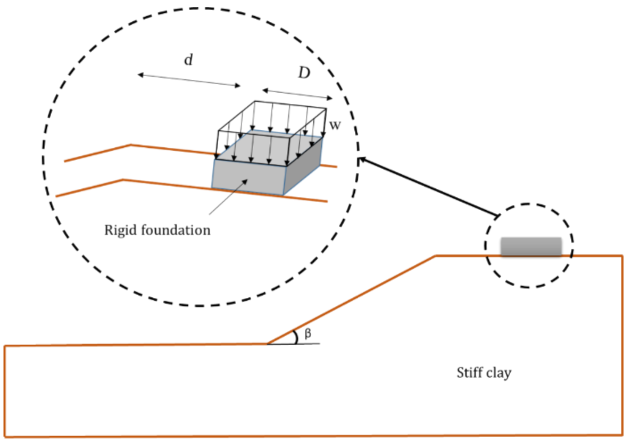
Despite the wide application of ANNs, these approaches are not so resistant against computational drawbacks like local minima. Therefore, hybrid metaheuristic algorithms have been suggested for optimization aims. Gandomi *et al*. (2015) and (2017) used various evolutionary techniques for slope stability optimization and proved the high efficiency biogeography-based optimization and flight krill herd for slope stability analysis. Scholars like Gordan *et al*. (2016) and Li *et al*. (2015) have focused on the usefulness of particle swarm optimization in this field. Moayedi *et al*. (2019a) used Harris hawks optimization (HHO) technique to overcome the computational shortcomings of a multi-layer neural network. The proposed algorithm reduced the prediction error by nearly 27% and increased the correlation between the real and forecasted FOSs from 0.8220 to 0.9253. Similar methodologies were applied to the problem of landslide susceptibility modeling carried out for different regions in Iran (Moayedi *et al*. 2018).

The literature review indicates the wide application of well-known optimization techniques used for enhancing the competency of machine learning methods like ANNs in the field of slope stability assessment (Bui *et al*. 2019b, Yuan and Moayedi 2019). But it is felt that employing more state-of-the-art metaheuristic techniques can lead to more capable predictors for dealing with the mentioned problem. Hence, four wise optimization algorithms, namely spotted hyena optimizer (SHO), salp swarm algorithm (SSA), shuffled frog leaping algorithm (SFLA), and league champion optimization algorithm (LCA) are proposed in this study to create neural ensembles for accurate appraisement of the FOS. Meanwhile, the best model is distinguished based on an accuracy evaluation process.

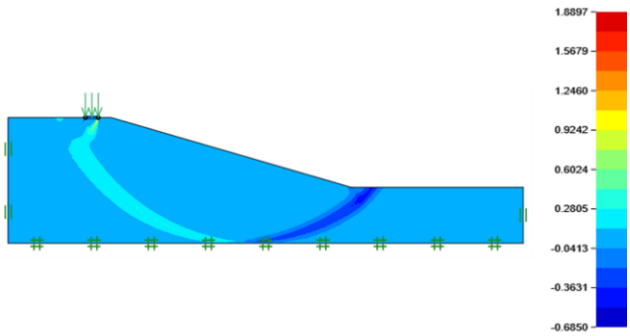
2. Methodology and established database

Needless to say, every intelligent model needs to be fed by proper data (in both classification and regression cases). These data can be derived from different ways like laboratory tests and experimental approaches, real-world observation, analytical computer software, etc. The soil data used to train and evaluate the intelligent models of this study are obtained from a finite element analysis carried out in Optum G2 software (Krabbenhoft *et al*. 2015). In this modeling, the FOS of a single-layer cohesive soil slope (with mechanical factors of Poisson ratio = 0.35, internal friction angle = 0°, and soil unit weight = 18 kN/m^3) is calculated by taking into consideration the effect of four variables, namely undrained cohesive strength (C_u), slope angle (β), the surcharge on the footing (w), and the ratio of setback distance (d/D). An illustration of the modeled system can be seen in Fig. 1(a). Also, an example of the Optum G2 analysis is shown in this figure.

A total of 630 stages were analyzed and the values of input factors (i.e., d/D, C_u, β, and w) along with the obtained FOS create the dataset. Fig. 2 depicts the relationship between the input and output factors. Out of this information, regarding the popular ratio of 80:20, the predictive models use 504 samples to infer the relationship



(a)



(b)

Fig. 1 A schematic vision of (a) the designed slope in the reality; and (b) Optum G2 analysis (horizontal strain diagram for d/D = 1, Cu = 75 kPa, β = 15°, and w = 100 KN/m²)

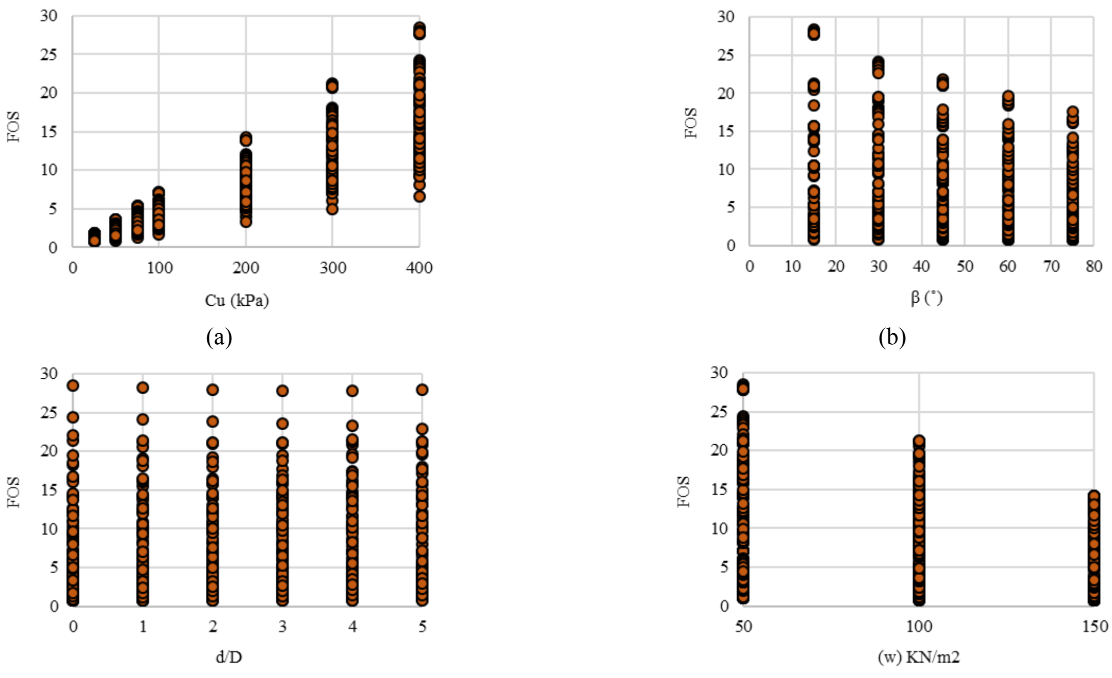


Fig. 2 The graphical description of the input/output factors

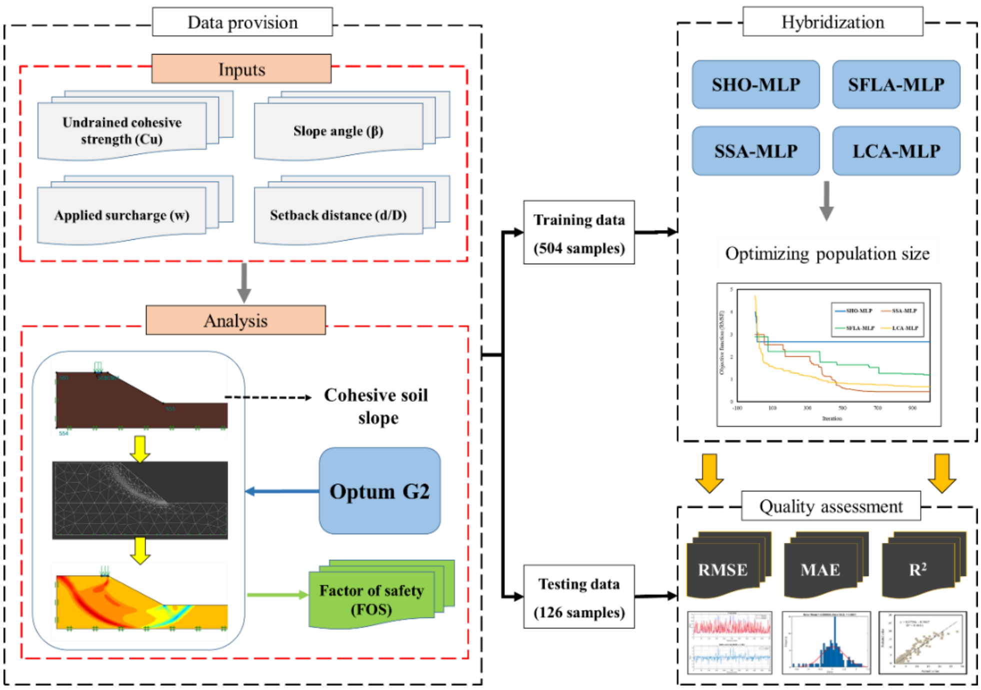


Fig. 3 The methodology of the study

between the mentioned parameters. Then, the trained models are applied to the remaining 126 samples for evaluating their prediction capability for unseen slope conditions.

2.1 Methodology

The graphical description of the methodology (i.e., the implemented steps) applied to achieve the goal of the study is illustrated in Fig. 3. After preparing a suitable database from finite element approaches, it is divided into the training and testing phases. Next, the hybridization process (combining the metaheuristic algorithms with the ANN) is implemented which is the main core of this work. The next step is optimizing the created models in terms of their population size. For this purpose, the convergence behavior of each algorithm is evaluated to ensure that it is being executed in the optimal conditions. The neural ensemble models then perform to predict the FOS, and lastly, their performance is validated by proper criteria.

2.1.1 Used metaheuristic schemes

SHO: The spotted hyena optimizer is a recently developed metaheuristic algorithm which is designed by Dhiman and Kumar (2018) for finding the optimal solutions to engineering problems. The high convergence speed can be noted as an advantage of the SHO. The social behavior of spotted hyenas for hunting a prey (e.g., a zebra) comprises several stages during which the individuals try to update their locations. These stages are described as follows:

a) Encircling prey

Spotted hyenas (SHs) can track the location of their prey for surrounding it. The elite candidate is the SH that is closest to the target. After defining the best solution, other agents update their locations. Mathematically

$$\vec{D}_h = \left|\vec{B} \, . \vec{P}_p(x) - \vec{P}(x)\right| \tag{1}$$

$$\vec{P}(x+1) = \vec{P}_p(x) - \vec{E} \, . \vec{D}_h \tag{2}$$

where X shows the current iteration, $\vec{D}_h$ represents the distance between the SH and the target, $\vec{B}$ and $\vec{E}$ are coefficient vectors. Also, $\vec{P}$ and $\vec{P}_p$ indicate the position of the SH and prey, respectively. The coefficient vectors are defined as follows

$$\vec{B} = 2 \, . r\,\vec{d}_1 \tag{3}$$

$$\vec{E} = 2\vec{h} \, . r\,\vec{d}_2 - \vec{h} \tag{4}$$

$$\vec{h} = 5 - \left(It \times \frac{5}{It_{max}}\right), \quad It = 1, 2, .., It_{max} \tag{5}$$

where It is the iteration. As this parameter increases, the $\vec{h}$ falls from 5 to 0 linearly to properly balance the exploration and exploitation. $r\,\vec{d}_1$ and $r\,\vec{d}_2$ symbolize random vectors in [0, 1].

b) Hunting

In the hunting process, it is assumed that the elite agents know the target location. Other agents try to update their locations accordingly. Assuming $\vec{P}_h$ and $\vec{P}_k$ as the position of the first elite SH and other SHs, respectively, we can write

$$\vec{D}_h = \left|\vec{B} \, . \vec{P}_h - \vec{P}_k\right| \tag{6}$$

$$\vec{P}_k = \vec{P}_h - \vec{E} \, . \vec{D}_h \tag{7}$$

$$\vec{C}_h = \vec{P}_k + \vec{P}_{k+1} + \cdots + \vec{P}_{k+N} \tag{8}$$

The parameter N defines the number of individuals that is obtained from the below relationship

$$N = count_{nos}(\vec{P}_h, \vec{P}_{h+1}, \dots, (\vec{P}_h + \vec{M})) \tag{9}$$

in which $\vec{M}$ is a random vector that its value may range from 0.5 to 1, *nos* stands for the number of solutions. The term $\vec{C}_h$ is a cluster of N number of optimal solutions.

c) Attacking target (Exploitation)

For doing this stage, the value of $\vec{h}$ is decreased. For this decrease, the variation in $\vec{E}$ is also reduced. The group of SHs attack target when $\left|\vec{E}\right| < 1$. The attacking procedure can be expressed as follows

$$\vec{P}(x+1) = \frac{\vec{C}_h}{N} \tag{10}$$

where the outcome of this relationship (i.e., $\vec{P}(x+1)$) adjusts the position of other SHs, based on the position of the elite one.

d) As explained, the SHs seek the target regarding the position of the SHs in vector $\vec{C}_h$. Once $\left|\vec{E}\right| > 1$, the SHs leave the prey. It leads the algorithm toward a global optimization. The exploration is carried out by assist of the vector $\vec{B}$. This component contains the random weight of the target that is produced in Eq. (10). This vector effectively helps the SHO to avoid local optimizations. The pseudocode of the SHO is presented below (Jia *et al.* 2019)

Algorithm 1: The pseudo-code of the SHO technique

Take the population (P_i, $i = 1,2,\dots,n$) as the input and show the elite search agent.

Initialize the SHO parameters h, N, E, and B
Compute the fitness of the search agents
$P_h = the\ elite\ search\ agent$
$C_h = the\ cluster\ of\ optimal\ solutions$
while (the number of iterations) **do**
for each search agent **do**
Update the position of the present SH by Eq. (3)
end for
Update h, N, E, and B
Adjust the agents gone behind the given space (if any)
Calculate the fitness of each agent
Update the elite SH if a better solution is found
Update the cluster C_h regarding the elite individual
$x = x + 1$
end while
Return P_h
End

More information about this algorithm can be found in Balasubbareddy *et al.* (2019), Dhiman (2019), Divya *et al.* (2020).

SSA: The salp swarm algorithm is proposed by Mirjalili *et al.* (2017), based on the foraging behavior of salps (a member of Salpidae family living in oceans). Similar to many other algorithms, the SSA is a population-based method in which the possible solutions are represented by salp individuals. The members in each salp chain are followers guided by a so-called member "leader" to find food sources (FSs).

The below matrix represents a group of k salps

$$X_i = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^k & x_2^k & \dots & x_d^k \end{bmatrix} \tag{11}$$

Given FS_j as the position of the FS in the dimension j, the position of the leader (x_j^1) is adjusted based on the following equation

$$x_j^1 = \begin{cases} FS_j + C_1\left(\left(ub_j - lb_j\right)C_2 + lb_j\right) & C_3 \geq 0.5 \\ FS_j - C_1\left(\left(ub_j - lb_j\right)C_2 + lb_j\right) & C_3 < 0.5 \end{cases} \tag{12}$$

where C_2 and C_3 are values randomly selected in [0, 1] that play a significant role in directing the coming positions (towards$+\infty$ or $-\infty$) and determining the step size. Also, ub_j and lb_j denote the upper bound and lower bound of the dimension. C_1 is a variable which is gently reduced over the process. Eq. (13) gives this parameter

$$C_1 = 2e^{-\left(\frac{4It}{It_{max}}\right)^2} \tag{13}$$

where the present iteration and the maximum number of iterations are shown by It and It_{max}, respectively.

Finally, the follower salps use Eq. (14) to update their positions

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right) \tag{14}$$

in which i ≥ 2. The pseudo-code of the SSA algorithm is presented in Algorithm 2 (Faris *et al*. 2018).

Algorithm 2: The pseudo-code of the SSA technique

Take the population ($P_i,\ i = 1,2,\dots,n$) as the input and show the elite search agent.

Initialize the salp population with respect to lower bound and upper bound
while (ending criterion is not met) **do**
Calculate the fitness of each agent
Set FSs the elite agent
Update c_1 by Eq. (13)
for (each search agent) **do**
if (i ==1) **then**
Update the leading salp's position by Eq. (12)
else
Update the follower salps' position by Eq. (14)
Update the population with respect to ub and lb of variables
Return *FS*

The SSA is further detailed in Hussien *et al*. (2017), Aljarah *et al*. (2018), Ibrahim *et al*. (2019).

SFLA: The shuffled frog leaping algorithm is one of the most popular search techniques introduced by Eusuff and Lansey (2003). As the name connotes, the individuals in this algorithm are a series of frogs where each one represents a solution. This algorithm presents a combination of the PSO and Memetic algorithm based genetic algorithm (Kimiyaghalam *et al*. 2012).

Similar to other algorithms which are inspired by animals' behavior, updating the frogs' positions is the main idea of the SFLA. To do this, a fitness value is assigned to each member to classify them in a number of containers named memeplexes (Chen *et al*. 2019).

Updating the frogs' positions can be expressed by the below relationships

$$X_{new} = X_w + S \tag{15}$$

where X_w symbolizes the worst frog's position and S is calculated as follows

$$S = rand() \times (X_b - X_w) \tag{16}$$

In the above equation, X_b is the frog's best position and *rand* () gives a random value between 0 and 1. Note that *S* can receive values between S_{max} (i.e., the maximum leap) and -S_{max}. Algorithm 3 gives the pseudocode of the SFLA (Supraja and Jayashri 2017)

Algorithm 3: The pseudo-code of the SFLA technique

Start
Initialize a random population of frogs
Calculate the fitness of each agent
Sort the frogs based on their fitness values
Divide the population into m memeplexes
for each memeplex **do**
Determine the best and worst member
Improve the position of the worst member using Eq. (15)
Repeat this for the given number of iterations
end for
Merge the evolved memeplexes
Sort the frogs regarding their descending fitness vales
Check if stopping criterion = true
End

The SFLA is also explained in the previous literature (Liping *et al*. 2012, Zhang *et al*. 2012, Roy *et al*. 2013).

LCA: Kashan (2009) presented the league champion optimization inspired by the competitions in an artificial league. Utilizing a single round-robin algorithm, the main schedule is developed for a season. If there are K teams in the league, a total of $K \times (K - 1) / 2$ matches are supposed to be held. Actually, there are $K - 1$ weeks in each season (S) that results in $S \times (K - 1)$ weeks of contests. This is worth noting that in order to donate rest to a team, a dummy team is taken into account when K is an odd number.

Regarding the playing strength of the attending teams, the chance of winning/losing is determined for them in the matches. More clearly, the team with larger playing strength is more likely to win the match. Based on this idealized rule (Kashan 2014)

$$\frac{f(X_i^t) - \hat{f}}{f(X_j^t) - \hat{f}} = \frac{P_j^t}{P_i^t} \tag{17}$$

where X_i^t and X_j^t define the formations, $f(X_i^t)$ and $f(X_j^t)$ are the playing strengths of the teams and the chance of team i for defeating the opponent is shown by P_i^t. Also, f represents a function that is projected to be minimized.

Based on another idealized rule of the LCA algorithm it can be written that

$$P_j^t + P_i^t = 1 \tag{18}$$

Hence, for a match between two teams i and j at the t^{th} week, the parameter P_i^t can be calculated by Eq. (19) (Moayedi *et al*. 2020b)

$$P_i^t = \frac{f(X_j^t) - \hat{f}}{f(X_j^t) + f(X_i^t) - 2\hat{f}} \tag{19}$$

The pseudo-code of the LCA can be found in Algorithm 4 (Bozorg-Haddad 2018).

Algorithm 4: The pseudo-code of the LCA technique

Start
Initialize the random formation of the teams
Generate league schedule for K teams
for $m = 1: K(S-1)$
Evaluate the strength of the teams
Compute the chance of each to beat its rival in the coming competition (P_i^t)
Generate a random value in [0, 1] (R_n)

If $R_n \leq P_i^t$
Team i is the winner
Else
Team j is the winner
End if

Generate a random value in [0,1] (r)
Calculate the number of changes in teams' best formation (B_i^t) for the next competition with respect to the truncated geometric distribution (q_i^t)
q_i^t players are randomly chosen (from B_i^t) and changed by the strength/weakness/ opportunity/threat (SWOT) matrix

If team i and team l are the winners
Select the S/T strategy
Else if team i is the winner and team l is the loser
Select the S/O strategy
Else if team i is the loser and team l is the winner
Select the W/T strategy
Else if both teams are losers
Select the W/O strategy

End if
end for
End

The LCA algorithm is better explained in Kashan (2011, 2014), Jalili *et al*. (2016).

2.1.2 Ensemble development and optimization

The hybridization process of the ANN is explained in this section. It consists of four steps including (a) obtaining the best basic structure of the ANN, (b) creating the equation of the found network and introducing it (as the problem function) to the optimizers, (c) executing the optimizers for finding the best parameters (i.e., the connecting weights and biases) for the network in a repetitive process, and (d) picking the most promising response regarding the stopping criteria. More clearly, the algorithms try to update the positions of their individuals (e.g., the frogs in the SFLA) to improve the goodness of their responses. It usually continues until it reaches a defined number of iterations or a specific fitness value.

The ANN proposed in this study is represented by an MLP (Hornik *et al*. 1989), a well-known notion of feedforward networks. The number of hidden neurons is one of the most influential variables the optimal value should be determined for that. This item was handled by trial and error efforts and it was shown that among ten tested values (1, 2, …, 10), the best MLP is which contained five hidden neurons. Also, Tansig was selected as the activation function of the hidden layer. Next, the general equation of the model is given to the SHO, SSA, SFLA, LCA algorithms to create SHO-MLP, SSA-MLP, SFLA-MLP, and LCA-MLP ensembles. In this task, the variables are the MLP computational parameters which are aimed to be optimized based on the relationship between the FOS with d/D, C_u, β, and w. Thus, the suggested solution is a matrix containing new weights and biases for constructing an MLP. The search process is performed for 1000 times (i.e., iterations). To evaluate the quality of the optimization, an objective function is defined to measure the accuracy at each iteration. This function is set to be root mean square error (RMSE) in this work (see Eq. (20)).

$$RMSE = \sqrt{\frac{1}{K}\sum_{i=1}^{K}[(O_{i_{observed}} - O_{i_{predicted}})]^2} \tag{20}$$

where $O_{i\ predicted}$ and $O_{i\ observed}$ represent the forecasted and real values of FOS and K symbolizes the number of samples.

Moreover, the number of involved individuals has a significant impact on the optimization results. This is why this parameter needs to be optimized. To this end, all four ensembles are implemented with nine different population

Table 1 The results of the sensitivity analysis

N_p	SHO-MLP		SSA-MLP		SFLA-MLP		LCA-MLP	
	RMSE	Time (*s*)	RMSE	Time (*s*)	RMSE	Time (*s*)	RMSE	Time (*s*)
10	3.322995	146.3396	0.865851	135.9052	1.255943	218.4249	1.022846	140.6084
25	3.373912	354.1796	0.832654	322.3489	1.288907	213.3355	0.951905	364.9705
50	3.488242	704.2673	0.540544	643.6628	1.668806	212.1432	0.790649	685.8581
75	3.332569	1112.581	0.523702	971.0438	1.225821	218.2488	0.84658	985.3211
100	3.380597	1301.352	0.508214	1404.67	1.270973	226.041	0.87237	1297.818
200	3.24693	2579.889	0.506516	2760.106	1.27593	230.4944	0.67041	2583.583
300	2.672776	4128.133	0.498274	4128.397	1.491712	263.4617	0.835979	3869.399
400	3.053384	5676.226	0.44636	5509.48	1.196237	264.7039	0.907868	14488.1
500	3.082803	6915.045	0.492831	6571.096	1.253728	279.0669	0.725523	6260.833

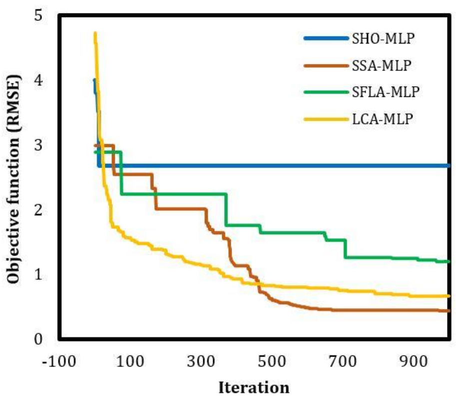


Fig. 4 The convergence curves of the elite models

sizes (N_ps), varying from 10 to 500. The results (the obtained RMSEs and the computation time) are available in Table 1 with a color intensity system. As is seen, the SHO, SSA, SFLA and LCA algorithms present the best optimization of the MLP with the population sizes of 300, 400, 400, and 200, respectively. Additionally, the convergence curves belonging to the best-fitted ensembles are illustrated in Fig. 4.

3. Results and discussion

3.1 Quality assessment indices

As well as the RMSE, mean absolute error (MAE) of the prediction results is calculated for both training and testing data. Moreover, the correlation between the real and calculated FOSs is reported by the coefficient of determination (R^2). As is known, the ideal value for the RMSE and MAE is zero, while the R^2 can range from 0 to 1 indicating the lowest and highest possible correlation, respectively. Assuming $\overline{O}_{observed}$ as the average of $O_{i\,observed}$, Eqs. (21) and (22) formulate the MAE and R^2.

$$MAE = \frac{1}{K}\sum_{i=1}^{K}|O_{i_{observed}} - O_{i_{predicted}}| \tag{21}$$

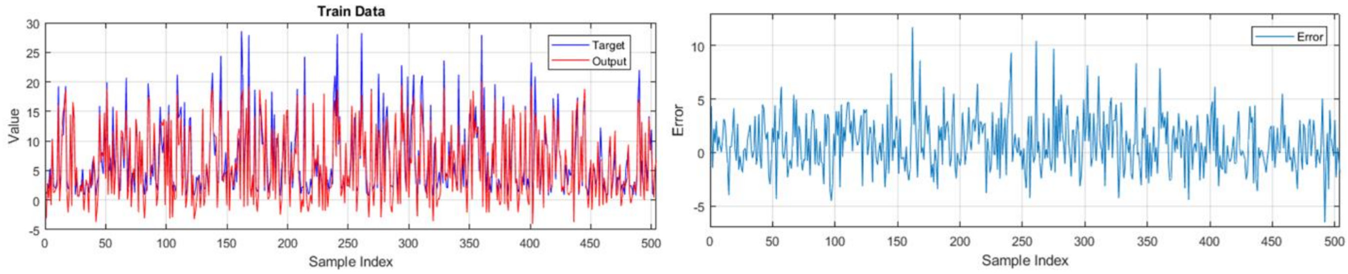
$$R^2 = 1 - \frac{\sum_{i=1}^{K}(O_{i_{predicted}} - O_{i_{observed}})^2}{\sum_{i=1}^{K}(O_{i_{observed}} - \overline{O}_{observed})^2} \tag{22}$$

3.2 Quality assessment and comparison

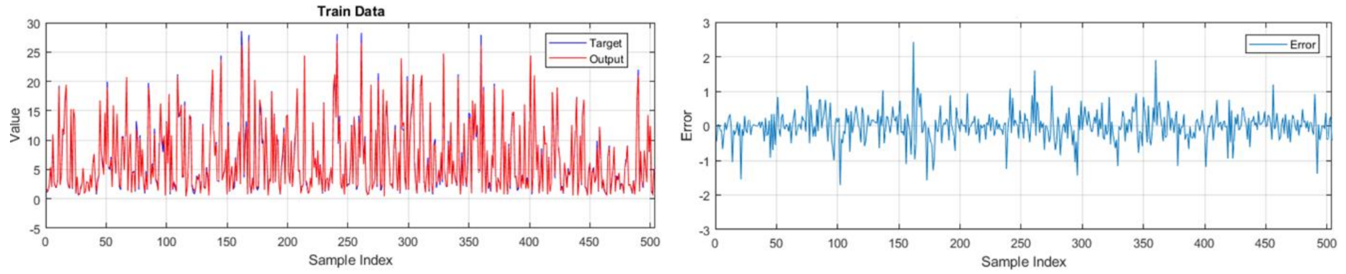
Utilizing the RMSE, MAE, and R^2 indices for comparing the real and forecasted FOSs, the performances of the SHO-MLP, SSA-MLP, SFLA-MLP, and LCA-MLP are evaluated. These criteria are once applied to the training data for assessing the learning potency of the models, as well as the testing data for evaluating the prediction ability. In fact, the neural-metaheuristic FOS pattern that is derived from the training data is applied to unseen slope circumstances.

Fig. 5 shows the training results. In this figure, the target FOSs (i.e., the real values) are compared with the forecasted ones. The pure error (= real value – forecasted value) of each sample is also depicted. As is seen, the FOS pattern is properly understood by all four models. The extent of learning errors for the the SHO-MLP, SSA-MLP, SFLA-MLP and LCA-MLP models is [-6.1336, 4.3488], [-1.1401, 1.5463], [-0.9040, 3.5451], and [-1.4348, 1.7213], respectively. Statistically, the RMSEs (2.6728, 0.4463, 1.1962, and 0.6704) and MAEs (2.0501, 0.3108, 0.8118, and 0.5172) obtained in this phase indicate that the FOS analysis carried out by the models is associated with an acceptable error. Moreover, considering the accommodation of the results, the calculated values of R^2 report 0.8368, 0.9947, 0.9621, and 0.9880 correlation between the outputs and target data.

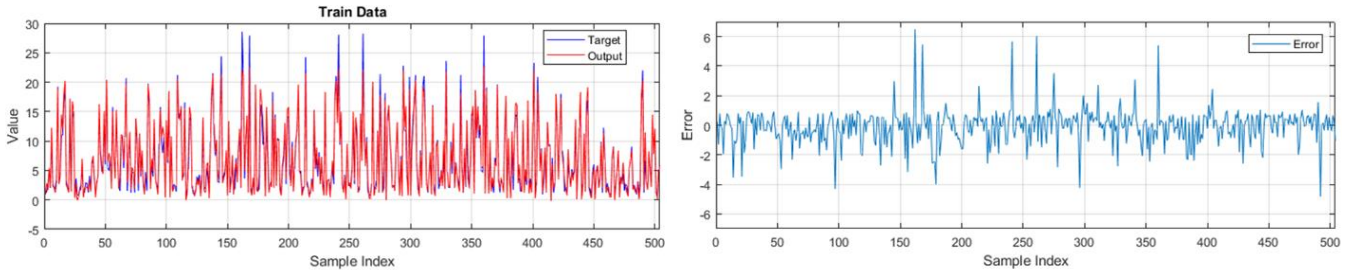
The results of the testing phase also indicate a high generalization efficiency for the models. Fig. 6 displays the correlation between the real and forecasted FOSs, along with the histogram of the errors. According to these charts, the values of the R^2 index show higher than 96 % accuracy for the SSA-MLP, SFLA-MLP, and LCA-MLP, while it is 0.8614 for the SHO-MLP prediction. Besides, the calculated RMSE of 2.6728, 0.4463, 1.1962, and 0.6704, as well as the MAEs of 2.0501, 0.3108, 0.8118, and 0.5172, reflect a good performance of the used ensembles for unseen conditions of the problem.



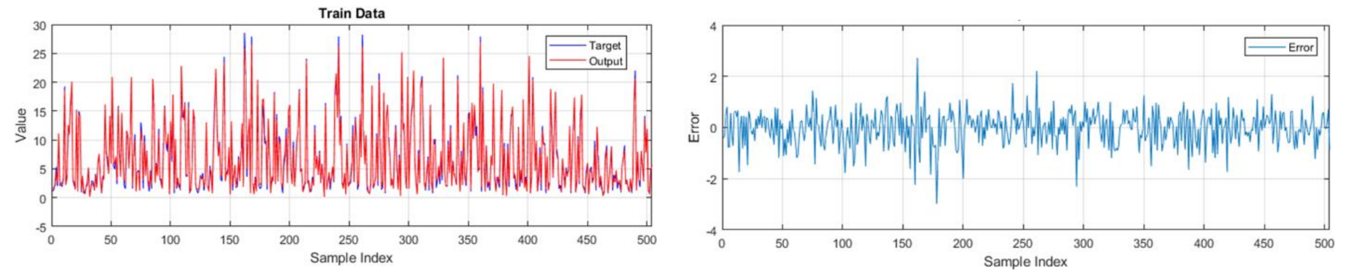
(a)



(b)



(c)



(d)

Fig. 5 The training results for the (a) SHO-MLP; (b) SSA-MLP; (c) SFLA-MLP; and (d) LCA-MLP prediction

3.3 Performance comparison

In this section, it is aimed to determine the most reliable metaheuristic algorithm (among used ones) for optimizing the ANN in analyzing the stability of single-layer soil slopes. To meet this goal, the models are assessed by taking into consideration the used accuracy criteria (i.e., the RMSE, MAE, and R^2 in Table 2) in both training and testing phases. A ranking score (between 1 to 4) is assigned to each model and the overall grade (OG) is calculated as the summation of them. Then, the models are ranked based on the resulted OGs. The results of this process are shown in Table 3. According to this table, the predictive model that is based on the SHO algorithm has gained the smallest score (i.e., 1) in all cells. After that, SFLA-based MLP has received a partial score of 2 in terms of all three indices. The scores of the MLP optimized by the LCA are 3 which indicates its better performance compared to two earlier optimizers. The largest score (i.r., 4) is obtained for the SSA-MLP in all cells which reflects the superiority of the

Table 2 The obtained values of RMSE, MAE, and R^2

Models	Network results					
	Training			Testing		
	RMSE	MAE	R^2	RMSE	MAE	R^2
SHO-MLP	2.6728	2.0501	0.8368	2.6071	2.0887	0.8614
SSA-MLP	0.4463	0.3108	0.9947	0.4825	0.3532	0.9937
SFLA-MLP	1.1962	0.8118	0.9621	1.1696	0.8176	0.9645
LCA-MLP	0.6704	0.5172	0.9880	0.7648	0.5430	0.9843

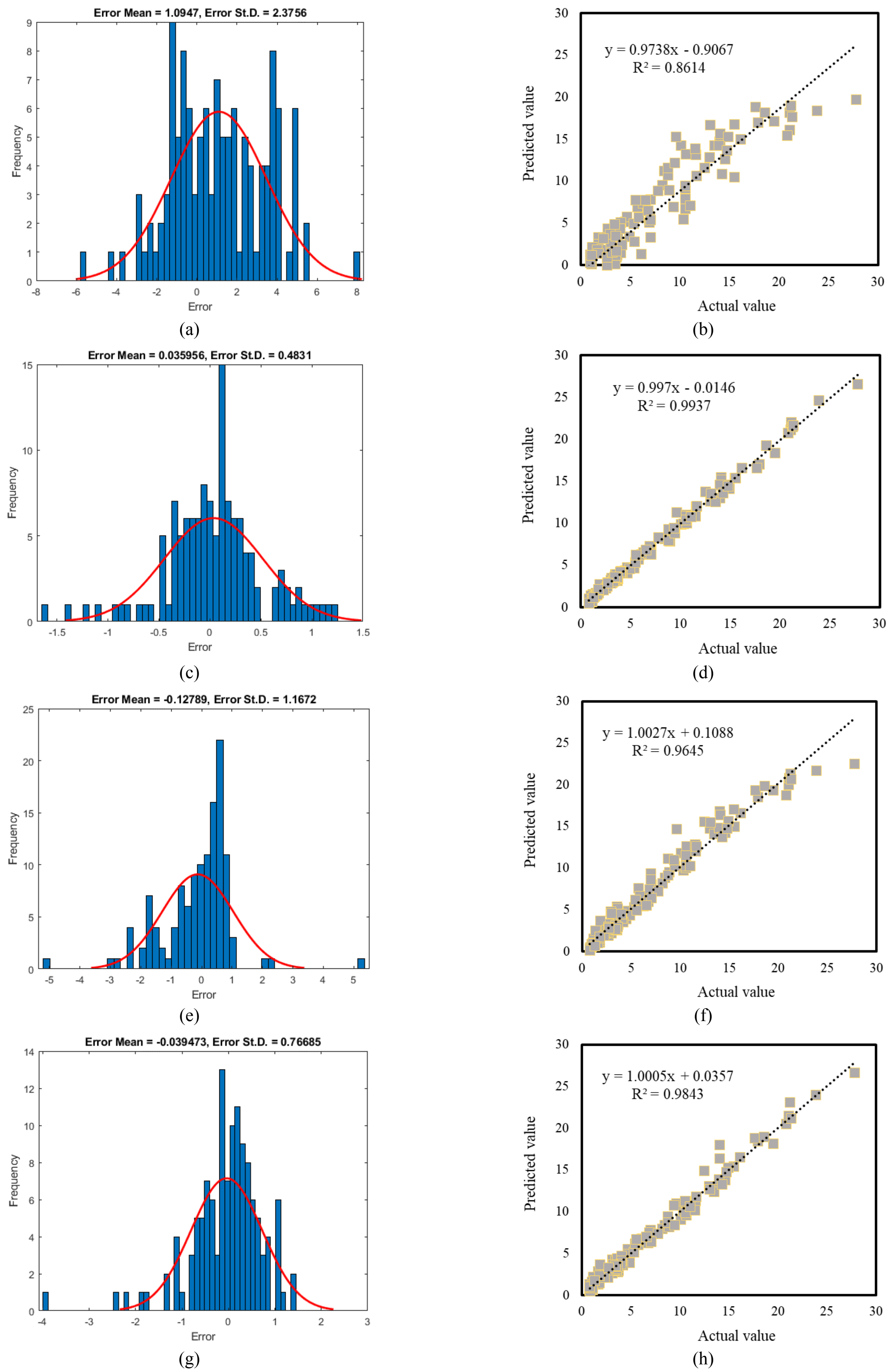


Fig. 6 The testing results for the (a) and (b) SHO-MLP; (c) and (d) SSA-MLP; (e) and (f) SFLA-MLP; and (g) and (h) LCA-MLP prediction

Table 3 The executed ranking system based on accuracy criteria

Models	Scores									
	Training					Testing				
	RMSE	MAE	R^2	Overall grade	Rank	RMSE	MAE	R^2	Overall grade	Rank
SHO-MLP	1	1	1	3	4	1	1	1	3	4
SSA-MLP	4	4	4	12	1	4	4	4	12	1
SFLA-MLP	2	2	2	6	3	2	2	2	6	3
LCA-MLP	3	3	3	9	2	3	3	3	9	2

SSA in all parts.

All in all, the SSA has gained the highest OG (= 12), followed by the LCA (OG = 9), SFLA (OG = 6), and SHO (OG = 4). It indicates that the proposed MLP neural network constructed by the SSA is better optimized than other colleagues. It is proper to mention that catching the same results in the testing phase means that there is no discrepancy between the learning and generalization capabilities of the used models. In other words, the more quality of learning, the more accuracy of prediction for stranger conditions.

3.4 Presenting the neural predictive formula

In the previous section, it was shown that the MLP parameters suggested by the SSA algorithm construct the most accurate neural network. In this section, the optimized parameters of the SSA-MLP ensemble are used to generate the FOS predictive formula. Since the final response of the MLP is released from the output layer, fulfilling this purpose entails extracting the weights and biases of the relevant neuron. These parameters are arranged in the form of Eq. (23). This equation is fed by six variables which are the product of hidden neurons (i.e., the response of neurons in the previous layer). Eq. (24) is used to achieve these variables by receiving the problem parameters (i.e., the *Cu*, *β*, *d*/*D*, and *w*).

$$FOS_{SSA-MLP} = -0.7312 \times A - 0.9610 \times B \\ -0.7498 \times C - 0.5534 \times D \\ -0.1017 \times E + 0.0691 \times F + 0.9808 \tag{23}$$

$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = Tansig\left(\left(\begin{bmatrix} 0.4783 & 1.1539 & 1.0187 & 1.4842 \\ 0.9375 & -1.3227 & 0.8848 & 1.1788 \\ 1.6702 & -0.6027 & -1.1291 & -0.6109 \\ 0.9165 & -1.8969 & -0.5821 & -0.1542 \\ 1.7448 & 0.9500 & 0.3645 & 0.8494 \\ 1.5837 & 1.4772 & -0.1290 & 0.3066 \end{bmatrix} \begin{bmatrix} Cu \\ \beta \\ d/D \\ w \end{bmatrix}\right) + \begin{bmatrix} -2.1911 \\ -1.3147 \\ -0.4382 \\ 0.4382 \\ 1.3147 \\ 2.1911 \end{bmatrix}\right) \tag{24}$$

$$Tansig\ (x) = \frac{2}{1 + e^{-2x}} - 1 \tag{25}$$

A significant outcome of this study was introducing the SSA-MLP as an excellent yet soft method to be used by experts and engineers. Besides, by taking some specific measures regarding neural analysis, the presented formula can be a useful and simple way for calculating the FOS of identical slopes in real projects. It helps engineers avoid complicated and time-consuming finite element analysis. Analyzing the importance of the FOS influential parameters is the last part of this study which can give significant contributions concerning selecting the appropriate parameters in the real-world analysis. Utilizing a regression tree model, the importance of the input factors is calculated. The results are depicted in Fig. 7. As is seen, C_u, β, b/B, and w have gained the importance values of 10.08, 4.48, 0.17, and 5.35, respectively. It denotes that undrained cohesive strength has the greatest effect (in the case of this dataset) on the FOS.

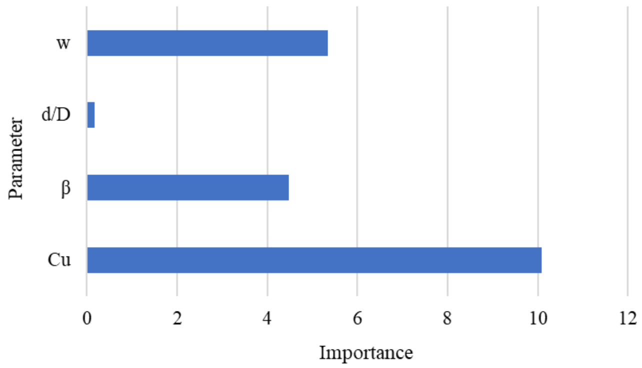


Fig. 7 The importance of the input factors in predicting the FOS

4. Conclusions

Analyzing the stability of soil slope is one of the most crucial issues in geotechnical projects which needs to be properly dealt with. It was the main motivation of the current study to employ four wise metaheuristic techniques namely spotted hyena optimizer, salp swarm algorithm, shuffled frog leaping algorithm, and league champion optimization algorithm for estimating the FOS using four effective factors. The algorithms were coupled with neural network for finding the optimal weights and biases. The population-based sensitivity analysis demonstrated that the best complexity of the SHO, SSA, SFLA, and LCA is 300, 400, 400, and 200, respectively. The results revealed that the combination of the ANN with metaheuristic science can successfully handle the mentioned task. A comparison between the models showed that SSA-MLP is the most accurate ensemble in both training (RMSEs were 2.6728, 0.4463, 1.1962, and 0.6704) and testing phases (RMSEs were 2.6071, 0.4825, 1.1696, and 0.7648). The authors would suggest the use of other optimizers toward achieving the most reliable predictive model in future studies. Also, taking into account the effect of other FOS-related parameters can be investigated.

References

Akin, O. and Sahin, M. (2017), "Active neuro-adaptive vibration suppression of a smart beam", *Smart Struct. Syst., Int. J.*, **20**(6), 657-668. https://doi.org/10.12989/sss.2017.20.6.657

Aljarah, I., Mafarja, M., Heidari, A.A., Faris, H., Zhang, Y. and Mirjalili, S. (2018), "Asynchronous accelerating multi-leader salp chains for feature selection", *Appl. Soft Comput.*, **71**, 964-979. https://doi.org/10.1016/j.asoc.2018.07.040

Bagheri Sereshki, A. and Derakhshani, A. (2018), "Optimizing the Mechanical Stabilization of Earth Walls with Metal Strips: Applications of Swarm Algorithms", *Arab. J. Sci. Eng.*, **445**(5), 4653-4666. https://doi.org/10.1007/s13369-018-3492-8

Balasubbareddy, M., Dwivedi, D. and Sathish, D. (2019), "Optimal Power Flow solution using Spotted Hyena Optimization Algorithm", *Pramana Res. J.*, **9**(3).

Barari, A., Behnia, M. and Najafi, T. (2015), "Determination of the ultimate limit states of shallow foundations using gene expression programming (GEP) approach", *Soils Found.*, **55**(3), 650-659. https://doi.org/10.1016/j.sandf.2015.04.015

Bozorg-Haddad, O. (2018), *Advanced Optimization by Nature-Inspired Algorithms*, Springer.

Bui, D.T., Moayedi, H., Gör, M., Jaafari, A. and Foong, L.K. (2019a), "Predicting slope stability failure through machine learning paradigms", *ISPRS Int. J. Geo-Inform.*, **8**(9), 395. https://doi.org/10.3390/ijgi8090395

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

Bui, D.T., Moayedi, H., Kalantar, B., Osouli, A., Pradhan, B., Nguyen, H. and Rashid, A.S.A. (2019c), "A novel swarm intelligence—Harris hawks optimization for spatial assessment of landslide susceptibility", *Sensors*, **19**(16), 3590. https://doi.org/10.3390/s19163590

Cai, F. and Ugai, K. (2004), "Numerical analysis of rainfall effects on slope stability", *Int. J. Geomech.*, **4**(2), 69-78. https://doi.org/10.1061/(ASCE)1532-3641(2004)4:2(69)

Camargo, J., Velloso, R.Q. and Vargas, E.A. (2016), "Numerical limit analysis of three-dimensional slope stability problems in catchment areas", *Acta Geotechnica*, **11**(6), 1369-1383. https://doi.org/10.1007/s11440-016-0459-3

Chahnasir, E.S., Zandi, Y., Shariati, M., Dehghani, E., Toghroli, A., Mohamad, E.T., Shariati, A., Safa, M., Wakil, K. and Khorami, M. (2018), "Application of support vector machine with firefly algorithm for investigation of the factors affecting the shear strength of angle shear connectors", *Smart Struct. Syst., Int. J.*, **22**(4), 413-424. https://doi.org/10.12989/sss.2018.22.4.413

Chen, W., Panahi, M., Tsangaratos, P., Shahabi, H., Ilia, I., Panahi, S., Li, S., Jaafari, A. and Ahmad, B.B. (2019), "Applying population-based evolutionary algorithms and a neuro-fuzzy system for modeling landslide susceptibility", *Catena*, **172**, 212-231. https://doi.org/10.1016/j.catena.2018.08.025

Chen, H., Zhang, Q., Luo, J., Xu, Y. and Zhang, X. (2020), "An enhanced Bacterial Foraging Optimization and its application for training kernel extreme learning machine", *Appl. Soft Computi.*, **86**, 105884. https://doi.org/10.1016/j.asoc.2019.105884

Choobbasti, A.J., Farrokhzad, F. and Barari, A. (2009), "Prediction of slope stability using artificial neural network (case study: Noabad, Mazandaran, Iran)", *Arab. J. Geosci.*, **2**(4), 311-319. https://doi.org/10.1007/s12517-009-0035-3

Dai, Z., Liu, Z., Liu, C. and He, M. (2008), "Numerical analysis of soil slope stability considering tension and shear failures", *Chinese J. Rock Mech. Eng.*, **2**.

Dhiman, G. (2019), "ESA: a hybrid bio-inspired metaheuristic optimization approach for engineering problems", *Eng. Comput.*, 1-31. https://doi.org/10.1007/s00366-019-00826-w

Dhiman, G. and Kumar, V. (2018), "Multi-objective spotted hyena optimizer: A Multi-objective optimization algorithm for engineering problems", *Knowledge-Based Syst.*, **150**, 175-197. https://doi.org/10.1016/j.knosys.2018.03.011

Divya, S., Kiran, E.L., Rao, M.S. and Vemulapati, P. (2020), *Data Communication and Networks*, Springer, pp. 59-67.

Eusuff, M.M. and Lansey, K.E. (2003), "Optimization of water distribution network design using the shuffled frog leaping algorithm", *J. Water Resour. Plann. Manage.*, **129**(3), 210-225. https://doi.org/10.1061/(ASCE)0733-9496(2003)129:3(210)

Fallahian, M., Khoshnoudian, F. and Talaei, S. (2018), "Application of couple sparse coding ensemble on structural damage detection", *Smart Struct. Syst., Int. J.*, **21**(1), 1-14. 10.12989/sss.2018.21.1.001

Faris, H., Mafarja, M.M., Heidari, A.A., Aljarah, I., Ala'M, A.Z., Mirjalili, S. and Fujita, H. (2018), "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems", *Knowledge-Based Syst.*, **154**, 43-67. https://doi.org/10.1016/j.knosys.2018.05.009

Gandomi, A.H., Kashani, A.R., Mousavi, M. and Jalalvandi, M. (2015), "Slope stability analyzing using recent swarm intelligence techniques", *Int. J. Numer. Anal. Methods Geomech.*, **39**(3), 295-309. https://doi.org/10.1002/nag.2308

Gandomi, A.H., Kashani, A.R., Mousavi, M. and Jalalvandi, M. (2017), "Slope stability analysis using evolutionary optimization techniques", *Int. J. Numer. Anal. Methods Geomech.*, **41**(2), 251-264. https://doi.org/10.1002/nag.2554

Ghiasi, R. and Ghasemi, M.R. (2018), "Optimization-based method for structural damage detection with consideration of uncertainties- a comparative study", *Smart Struct. Syst., Int. J.*, **22**(5), 561-574. https://doi.org/10.12989/sss.2018.22.5.561

Gordan, B., Armaghani, D.J., Hajihassani, M. and Monjezi, M. (2016), "Prediction of seismic slope stability through combination of particle swarm optimization and neural network", *Eng. Comput.*, **32**(1), 85-97. https://doi.org/10.1007/s00366-015-0400-7

Hecht-Nielsen, R. (1992), *Neural Networks for Perception*, Elsevier, pp. 65-93.

Hornik, K., Stinchcombe, M. and White, H. (1989), "Multilayer feedforward networks are universal approximators", *Neural Networks*, **2**(5), 359-366.

Hussien, A.G., Hassanien, A.E. and Houssein, E.H. (2017), "Swarming behaviour of salps algorithm for predicting chemical compound activities", *Proceedings of 2017 Eighth International Conference on Intelligent Computing and Information Systems* (*ICICIS*), Cairo, Egypt, January, pp. 315-320.

https://doi.org/10.1109/INTELCIS.2017.8260072

Ibrahim, R.A., Ewees, A.A., Oliva, D., Elaziz, M.A. and Lu, S. (2019), "Improved salp swarm algorithm based on particle swarm optimization for feature selection", *Journal of Ambient Intelligence and Humanized Computing*, **10**(8), 3155-3169. https://doi.org/10.1007/s12652-018-1031-9

Jalili, S., Kashan, A.H. and Hosseinzadeh, Y. (2016), "League championship algorithms for optimum design of pin-jointed structures", *J. Comput. Civil Eng.*, **31**(2), 04016048. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000617

Jia, H., Li, J., Song, W., Peng, X., Lang, C. and Li, Y. (2019), "Spotted hyena optimization algorithm with simulated annealing for feature selection", *IEEE Access*, **7**, 71943-71962. https://doi.org/10.1109/ACCESS.2019.2919991

Kashan, A.H. (2009), "League championship algorithm: a new algorithm for numerical function optimization", *Proceedings of 2009 International Conference of Soft Computing and Pattern Recognition*, Malacca, Malaysia, December, pp. 43-48. https://doi.org/10.1109/SoCPaR.2009.21

Kashan, A.H. (2011), "An efficient algorithm for constrained global optimization and application to mechanical engineering design: League championship algorithm (LCA)", *Comput.-Aided Des.*, **43**(12), 1769-1792. https://doi.org/10.1016/j.cad.2011.07.003

Kashan, A.H. (2014), "League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships", *Appl. Soft Comput.*, **16**, 171-200. https://doi.org/10.1016/j.asoc.2013.12.005

Keshavarz, Z. and Torkian, H. (2018), "Application of ANN and ANFIS models in determining compressive strength of concrete", *Soft Comput. Civil Eng.*, **2**(1), 62-70. https://doi.org/10.22115/SCCE.2018.51114

Kimiyaghalam, A., Ashouri, A. and Beykverdi, M. (2012), "Application of IBSFLA and BSFLA approaches for locating of fault indicators in distribution networks", *Proceedings of 17th Conference on Electrical Power Distribution*, Tehran, Iran, August, pp. 1-7.

Kohestani, V.R., Vosoghi, M., Hassanlourad, M. and Fallahnia, M. (2017), "Bearing capacity of shallow foundations on cohesionless soils: A random forest based approach", *Civil Eng. Infrastruct. J.*, **50**(1), 35-49. https://doi.org/10.7508/CEIJ.2017.01.003

Krabbenhoft, K., Lyamin, A. and Krabbenhoft, J. (2015), "Optum Computational Engineering (Optum G2)". Available on: www.optumce.com

Leshchinsky, B. and Ambauen, S. (2015), "Limit equilibrium and limit analysis: comparison of benchmark slope stability problems", *J. Geotech. Geoenviron. Eng.*, **141**(10), 04015043. https://doi.org/10.1061/(ASCE)GT.1943-5606.0001347

Li, B., Li, D., Zhang, Z., Yang, S. and Wang, F. (2015), "Slope stability analysis based on quantum-behaved particle swarm optimization and least squares support vector machine", *Appl. Mathe. Model.*, **39**(17), 5253-5264. https://doi.org/10.1016/j.apm.2015.03.032

Li, A.J., Khoo, S., Lyamin, A.V. and Wang, Y. (2016), "Rock slope stability analyses using extreme learning neural network and terminal steepest descent algorithm", *Automat. Constr.*, **65**, 42-50. https://doi.org/10.1016/j.autcon.2016.02.004

Li, A., Fang, Q., Zhang, D., Luo, J. and Hong, X. (2018), "Blast vibration of a large-span high-speed railway tunnel based on microseismic monitoring", *Smart Struct. Syst., Int. J.*, **21**(5), 561-569. https://doi.org/10.12989/sss.2018.21.5.561

Lim, K., Li, A.J., Schmid, A. and Lyamin, A.V. (2016), "Slope-stability assessments using finite-element limit-analysis methods", *Int. J. Geomech.*, **17**(2), 06016017. https://doi.org/10.1061/(ASCE)GM.1943-5622.0000715

Liping, Z., Weiwei, W., Yi, H., Yefeng, X. and Yixian, C. (2012), "Application of shuffled frog leaping algorithm to an uncapacitated SLLS problem", *AASRI Procedia*, **1**, 226-231. https://doi.org/10.1016/j.aasri.2012.06.035

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

Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H. and Mirjalili, S.M. (2017), "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems", *Adv. Eng. Software*, **114**, 163-191. https://doi.org/10.1016/j.advengsoft.2017.07.002

Moayedi, H. and Hayati, S. (2018), "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., Mehrabi, M., Mosallanezhad, M., Rashid, A.S.A. and Pradhan, B. (2018), "Modification of landslide susceptibility mapping using optimized PSO-ANN technique", *Eng. Comput.*, **35**, 967-984. https://doi.org/10.1007/s00366-018-0644-0

Moayedi, H., Osouli, A., Nguyen, H. and Rashid, A.S.A. (2019a), "A novel Harris hawks' optimization and k-fold cross-validation predicting slope stability", *Eng. Comput.*, 1-11. https://doi.org/10.1007/s00366-019-00828-8

Moayedi, H., Tien Bui, D., Dounis, A., Kok Foong, L. and Kalantar, B. (2019b), "Novel Nature-Inspired Hybrids of Neural Computing for Estimating Soil Shear Strength", *Appl. Sci.*, **9**(21), 4643. https://doi.org/10.3390/app9214643

Moayedi, H., Tien Bui, D., Gör, M., Pradhan, B. and Jaafari, A. (2019c), "The feasibility of three prediction techniques of the artificial neural network, adaptive neuro-fuzzy inference system, and hybrid particle swarm optimization for assessing the safety factor of cohesive slopes", *ISPRS Int. J. Geo-Inform.*, **8**(9), 391. https://doi.org/10.3390/ijgi8090391

Moayedi, H., Gör, M., Lyu, Z. and Bui, D.T. (2020a), "Herding Behaviors of grasshopper and Harris hawk for hybridizing the neural network in predicting the soil compression coefficient", *Measurement*, **152**, 107389. https://doi.org/10.1016/j.measurement.2019.107389

Moayedi, H., Tien Bui, D., Dounis, A. and Ngo, P.T.T. (2020b), "A novel application of league championship optimization (LCA): hybridizing fuzzy logic for soil compression coefficient analysis", *Appl. Sci.*, **10**(1), 67. https://doi.org/10.3390/app10010067

Mosallanezhad, M. and Moayedi, H. (2017), "Comparison analysis of bearing capacity approaches for the strip footing on layered soils", *Arab. J. Sci. Eng.*, **42**(9), 3711-3722. https://doi.org/10.1007/s13369-017-2490-6

Ng, C. and Shi, Q. (1998), "A numerical investigation of the stability of unsaturated soil slopes subjected to transient seepage", *Comput. Geotech.*, **22**(1), 1-28. https://doi.org/10.1016/S0266-352X(97)00036-0

Nguyen, H., Mehrabi, M., Kalantar, B., Moayedi, H. and Abdullahi, M.A.M. (2019), "Potential of hybrid evolutionary approaches for assessment of geo-hazard landslide susceptibility mapping", *Geomat. Natural Hazards Risk*, **10**(1), 1667-1693. https://doi.org/10.1080/19475705.2019.1607782

Onat, O. and Gul, M. (2018), "Application of Artificial Neural Networks to the prediction of out-of-plane response of infill walls subjected to shake table", *Smart Struct. Syst., Int. J.*, **21**(4), 521-535. https://doi.org/10.12989/sss.2018.21.4.521

Pradatta, S.P., Kumar, S.S. and Alireza, M. (2018), "Analytical expressions for determining the stability of cohesionless soil slope under generalized seismic conditions", *J. Mount. Sci.*, **15**(7), 1559-1571. https://doi.org/10.1007/s11629-017-4780-6

Qiao, W., Moayedi, H. and Foong, L.K. (2020), "Nature-inspired hybrid techniques of IWO, DA, ES, GA, and ICA, validated through a k-fold validation process predicting monthly natural gas consumption", *Energy Build*., 110023. [In press] https://doi.org/10.1016/j.enbuild.2020.110023

Roy, P., Roy, P. and Chakrabarti, A. (2013), "Modified shuffled frog leaping algorithm with genetic algorithm crossover for solving economic load dispatch problem with valve-point effect", *Appl*. *Soft Comput*., **13**(11), 4244-4252. https://doi.org/10.1016/j.asoc.2013.07.006

Sakellariou, M. and Ferentinou, M. (2005), "A study of slope stability prediction using neural networks", *Geotech*. *Geol*. *Eng*., **23**(4), 419. https://doi.org/10.1007/s10706-004-8680-5

Shen, L., Chen, H., Yu, Z., Kang, W., Zhang, B., Li, H., Yang, B. and Liu, D. (2016), "Evolving support vector machines using fruit fly optimization for medical data classification", *Knowledge-Based Syst*., **96**, 61-75. https://doi.org/10.1016/j.knosys.2016.01.002

Supraja, P. and Jayashri, S. (2017), "Optimized neural network for spectrum prediction scheme in cognitive radio", *Wireless Personal Commun*., **94**(4), 2597-2611. https://doi.org/10.1007/s11277-016-3818-3

Wang, M. and Chen, H. (2020), "Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis", *Appl*. *Soft Comput*., **88**, 105946. https://doi.org/10.1016/j.asoc.2019.105946

Wang, M., Chen, H., Yang, B., Zhao, X., Hu, L., Cai, Z., Huang, H. and Tong, C. (2017), "Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses", *Neurocomputing*, **267**, 69-84. https://doi.org/10.1016/j.neucom.2017.04.060

Wang, L., Chen, B. and Li, J. (2019), "Three-dimensional seismic stability of unsaturated soil slopes using a semi-analytical method", *Comput*. *Geotech*., **110**, 296-307. https://doi.org/10.1016/j.compgeo.2019.02.008

Xu, X. and Chen, H.L. (2014), "Adaptive computational chemotaxis based on field in bacterial foraging optimization", *Soft Computing*, **18**(4), 797-807. https://doi.org/10.1007/s00500-013-1089-4

Xu, Y., Chen, H., Luo, J., Zhang, Q., Jiao, S. and Zhang, X. (2019), "Enhanced Moth-flame optimizer with mutation strategy for global optimization", *Inform*. *Sci*., **492**, 181-203. https://doi.org/10.1016/j.ins.2019.04.022

Yuan, C. and Moayedi, H. (2019), "The performance of six neural-evolutionary classification techniques combined with multi-layer perception in two-layered cohesive slope stability analysis and failure recognition", *Eng*. *Comput*., 1-10. https://doi.org/10.1007/s00366-019-00791-4

Zhang, X. and Zhou, X. (2018), "Analysis of the numerical stability of soil slope using virtual-bond general particle dynamics", *Eng*. *Geol*., **243**, 101-110. https://doi.org/10.1016/j.enggeo.2018.06.018

Zhang, X., Zhang, Y., Shi, Y., Zhao, L. and Zou, C. (2012), "Power control algorithm in cognitive radio system based on modified shuffled frog leaping algorithm", *AEU-Int*. *J*. *Electron*. *Commun*., **66**(6), 448-454. https://doi.org/10.1016/j.aeue.2011.10.004

Zhao, X., Li, D., Yang, B., Ma, C., Zhu, Y. and Chen, H. (2014), "Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton", *Appl*. *Soft Comput*., **24**, 585-596. https://doi.org/10.1016/j.asoc.2014.07.024

Zhao, X., Zhang, X., Cai, Z., Tian, X., Wang, X., Huang, Y., Chen, H. and Hu, L. (2019), "Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients", *Computat*. *Biol*. *Chem*., **78**, 481-490. https://doi.org/10.1016/j.compbiolchem.2018.11.017

Zhou, G., Moayedi, H., Bahiraei, M. and Lyu, Z. (2020), "Employing artificial bee colony and particle swarm techniques for optimizing a neural network in prediction of heating and cooling loads of residential buildings", *J*. *Cleaner Prod*., **254**, 120082. https://doi.org/10.1016/j.jclepro.2020.120082

Zounemat-Kermani, M., Kişi, Ö., Adamowski, J. and Ramezani-Charmahineh, A. (2016), "Evaluation of data driven models for river suspended sediment concentration modeling", *J*. *Hydrol*., **535**, 457-472. https://doi.org/10.1016/j.jhydrol.2016.02.012

CC