Contents lists available at ScienceDirect

Ain Shams Engineering Journal

journal homepage: www.sciencedirect.com

Predicting longitudinal dispersion coefficient using ensemble models and optimized multi-layer perceptron models

Mahsa Gholami [a], Elham Ghanbari-Adivi [b], Mohammad Ehteram [c,*], Vijay P. Singh [d], Ali Najah Ahmed [e], Amir Mosavi [f,g,*], Ahmed El-Shafie [h,i]

[a] Department of Civil Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran
[b] Department of Water Science Engineering, Shahrekord University, Shahrekord, Iran
[c] Department of Civil Engineering, Semnan University, Semnan, Iran
[d] Department of Biological and Agricultural Engineering, Zachry Department of Civil Engineering, Texas A & M University, College Station, TX, 77843-2117, USA
[e] Institute of Energy Infrastructure (IEI) and Department of Civil Engineering, College of Engineering, University Tenaga Nasional (UNITEN), Selangor, Malaysia
[f] John von Neumann Faculty of Informatics, Obuda University, 1034 Budapest, Hungary
[g] German Research Center for Artificial Intelligence, Oldenburg, Germany
[h] Department of Civil Engineering, Faculty of Engineering, University of Malaya (UM), Kuala Lumpur, Malaysia
[i] National Water and Energy Center, United Arab Emirates University, Al Ain, United Arab Emirates

ARTICLE INFO

Article history:
Received 3 October 2022
Revised 16 December 2022
Accepted 20 February 2023
Available online xxxx

Keywords:
Longitudinal dispersion coefficient
Multilayer perceptron
Optimization
Artificial intelligence
Machine learning
Deep learning
Big data

ABSTRACT

Prediction of the longitudinal dispersion coefficient (LDC) is essential for the river and water resources engineering and environmental management. This study proposes ensemble models for predicting LDC based on multilayer perceptron (MULP) methods and optimization algorithms. The honey badger optimization algorithm (HBOA), salp swarm algorithm (SASA), firefly algorithm (FIFA), and particle swarm optimization algorithm (PASOA) are used to adjust the MULP parameters. Then, the outputs of the MULP-HBOA, MULP-SASA, MULP-PASOA, MULP-FIFA, and MULP models were incorporated into an inclusive multiple model (IMM). For IMM at the testing level, the mean absolute error (MEAE) was 15, whereas it was 17, 18, 23, 24, and 25 for the MULP-HBOA, MULP-SASA, MULP-FIFA, MULP-PASOA, and MULP models. The study also modified the structure of MULP models using a goodness factor which decreased the CPU time. Removing redundant neurons reduces CPU time. Thus, the modified ANN model and the suggested IMM model can decrease the computational time and further improve the performance of models.

1. Introduction

The quality of water-both surface and ground water- is affected by a variety of contaminants emanating from air, industries, sewage, and agriculture [12]. The contaminants can be physical, chemical or biological. For control of physical, chemical and biological pollution, it is essential to determine the transport of pollutants in natural streams [9]. While pollutants in receiving streams are dispersed and mixed in all three dimensions, longitudinal dispersion is predominant [4]. The longitudinal dispersion coefficient (LDC) determination [20,22] can be done experimentally or mathematically. Experimentation is often costly and time-consuming. The LDC determination is challenging because different parameters influence the mixing process. Because of the difficulty of estimating the transverse profiles of flow velocity, it is not easy to theoretically determine LDC [4]. LDC prediction is complicated because different factors affect it. Different factors and inputs should be identified to model LDC. The inputs of models may have uncertainties. These uncertainties may decrease the accuracy of the final outputs. Since LDC prediction is an important topic, developing robust models is necessary. While numerical models may be used for predicting LDC, they may be time-consuming. Numerical models may include complex equations. Therefore, modelers should develop robust alternatives to these models. Hence, in recent years LDCs have been predicted using soft computing models.

High accuracy, flexibility, and efficiency are the advantages of soft computing models [22]. Parsaie and Haghiabi [25] predicted LDC using empirical formulas, radial basis functions, and MULP. Results showed that the MULP model was suitable for predicting LDC. For predicting LDC in rivers, Noori et al. [21] evaluated the

* Corresponding authors at: Obuda University, Mohammad Ehteram; Semnan University, Iran (A. Mosav).

E-mail addresses: mohammdehteram@semnan.ac.ir (M. Ehteram), amir.mosavi@nik.uni-obuda.hu (A. Mosavi).

https://doi.org/10.1016/j.asej.2023.102223
2090-4479/

reliability of artificial neural networks (ANN), adaptive neuro-fuzzy inference system (ANFIS), and support vector machines. Although the models predicted LDC satisfactorily, they had high uncertainty.

Alizadeh et al. [1] used four metaheuristic algorithms to train ANN models, including genetic, imperialist competitive, bee, and cuckoo search algorithms, to predict LDC. They found that metaheuristic algorithms enhanced the accuracy of conventional ANN models. Using evolutionary polynomial regression, Balf et al. [5] accurately predicted LDC, with the coefficient of determination and root mean square error of 0.82 and 79 m^2/s, respectively. Nezaratian et al. [20] used several empirical and data-driven models to predict LDC, and based on Monte Carlo simulation, most of these models were highly uncertain for upper LDC values. Ghiasi et al. [10] found that the hybrid granular computing - ANN model (GRC-ANN) was the best model for estimating the LDC in rivers. Utilizing subset selection by maximum dissimilarity, Riahi-Madvar et al. [27] found that ANFIS hybridized with FFA would perform better than ANFIS. Goliatt et al. [12] combined Gaussian Process Regression (GPR) and evolutionary feature selection (FS) to estimate LDC in streams. They found that an integrated GPR model based on an FS approach produced the minimum root mean square error.

Based on previous studies, these models have different drawbacks. Each model, such as ANN, ANFIS, and SVM, has its own advantages and disadvantages [24]. Previous studies inserted input data into individual models [11]. When individual models are used, the results may not be accurate. An additional drawback of prior models is their time-consuming nature. Modelers have not attempted to reduce computational costs when using these models. Computational costs can be reduced by modifying the structure of these models. By developing novel ensemble models and modifying the structure of ANN models, this study attempts to address these research gaps. Various types of ensemble models exist. A Bayesian model averaging (BMA) is an ensemble model. While this model can be successfully used to estimate variables, it has some disadvantages. It is difficult to calculate posterior distributions and prior distributions [24]. However, statistical computations of models can be complex and time-consuming. For predicting LDC, we propose a novel ensemble model as a robust alternative to the BMA models. Even though the ANN model is an individual model, modelers can convert it to an ensemble model. As an ensemble model, an ANN receives the outputs of other models as inputs to estimate target variables. Multiple models can be incorporated into an ANN model without requiring computational complexity, such as calculating posterior and prior distributions. ANNs receive outputs from individual models. Clearly, the modeling process is not complicated or difficult. This ensemble model has broader applications than just predicting LDCs. It can be used for estimating different variables such as solar radiation, runoff, and sediment load. The IMM model requires outputs of ANN models for predicting outputs, so we modify the structure of ANN models to satisfy modeling processes and reduce computation time.

Thus another innovation of this study is the modification of the structure of ANN models. In order to simplify the modeling process, it is useful to adjust the structure of an ANN model. Some of neurons of ANN models are inactive during the modeling process. These neurons can be identified and removed from the modeling process. Due to the number of inactive hidden neurons, the modelling process may be time consuming Thus, the other aim of the article is to develop a novel structure for ANN models. This modified ANN model can be useful when modelers encounter a large number of data. Our main objective is to use modified ANN models and a novel ensemble model to predict LDC. The present study develops an IMM model for computing LDC. Khatibi and Nadiri [14] used IMM models to predict groundwater levels and showed that the IMM improved individual model accuracy in two steps.

In Step 1, input and output variables were used to train independent intelligence models using the basic input and output variables. Step 2 begins once step 1 is completed, and it must be fully implemented. As input for step 2, step 1′s results were used. The first step of this study was to estimate LDC using individual ANNs. An ANN model was used as an IMM model (combiner) in the second step to predict the LDC based on the outputs of individual ANN models. Therefore, the main contribution of this paper is the introduction of the IMM model for predicting LDC. Modifying the structure of ANN models is another innovation of the paper.

The computational time and cost of an ANN may be high. This paper uses structural learning to optimize the hidden layer of an ANN. Using this method, we can reduce computation time and cost. Finally, this paper uses new optimization algorithms to train ANN models. In the second section of this paper, we discuss the methods. The third section explains data collection. In the fourth section, results and discussion are presented. The fifth section concludes the paper.

2. Material and method

2.1. Structure of ANN model

The MULP model is one of the well-known ANN models. There are different types of computational layers in MULPs. Layer one handles input data. The second layer, called the hidden layer, contains the activation function which processes the data. A neuron is MULP's small processor. Layers are connected by connections called weights [8]. In the final layer, the output is given.

Fig. 1a and b illustrate the structure and mathematical of MULP. The number of units in an ANN should be sufficient so the network can estimate values based on many data points [8].

The number of hidden units can be affected by several factors, including the learning method and the number of input and output units [30,31]. Reducing the number of hidden units reduces the computational time of modeling [30].

This study evaluated the effect of hidden units in each layer and built an eliminating learning algorithm with a proposed goodness factor (Fig. 1c). [16] introduced a goodness factor to remove redundant units from a computational system. In layer k, the unit with the smallest goodness factor is regarded as the most useless unit, labeled as "bad." In the first step, the number of hidden units is determined. Next, the connection weights and biases are determined using training algorithms. In the next level, the goodness factor is used to determine ineffective neurons.

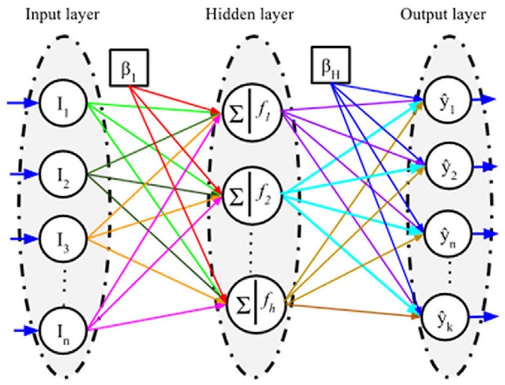
This study used optimization algorithms for adjusting the MULP parameters since traditional training algorithms may fall into the local optimum [11].

2.2. Optimization algorithms

The MULP parameters were adjusted using optimization algorithms. This paper uses algorithms that are highly accurate, flexible for coupling with machine learning models, and have a high potential for global searches. Due to these factors, we used the SASA, HBOA, PASOA, and FIFA in this study. These optimization algorithms are used because of their advanced operators. The fast convergence and high accuracy are the advantages of these algorithms.

2.2.1. Structure of honey badger optimization algorithm (HBOA)

HOBA is a new metaheuristic optimization algorithm developed by Hashim et al. [13]. Based on honey badgers' intelligent foraging



(I: input, f: activation function, y: output, β : bias)

a

$$Y_k = f_0\left[\sum_{i=1}^{M_N} W_{kj}\times f_h\left(\sum_{i=1}^{N_N} W_{ji}X_i + b_{jo}\right)+b_{ko}\right]$$

f_o: The activation function of output, f_h: The activation function of hidden layer, X_i: The input variable, W_{ij} :a weight in the hidden layer connecting the ith neuron in the input layer and the jth neuron in the hidden layer, W_{kj} : a weight in the output layer connecting the jth neuron in the hidden layer and the kth neuron in the output layer b_{jo} :the bias for the jth hidden neuron, b_{ko} : the bias for the kth output neuron, N_N: The Number of inputs, and M_N: The number of hidden neurons.

b

$$G_i^k = \sum_n\sum_j \left(W_{i,j}^k o_{i,j}^k\right)^2$$

Where G_i^k : goodness factor for the ith unit in the kth layer, $W_{i,j}^k$ the weight that links the ith unit in k layer with the jth unit in the layer k + 1, $o_{i,j}^k$: the *i* output unit from the *k* layer, and n: summation over whole leaning patterns

c

Fig. 1. A: mathematical model of ann, b: structure of ann model, and c: mathematical model of goodness factor.

behavior, a search strategy was developed mathematically to solve optimization problems [13]. For finding food sources, honey badgers smell, dig or follow honeyguide birds. HBOA has two modes. First, the digging mode is used, and then the honey mode is used. When honey badgers dig, they use their olfactory sense to find and select the best location for catching their prey.

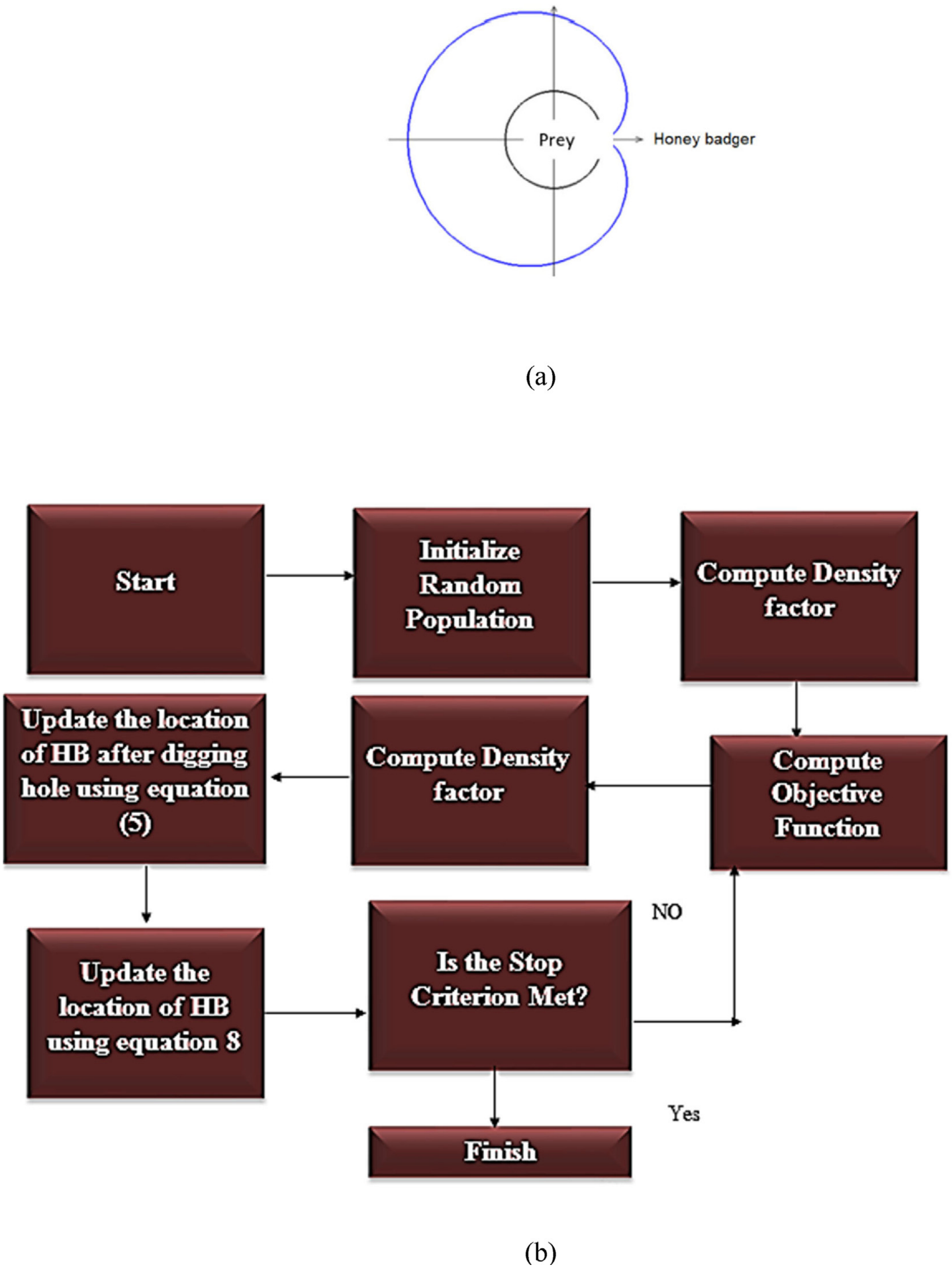

Prey
Honey badger
(a)
Start
Initialize Random Population
Compute Density factor
Compute Objective Function
Compute Density factor
Update the location of HB after digging hole using equation (5)
Update the location of HB using equation 8
Is the Stop Criterion Met?
NO
Yes
Finish
(b)

Fig. 2. a: the digging step: the external outline refers to the smell intensity and the inner circular illustrates the prey location and b: the HBOA flowchart.

The honeyguide bird guides the honey badger to find beehives in the latter mode. The following equation determines the initial population of honey badgers, as described in [13] as follows.

$$HoB_i = LB_i + ra(UP_i - LB_i) \tag{1}$$

where HoB_i: the location of the ith honey badger, ra: the random number, UP_i and LB_i: the upper and lower bounds of the decision variable Using their olfactory senses, honey badgers find food. HBs are closer to their prey when the smell intensifies [13].

$$I_i = r_2 \times \frac{S}{4\pi dis_i^2} \tag{2}$$

$$S = (HoB_i - HoB_{i+1})^2 \tag{3}$$

$$dis_i = HoB_{prey} - HoB_i \tag{4}$$

where S: the source strength, HoB_i: the location of the ith HB, HoB_{i+1}: the location of the i + 1th HB, dis_i: the distance between prey and the ith Honeybadger, r$_2$: the random number, HoB_{prey}: the prey position, and I_i: the smell intensity. A honey badger's digging action is similar to a cardioid shape in Fig. 2a. Eq. (5) simulates the cardioid motion:

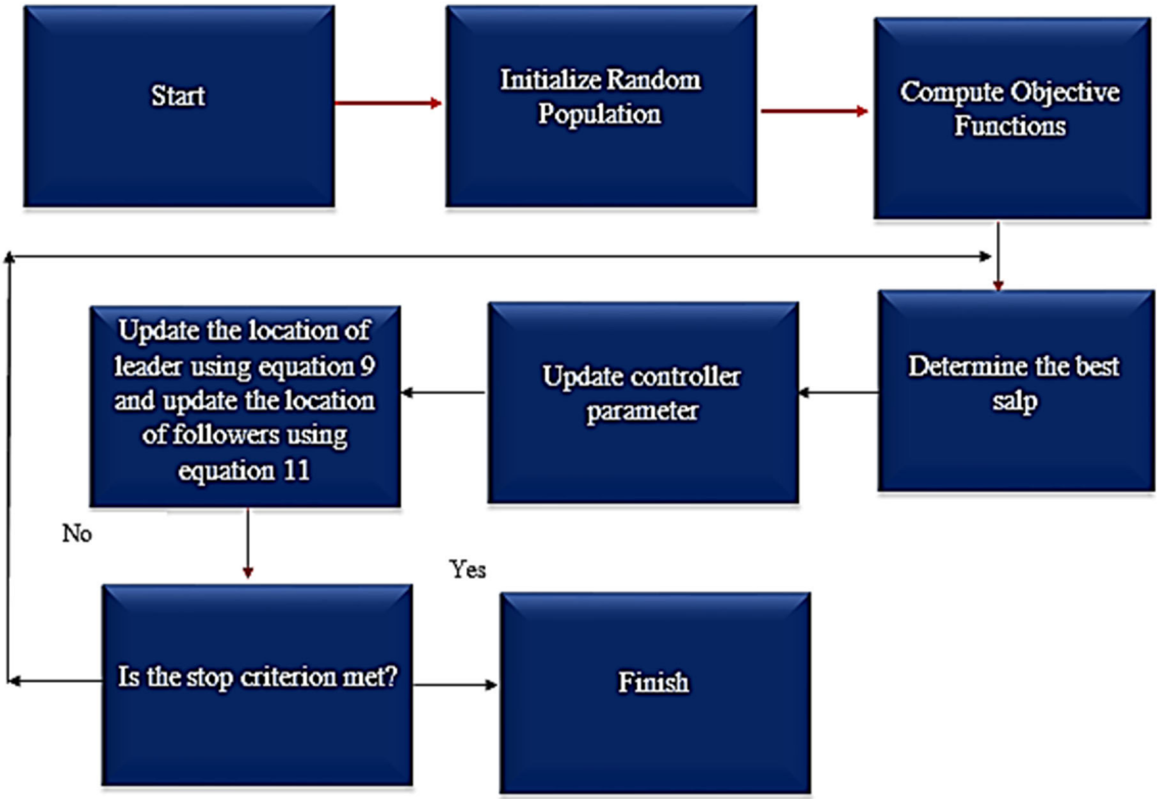

Start
Initialize Random Population
Compute Objective Functions
Update the location of leader using equation 9 and update the location of followers using equation 11
Update controller parameter
Determine the best salp
No
Yes
Is the stop criterion met?
Finish

Fig. 3. Proposed flowchart of SSA.

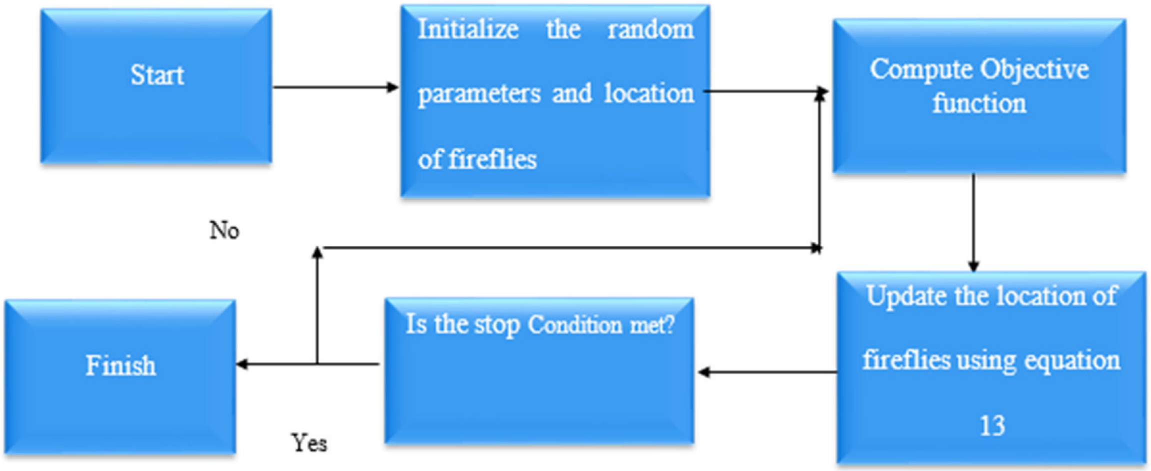

Start
Initialize the random parameters and location of fireflies
Compute Objective function
No
Finish
Is the stop Condition met?
Update the location of fireflies using equation 13
Yes

Fig. 4. Proposed flowchart of FFA.

$$HoB_{new} = HoB_{prey} + F \times \eta \times I \times HoB_{prey} + F \times rand_1 \times \alpha \times dis_i \times (\cos(2\pi rand_2) \times (1 - \cos(2\pi rand_3))) \quad (5)$$

where HoB_{new}: the new location of the honey badger; rand_1, rand_2, and rand_3: the random numbers; α: the density factor; η: the controller parameter; and F: the flag. F is computed as follows, according to [13].

$$F = \begin{bmatrix} 1 \leftarrow if(rand_4) \leqslant 0.50 \\ -1 \leftarrow else \end{bmatrix} \quad (6)$$

$$\alpha = C \times \exp\left(\frac{-it}{it_{\max}}\right) \quad (7)$$

where C, and it are constant value (2), the number of iterations (NOI), respectively; and $it_{\max}$: the maximum number of iterations (MNI). Honey badgers follow a honeyguide bird to get to beehives during the honey phase. This behavior can be simulated using Eq. (8):

$$HoB_{new} = HoB_{prey} + F \times rand_4 \times \alpha \times dis_i HoB_{new} = HoB_{prey} + F \times rand_4 \times \alpha \times dis_i \quad (8)$$

The HBOA flowchart is shown in Fig. 2b.

2.2.2. Structure of salp swarm algorithm (SASA)

Mirjalili et al. [19] developed the SASA to solve optimization problems. SASA mimics the salps' behavior in nature. Salp chains

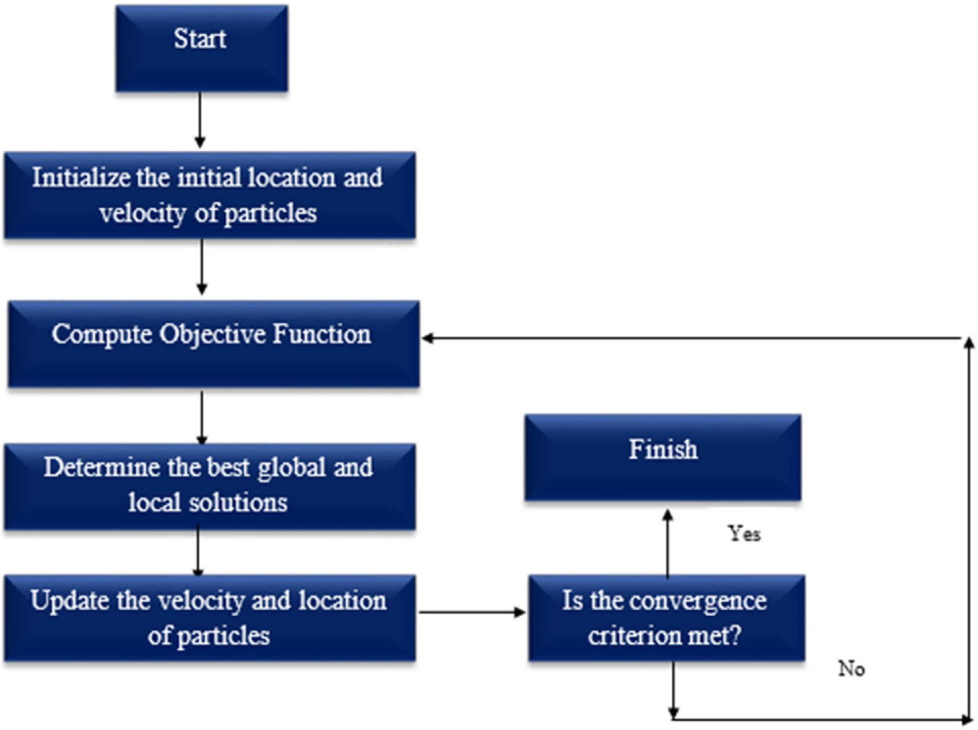

Start
Initialize the initial location and velocity of particles
Compute Objective Function
Determine the best global and local solutions
Update the velocity and location of particles
Is the convergence criterion met?
Yes
Finish
No

Fig. 5. Proposed flowchart of PSO.

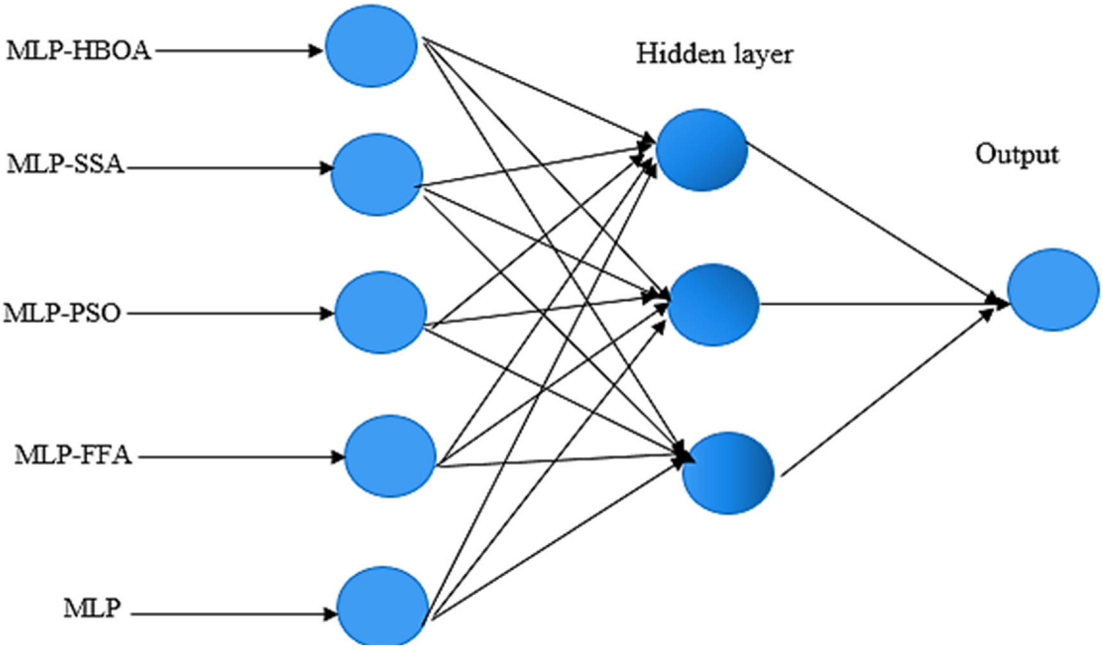


Fig. 6. Structure of IMM.

Table 1
Details of data points for this study [7,15,6,26;2].

Parameter	Minimum	Maximum	Median	Standarad deviation	Average
LDC (m^2/s)	0.0047	1798.60	14.93	192.18	71.53
Shear velocity (m/s)	0.001	0.99	0.0543	0.007	0.065
Velocity (m/)	0.022	1.74	0.43	0.31	0.48
Flow depth	0.034	19.9	0.71	2.3	1.41
Width	0.2	867	27	11.06	56.77

Table 2
Error indices for comparing models.

Index	Equation
Root mean square error	$RMSE = \sqrt{\frac{\sum_{i=1}^{NU}(LDC_{iob} - LDC_{ies})}{NU}}$
Mean absolute error (MEAE)	$MEAE = \frac{\sum_{i=1}^{NU}\lvert LDC_{iobs} - LDC_{ies}\rvert}{NU}$
Nash Sutcliff efficiency	$NSE = 1 - \frac{\sum_{i=1}^{NU}(LDC_{i0b} - LDC_{ies})^2}{\sum_{i=1}^{NU}\left(LDC_{i0b} - \bar{LDC}_{iob}\right)^2}$
Percent Bias	$PBIAS = \frac{\sum_{i=1}^{N}(LDC_{iobs} - LDC_{ies})}{\sum_{i=1}^{N}(LDC_{iobs})}$
Centered Root Mean Square Difference Error	$CRMSE = \sqrt{\frac{\sum_{i=1}^{NU}\left(LDC_{iobs} - \bar{LDC}_{iobs}\right) - \left(LDC_{ies} - \bar{LDC}_{ies}\right)}{NU}}$

LDC_{i0b}: the measured data, LDC_{ies}: the estimated data, $\bar{LDC}_{iob}$: the average observed data, $\bar{LDC}_{ies}$: the average estimated data, NU: the number of data.

contain a leader salp, followed by the rest of the salps. Food sources indicate the swarm's target. Like other swarm algorithms, SASA creates a set of salps with random positions. The fitness of each salp determines which one is the best. As a food source, the salp chain searches for the best place. Eq. (9) is utilized to change the position of the leader [19].

$$Le_i^l = \begin{bmatrix} food_i + r_1((up_i - low_i)r_2 + low_i) \leftarrow r_3 \geqslant 0 \\ food_i - r_1((up_i - low_i)r_2 + low_i) \leftarrow r_3 < 0 \end{bmatrix} \tag{9}$$

where Le_i^l: the leader's location, $food_i$: the location of food source, up_i and low_i: the upper and lower bound of variables, and r$_2$, and r$_3$: random numbers. r$_1$ is computed as follows:

$$r_1 = 2e^{\left(-\frac{4l}{L}\right)^2} \tag{10}$$

where L is the MNI, and l is the NOI. The SASA updates the followers' positions after updating the leaders' positions.

$$salp_j^i = \frac{1}{2}\left(salp_j^i + salp_j^{i-1}\right) \tag{11}$$

where $salp_j^i$: the ith follower location within the j-th dimension. SASA's advantages are its simplicity, speed, and ease of hybridization with other techniques. The algorithm has another unique advantage. There is only one parameter (r$_1$), which balances exploration and exploitation. Fig. 3 illustrates the structure of SASA.

2.2.3. Structure of firefly algorithm (FIFA)

Yang (2009) developed the FIFA based on the flashing patterns of fireflies. FIFA is flexible, simple, and easy to use [31]. A firefly is attracted to another firefly regardless of its gender. The firefly with less brightness goes toward the firefly with more brilliance. A firefly's brightness is determined by its objective function. The following equation is used to compute the distance between two fireflies:

$$r_{ij} = \sqrt{\sum_{k=1}^{D}(f_{ik} - f_{jk})^2} \tag{12}$$

where r_{ij}: the distance between two fireflies i and j, f_{ik}: the location of firefly i in the kth dimension, and f_{jk}: the location of firefly j in the kth dimension, and D: the number of dimensions. After calculating the distance between two fireflies, the next place of the firefly is determined.

$$f_i' = f_i + \beta_0 e^{-\gamma r^m}(f_i - f_j) + \delta\varepsilon_i \tag{13}$$

where f': the new location of firefly, f_i: the location of the ith firefly, f_j: the location of the jth firefly, ε_i: the random parameter, δ: the scaling factor, β_0: the initial brightness, and m: a positive number. One of the advantages of FIFA is the diversity of its solutions. The first step involves determining the initial location of fireflies. In the second step, each firefly's objective function is computed. Each firefly's location is updated using Eq. (13). The process contin-

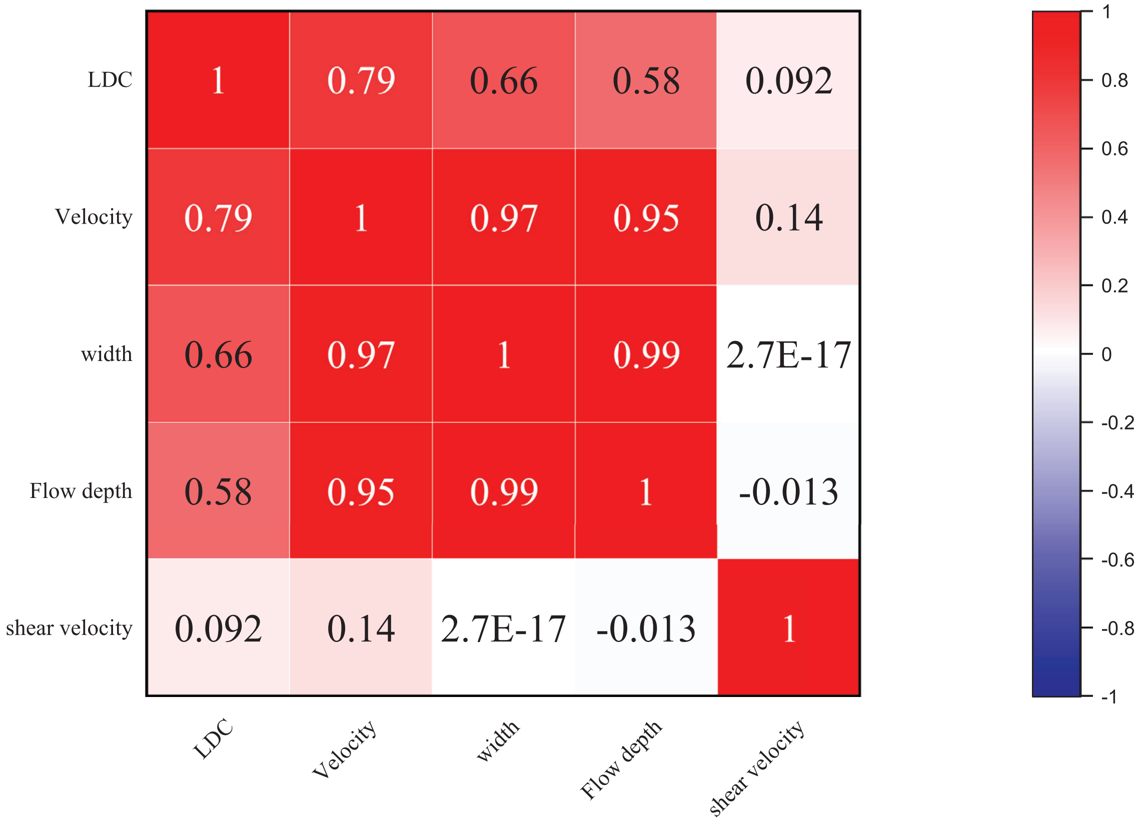
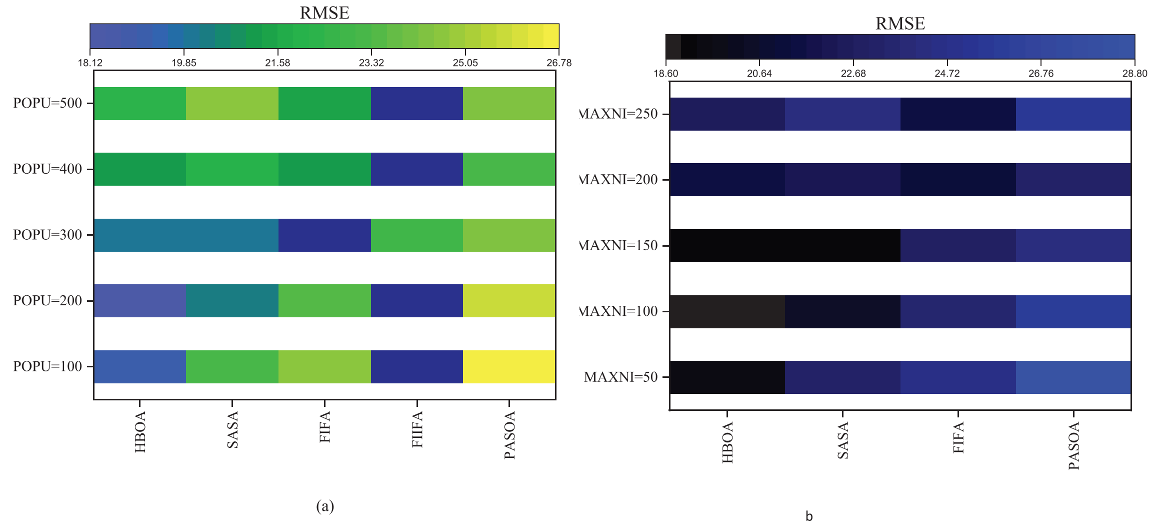


Fig. 7. Pearson correlation coefficient between inputs and outputs.



(a)

b

Fig. 8. Sensitivity Analysis for a: Population size and b: maximum number of iterations.

ues until the stop condition is satisfied. The structure of SASA is shown in Fig. 4.

2.2.4. Structure of particle swarm optimization (PASOA)

PASOA is an optimization method for continuous nonlinear functions. [17]. With PASOA, the memory usage is lower, while the learning speed is higher [3]. The algorithm can simulate cooperation and competition among individuals in a population of particles. Particles' positions and velocities are randomly altered during each iteration to accelerate them towards their personal best and global locations. The following equations update the position and velocity of the particles:

$$VE_{i,d}(t+1) = \omega VE_{i,d}(t) + c_1 ran_1 (P_{i,d}(t) - LO_{i,d}(t)) + c_2 ran_1 (P_{g,d}(t) - LO_{i,d}(t)) \quad (14)$$

$$LO_{i,d}(t+1) = LO_{i,d}(t) + VE_{i,d}(t+1) \quad (15)$$

where $VE_{i,d}(t+1)$: the new velocity of particles, $VE_{i,d}(t)$: the velocity of the ith particle at iteration t, d: the dimension index, $P_{i,d}(t)$: the local best position, $P_{g,d}(t)$: the global solution, $LO_{i,d}(t+1)$: the location of the ith particle, ω: the inertia weight, and c_1 and c_2: the acceleration coefficients. Small inertia weights are used for local searches, while large inertia weights explore large search spaces. The structure of PASOA is shown in Fig. 5.

2.2.5. Hybrid ANN models

A key issue in modeling is selecting the optimal MULP parameters (weight and bias). The ANN parameters are optimized based on the following steps:

1) The MULP is run at the training stage.
2) If the stop condition is met, the MULP moves to the testing level; otherwise, it goes to the third level.
3) For an optimization problem, MULP parameters are the decision variables. The locations of agents in algorithms show the value of decision variables.
4) The locations of agents are updated using operators of optimization algorithms. MULP parameters are updated when the location of agents is updated.
5) The MULP runs based on the new parameters.
6) The RMSE (objective function) is used to determine the quality of solution.

7) The model goes to step 2 if the stop convergence occurs; otherwise, it goes to step 3.

2.2.6. Structure of IMM models

An IMM fosters collaboration among individual models [28]. So, the advantages of multiple individual models are incorporated into the structure of IMM. Thus, IMM improves individual models' accuracy. As seen in Fig. 6, the individual models (MULP-HBOA, MULP-SASA, MULP-PASOA, and MULP-FIFA) were run individually in the first step. Final results were obtained by integrating the outputs of individual models into an ANN. While there are several types of ensemble models, IMM has several advantages. Models like Bayesian model averaging require complex computations, but IMM can be easily implemented. The IMM model does not need prior and posterior distributions of variables. The number of input neurons were 5 based on Fig. 6. The number of hidden layers and hidden neuros were 1, 3.

3. Data set

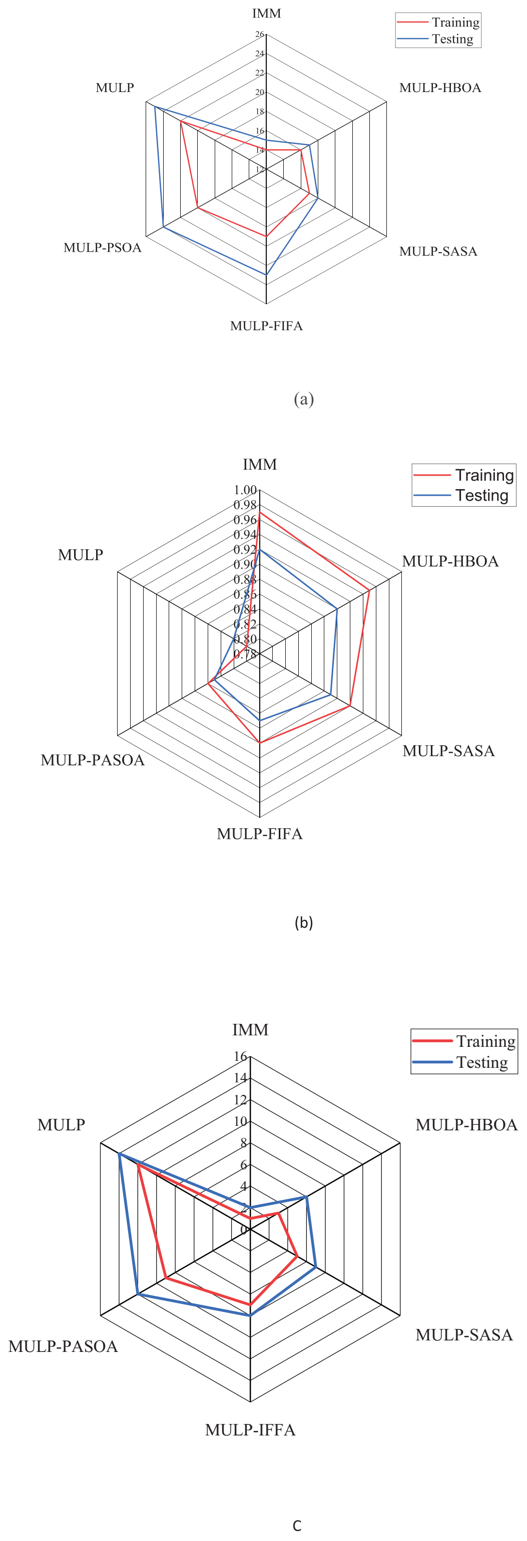
For predicting LDC, this study used optimized MULP and IMM models. Data were required for training and testing. Using the open-source literature, we collected data from 30 streams. The 495 sample points were split into 396 training points and 99 testing points. Four input variables, including channel width (CW), flow depth (FD), flow velocity (FV), and shear velocity (SV), had the greatest impact on LDC, as shown by previous studies. Thus, this study used CW, FD, FV, and SV to predict LDC. Table 1 shows the details of data points. The input data are normalized to use in the next levels.

Model accuracy was assessed using several indices. Table 2 shows the equations of indices.

4. Results and discussion

4.1. Determination of Pearson correlation coefficient between LDC and input data

This section examined the Pearson correlation coefficient between inputs and outputs. The heat map of correlation between inputs and LDC is shown in Fig. 7. Velocity and shear velocity showed the highest and lowest correlations with LDC, respectively. The previous studies confirm this finding. Gholami et al. [11] stated that the velocity and shear velocity had the highest correlation with LDC. This study used all inputs to achieve the best accuracy and the best performance.



(a)

(b)

c

Fig. 9. Radar plots for comparison of the models based on a) MAE, b: NSE, and c: PBIAS.

4.2. Determination of random parameters

An optimization algorithm's performance depends mainly on random parameters (RPs). This section determined the best values of RPs using sensitivity analysis. Sensitivity analysis shows how the objective function varies with random parameter values. For the best values of random parameters, the objective function is the lowest. The size of population and the MNI should be determined before the optimization process begins In Fig. 8a and 8b, we illustrate how we determine the size of the population and the maximum number of iterations (MAXNI) based on sensitivity analysis.

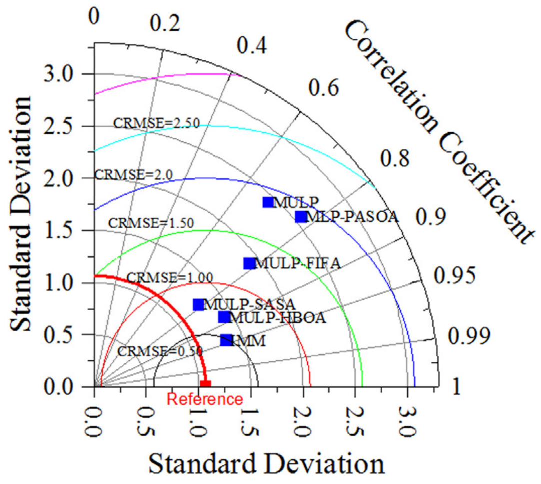


Fig. 10. Taylor diagram for evaluation of the accuracy of models.

The population size (POPU) of the HBOA ranged from 100 to 500. With POPU = 200, the HBOA had the lowest objective function value. The lowest objective functions were provided by POUP = 300, POPU = 400, and POPU = 400, respectively, for SASA, FIFA, and PASOA. The MNI of the HBOA ranged from 50 to 200. With POPU = 100, the HBOA had the lowest objective function value. MAXNI = 150, MAXNI = 200, and MAXNI = 200 provided the lowest objective functions for PASOA, FIFA, and SASA.

4.3. Investigation of the accuracy of models

This section investigates the accuracy of models and determines the best model with the highest accuracy. The MEAE values for the models are shown in Fig. 9a. The IMM decreased the MEAE of trained MULP-HBOA, MULP-SASA, MULP-FIFA, MULP-PASOA, and MULP models by 12%, 17%, 26%, 30%, and 36%, respectively. For the IMM, the MEAE during testing was 15, whereas it was 17, 18, 23, 24, and 25 for the MULP-HBOA, MLP-SASA, MULP-FIFA, MULP-PASOA, and MULP models. Among optimization algorithms, HBOA gave the best accuracy. MULP-HGBOA decreased training MEAEs of MULP-SASA, MULP-FAFA, and MULP-PASOA by 5.8%, 15%, and 25%, respectively, MULP-HBOA decreased testing MEAEs of MULP-SASA, MULP-FAFA, and MULP- PASOA by 5%, 26%, and 30%, respectively. Since a honeyguide bird helps search agents escape local optimums, HBOA performed better than the other optimization algorithms. The operators of optimization algorithms affect accuracy. An algorithm that uses advanced operators is the most accurate.

The NSE values for the models are shown in Fig. 9b. MULP and IMM had the lowest and highest NSE at the training level. At the testing level, the NSE of the IMM, MULP-HBOA, MULP-SASA, MULP-FIFA, MULP-PASOA, and MULP models was 0.92, 0.90, 0.89, 0.87, 0.85, and 0.82, respectively. The PBIAS values for the models are shown in Fig. 9c. At the testing level, PBIAS of the IMM, MULP-HBOA, MULP-SSA, MLP-FIFA, MULP-PASOA, and MULP models was 2, 6, 7, 8, 12, 12, 14%, respectively.

A Taylor diagram graphically evaluates the accuracy of models using CRMSE, correlation coefficients, and normalized standard deviations (Fig. 10). The correlation coefficients of the IMM, MULP-HBOA, MULP-SASA, MULP-FIFA, MULP-PASOA, and MULP

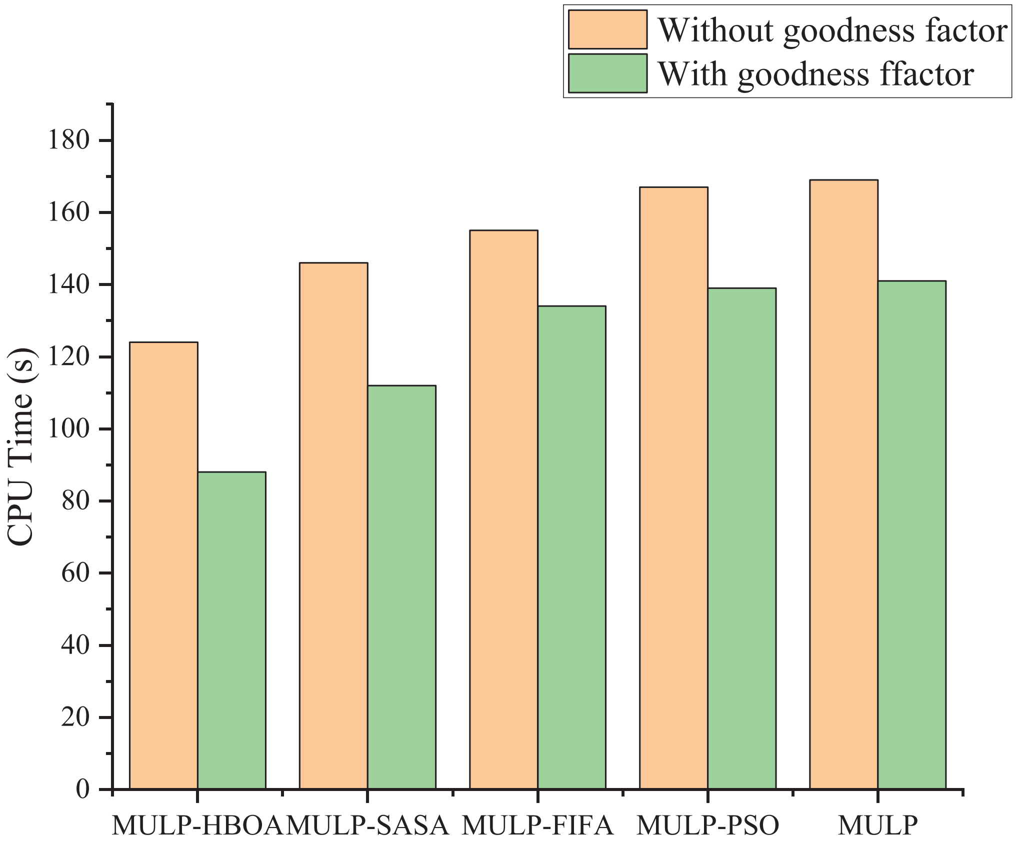


Fig. 11. CPU times of models with and without goodness factor.

model was 0.94, 0.88, 0.78, 0.77, and 0.68. The IMM and the MULP models had the best and worst performances among other models. The closet models to the reference point (Observed data) are ideal. As can be seen from Fig. 10, The HBOA outperformed other algorithms because HBOA used advanced operators for local and global searches. The traditional MULP models did not achieve the high precision because it did not use advanced tool for setting parameters.

The CPU time of models with and without goodness factors is shown in Fig. 11. CPU times for MULP-HBOA, MULP-SASA, MULP-FIFA, MULP-PSOA, and MULP were reduced from 124 to 88 s, 146 to 112 s, 155 to 134 s, 167 to 139 s, and 169 to 141 s by employing the goodness factor. In general, these results were found:

1) The IMM was better than the hybrid and standalone MULP models. Different advantages of MULP are incorporated into the structure of IMM. As IMM uses the advantages of multiple MULP models, it can achieve the most accurate results.
2) MULP produced the poorest outputs due to its lack of advanced algorithms for adjusting its parameters. The results of MULP indicated that advanced algorithms are essential for getting the most accurate results.
3) 3)There are differences between the outputs of HBOA and other optimization algorithms. HBOA is more accurate because it uses more advanced operators.

4.4. Discussion

The IMM and optimized MULP models were used to estimate LDC in this study. Previous studies attempted to predict LDC. For predicting LDC, Memarzadeh et al. [18] developed a whale optimization algorithm with a MAE of 46.930. Using the IMM model, the current study increased model accuracy by 63%.

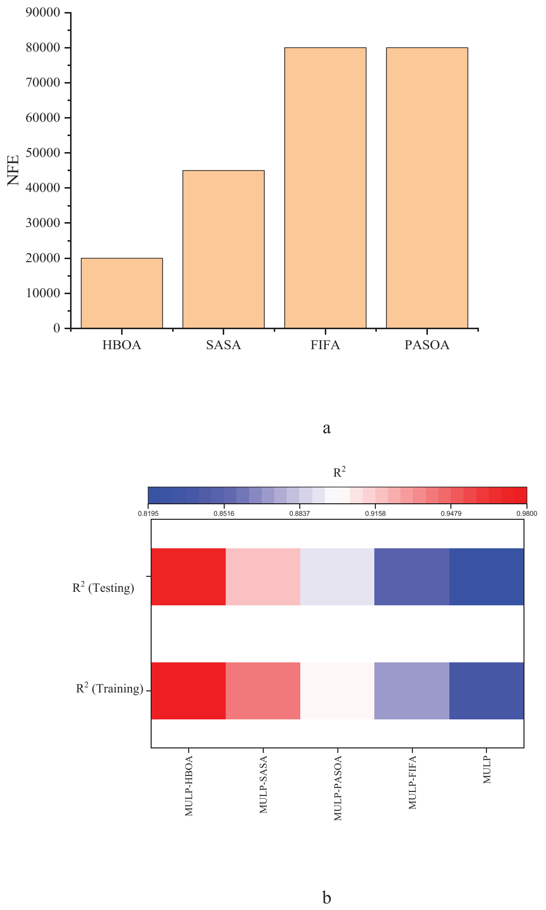


Fig. 12. a: Values of nfe for different algorithms, b: R^2 values of models.

Gholami et al. [11] merged multiple intelligent models to predict LDC. The committee machine had a MAE of 19.026. Using the IMM model, the current study increased model accuracy by 10%. Overall, the IMM performed better than previous models. Using the IMM model, the best accuracy is achieved by combining the advantages of multiple individual models. At the same time, the IMM model uses no complex process to estimate target variables. The previous studies confirmed IMM models' high accuracy.

Norouzi et al. [23] reported that IMM was the best model for estimating the oblique weir discharge coefficient. Shabani et al. [28] found that the IMM model with a correlation coefficient of 0.81 and an RMSE of 0.69 accurately predicted CO2 emissions. The paper also showed that optimization algorithms improved model accuracy. An optimization algorithm has advantages and disadvantages. The optimization algorithms use different operators to solve optimization problems. As a result, they provide different levels of accuracy. HBOA was found to be the most accurate optimization algorithm in this study. HBOA's advanced operators provided the best results.

Individual models can be guided to promising exploration areas using the intensity parameter (I). The honey and digging phases improve the solution update process by balancing exploration and exploitation. The number of functional evaluations (NFE) (NFE = Population size* the maximum number of iterations) is an

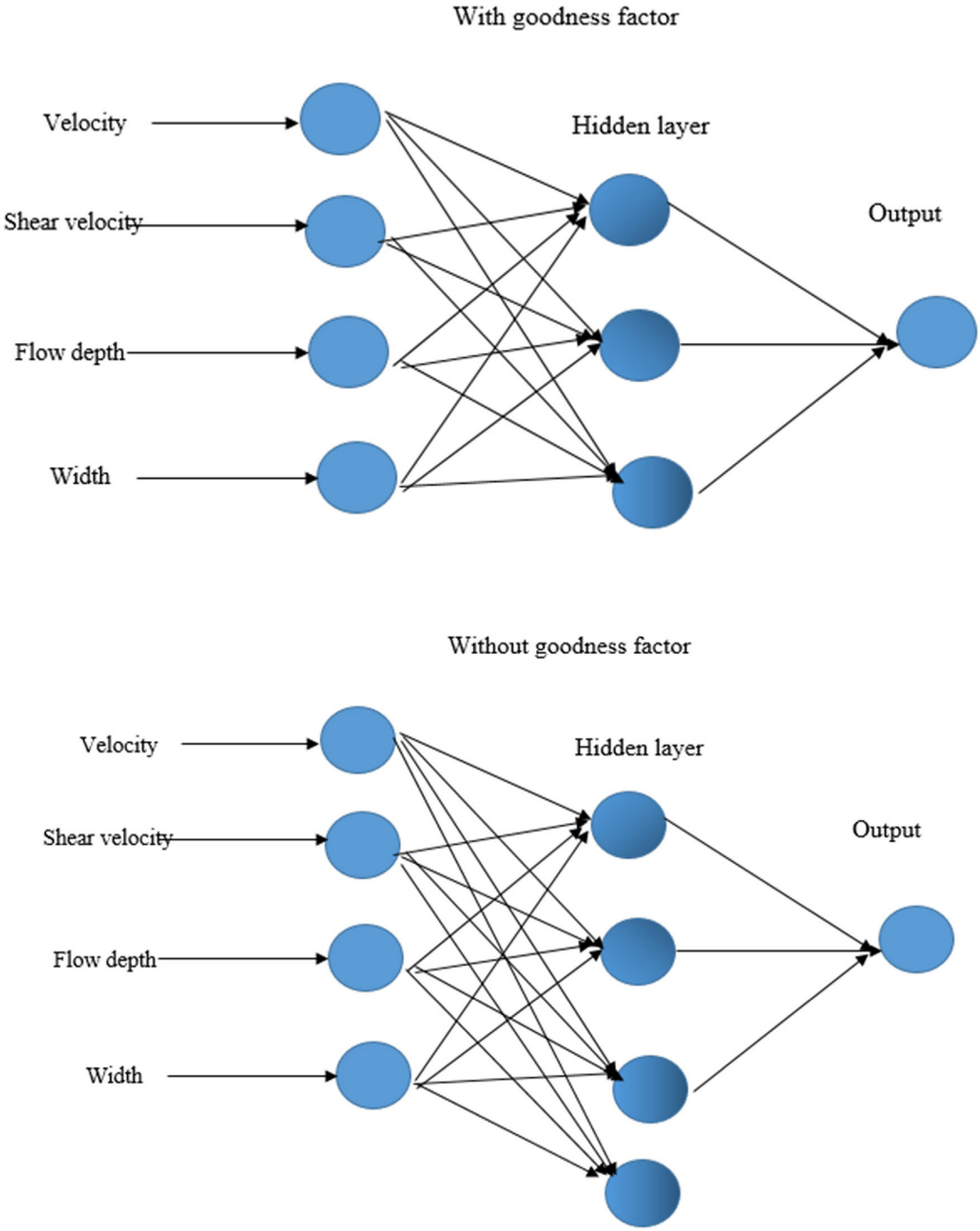


Fig. 13. Structure of MLP with and without goodness factor.

Table 3
Comparison of the accuracy of IMM with previous studies.

Author	Model	Results

Wang and Huai [29]	Regression analysis	MEAE = 49.68
		NSE = 0.68

Rezaei Balf et al. [5]	Evolutionary polynomial regression	MEAE = 47.076
		NSE = 0.776

Ghiasi et al. [10]	ANN	MAE = 60.98
Ghiasi et al. [10]	ANFIS	MAE = 34.87
Memarzadeh et al. [18]	Whale optimization algorithm	MAE = 26.84–198
Arya Azar et al. [4]	ANFIS	R^2 = 0.91
Gholami et al. [11]	Optimized ANN	R^2 = 0.91
Gholami et al. [11]	Optimized ANFIS	R^2 = 0.90
Gholami et al. [11]	Optimized support vector machine	R^2 = 0.94
Gholami et al. [11]	committee machine	R^2 = 0.94

index for evaluating optimization algorithms. With the lowest NFE, the algorithm had the lowest computational cost. The NFE for different algorithms is shown in Fig. 12. The HBOA and SSA had the lowest NFEs, as displayed in Fig. 12a. Thus, HBOA outperformed other algorithms. Fig. 12b shows the R^2 values of different models. The testing R^2 value of MULP-HBOA, MUL-SASA, MULP-FIFA, MULP-PSOA, and MLP models were 0.97, 0.92, 0.89, 0.85, and 0.82, respectively.

Fig. 13 shows the effect of the goodness factor on the structure of an MULP model. The goodness factor decreases the hidden units of hidden layers, as shown in Fig. 13. The CPU time decreases when the number of redundant units decreases. Thus, a goodness factor is an effective method to modify the structure of ANN model because it can decrease CPU time and redundant neurons. The modified and unmodified structure of ANN model had four input neurons. The number of hidden neuron of unmodified ANN models was higher the number of hidden neurons of modified ANN models. Table 3 compares the accuracy of IMM with other models.

Using a regression model, Wang and Huai [29] predicted LDC. Specifically, their model had an MAE of 49.68 and an NSE of 0.68. The IMM model had NSE and MAE of 0.97 and 15, respectively. The IMM performed better than the regression model. Balf et al. [5] estimated LDC using an evolutionary polynomial regression model. The MEAE and NSE were 47.07 and 0.77, respectively. Therefore, the IMM performed better than their model. Ghiasi et al. [10] used an ANN to estimate LDC. The ANN model had an MAE of 60.98. Therefore, IMM had better performance than their model. Memarzadeh et al. [18] predicted LDC using a whale optimization algorithm. During the calibration and verification periods, their MEAE ranged from 26.94 to 198. The IMM model had a lower MEAE than the whale optimization algorithm. Gholami et al. [11] estimated LDC based on a support vector machine model. The R^2 of their model was 0.94. The IMM model showed an R^2 of 0.98 and 0.97 during training and testing. The IMM improved the results of previous research in this section.

5. Conclusion

Because open channels have a complex mixing mechanism, determining LDC can be difficult. The highly stochastic and nonlinear nature of LDC has made its estimation challenging. This study estimated the LDC in natural streams using ensemble models and optimized MULPs. As an ensemble model, IMM was created using the outputs of individual models. MULP parameters were adjusted using HBOA as a new optimization algorithm. The models' input included flow width, depth, velocity, and shear velocity. The IMM decreased the MEAE of MULP-HBOA, MULP-SASA, MULP-FIFA, MULP-PASOA, and MULP models by 12%, 17%, 26%, 30%, and 36%, respectively, at the training level. In this article, the ensemble models provided better prediction accuracy than the individual models. Unlike other ensemble models such as Bayesian model averaging, which may be extremely difficult to develop, the IMM can be easily implemented to predict LDC Also, the paper showed that optimization algorithms enhanced the precision of standalone MULPs.

In future studies, other ensemble models can be used to predict LDC. The next study should also consider the impact of input uncertainty and model parameters on the output data.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Alizadeh MJ, Shabani A, Kavianpour MR. Predicting longitudinal dispersion coefficient using ANN with metaheuristic training algorithms. Int J Environ Sci Technol 2017. doi: https://doi.org/10.1007/s13762-017-1307-1.

[2] Ahmad Z. Prediction of longitudinal dispersion coefficient using laboratory and field data: relationship comparisons. Hydrol Res 2013;44(2):362–76.

[3] Ansari HR, Gholami A. An improved support vector regression model for estimation of saturation pressure of crude oils. Fluid Phase Equilib 2015;402:124–32.

[4] Arya Azar N, Ghordoyee Milan S, Kayhomayoon Z. The prediction of longitudinal dispersion coefficient in natural streams using LS-SVM and ANFIS optimized by Harris hawk optimization algorithm. J Contam Hydrol 2021. doi: https://doi.org/10.1016/j.jconhyd.2021.103781.

[5] Balf MR, Noori R, Berndtsson R, Ghaemi A, Ghiasi B. Evolutionary polynomial regression approach to predict longitudinal dispersion coefficient in rivers. J Water Supply Res Technol AQUA 2018. doi: https://doi.org/10.2166/aqua.2018.021.

[6] Carr ML, Rehmann CR. Measuring the dispersion coefficient with acoustic Doppler current profilers. J Hydraul Eng 2007;133(8):977–82.

[7] Deng ZQ, Singh VP, Bengtsson L. Longitudinal dispersion coefficient in straight rivers. J Hydraul Eng 2001;127(11):919–27.

[8] Ehteram M, Ferdowsi A, Faramarzpour M, Al-Janabi AMS, Al-Ansari N, Bokde ND, et al. Hybridization of artificial intelligence models with nature inspired optimization algorithms for lake water level prediction and uncertainty analysis. Alex Eng J 2021;60(2):2193–208.

[9] Ghaemi A, Zhian T, Pirzadeh B, Hashemi Monfared S, Mosavi A. Reliability based design and implementation of crow search algorithm for longitudinal dispersion oefficient estimation in rivers. Environ Sci Pollut Res 2021. doi: https://doi.org/10.1007/s11356-021-12651-0.

[10] Ghiasi B, Sheikhian H, Zeynolabedin A, Niksokhan MH. Granular computing-neural network model for prediction of longitudinal dispersion coefficients in rivers. Water Sci Technol 2019. doi: https://doi.org/10.2166/wst.2020.006.

[11] Gholami A, Amirpour M, Ansari HR, Seyedali SM, Semnani A, Golsanami N, et al. Porosity prediction from pre-stack seismic data via committee machine with optimized parameters. J Pet Sci Eng 2022;210:110067.

[12] Goliatt L, Sulaiman SO, Khedher KM, Farooque AA, Yaseen ZM. Estimation of natural streams longitudinal dispersion coefficient using hybrid evolutionary machine learning model. Eng Applications Computational Fluid Mech 2021. doi: https://doi.org/10.1080/19942060.2021.1972043.

[13] Hashim FA, Houssein EH, Hussain K, Mabrouk MS, Al-Atabany W. Honey Badger Algorithm: new metaheuristic algorithm for solving optimization problems. Math Comput Simul 2022. doi: https://doi.org/10.1016/j.matcom.2021.08.013.

[14] Khatibi R, Nadiri AA. Inclusive Multiple Models (IMM) for predicting groundwater levels and treating heterogeneity. Geosci Front 2021;12(2):713–24.

[15] Kashefipour SM, Falconer RA. Longitudinal dispersion coefficients in natural channels. Water Res 2002;36(6):1596–608.

[16] Matsunaga Y. A modified back propagation algorithm that automatically removes the redundant hidden units by competition. IEICE Trans Inf & Syst 1995;79(3):403–12.

[17] Kennedy J, Eberhart, RC. Particle swarm optimization. Paper IEEE 1995.

[18] Memarzadeh R, Zadeh HG, Dehghani M, Riahi-Madvar H, Seifi A, Mortazavi SM. A novel equation for longitudinal dispersion coefficient prediction based on the hybrid of SSMD and whale optimization algorithm. Sci Total Environ 2020;716:137007.

[19] Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 2017;114:163–91.

[20] Nezaratian H, Zahiri J, Kashefipour SM. Sensitivity analysis of empirical and data-driven models on longitudinal dispersion coefficient in streams. Environ Process 2018. doi: https://doi.org/10.1007/s40710-018-0334-3.

[21] Noori R, Deng Z, Kiaghadi A, Kachoosangi FT. How reliable are ANN, ANFIS, and SVM techniques for predicting longitudinal dispersion coefficient in natural

rivers? J Hydraul Eng 2016. doi: https://doi.org/10.1061/(asce)hy.1943-7900.0001062.

[22] Noori R, Mirchi A, Hooshyaripor F, Bhattarai R, Torabi Haghighi A, Kløve B. Reliability of functional forms for calculation of longitudinal dispersion coefficient in rivers. Sci Total Environ 2021. doi: https://doi.org/10.1016/j.scitotenv.2021.148394.

[23] Norouzi R, Arvanaghi H, Salmasi F, Farsadizadeh D, Ghorbani MA. A new approach for oblique weir discharge coefficient prediction based on hybrid inclusive multiple model. Flow Meas Instrum 2020. doi: https://doi.org/10.1016/j.flowmeasinst.2020.101810.

[24] Panahi F, Ehteram M, Ahmed AN, Huang YF, Mosavi A, El-Shafie A. Streamflow prediction with large climate indices using several hybrid multilayer perceptrons and copula Bayesian model averaging. Ecol Ind 2021;133:108285.

[25] Parsaie A, Haghiabi AH. Predicting the longitudinal dispersion coefficient by radial basis function neural network. Modeling Earth Syst Environ 2015. doi: https://doi.org/10.1007/s40808-015-0037-y.

[26] Riahi-Madvar H, Ayyoubzadeh SA, Khadangi E, Ebadzadeh MM. An expert system for predicting longitudinal dispersion coefficient in natural streams by using ANFIS. Expert Syst Appl 2009;36(4):8589–96.

[27] Riahi-Madvar H, Dehghani M, Parmar KS, Nabipour N, Shamshirband S. Improvements in the explicit estimation of pollutant dispersion coefficient in rivers by subset selection of maximum dissimilarity hybridized with ANFIS-Firefly Algorithm (FFA). IEEE Access 2020. doi: https://doi.org/10.1109/ACCESS.2020.2979927.

[28] Shabani E, Hayati B, Pishbahar E, Ghorbani MA, Ghahremanzadeh M. A novel approach to predict CO_2 emission in the agriculture sector of Iran based on Inclusive Multiple Model. J Clean Prod 2021. doi: https://doi.org/10.1016/j.jclepro.2020.123708.

[29] Wang Y, Huai W. Estimating the longitudinal dispersion coefficient in straight natural rivers. J Hydraul Eng 2016;142(11):04016048.

[30] Watada J. Structural learning of neural networks for forecasting stock prices, in: International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, Springer, Berlin, Heidelberg, 2006, October, p. 972-79.

[31] Yang XS. Firefly algorithms for multimodal optimization. In International symposium on stochastic algorithms, Springer, Berlin, Heidelberg, 2009, October, p. 169-78.