Progress in Earth and
Planetary Science

RESEARCH ARTICLE Open Access

Development of particle swarm clustered optimization method for applications in applied sciences

Amin Mahdavi-Meymand and Wojciech Sulisz*

Abstract

An original particle swarm clustered optimization (PSCO) method has been developed for the implementations in applied sciences. The developed PSCO does not trap in local solutions in contrary to corresponding solutions obtained by the applications of particle swarm optimization algorithm that is frequently used in many disciplines of applied sciences. The integrations of PSCO with multilayer perceptron neural network, adaptive neuro-fuzzy inference system (ANFIS), linear equation, and nonlinear equation were applied to predict the Vistula river discharge. The performance of PSCO was also compared with autonomous groups particle swarm optimization, dwarf mongoose optimization algorithm, and weighted mean of vectors. The results indicate that the PSCO has no tendency to trap in local solutions and its global solutions are more accurate than other algorithms. The accuracy of all developed models in predicting river discharge was acceptable ($R^2 > 0.9$). However, the derived nonlinear models are more accurate. The outcome of thirty consecutive runs shows that the derived PSCO improves the performance of machine learning techniques. The results also show that ANFIS-PSCO with $RMSE = 108.433$ and $R^2 = 0.961$ is the most accurate model.

Keywords River discharge, Swarm optimization, Neural network, ANFIS, PSCO

1 Introduction

Prediction of river discharge is essential for planning environmental programs and water management projects (Shakti and Sawazaki 2021). Linear stochastic models are commonly and predominantly used methods for the prediction of hydrological and environmental events (Papacharalampous et al. 2019). Most environmental problems, are complex problems and nonlinear methods, as more accurate, have been recommended to be applied to achieve solutions. Machine learning (ML) models are robust nonlinear methods that have been widely applied to solve a number of engineering and scientific problems.

Dibike and Solomatine (2001) used two types of artificial neural networks (ANNs) including multilayer perceptron network (MLP) and radial basis function network (RBF) for river flow forecasting. The results showed that the ANNs models are slightly better than the conceptual rainfall–runoff model. Solomatine and Dulal (2003) showed that the ANNs are slightly better than model trees (MTs) in rainfall–runoff modeling. Behzad et al. (2009) investigated the performance of support vector machine (SVM) in runoff modeling and showed that the prediction accuracy of SVM is close to the ANNs models. Badrzadeh et al. (2013) developed wavelet neural networks (WNN) and wavelet neuro-fuzzy (WNF) models for forecasting river flow. The study indicated that the results of hybrid models are significantly better than the outcome of the original ANN and ANFIS models. FajardoToro et al. (2013) developed a hybrid case-based reasoning (CBR) model for river flow forecasting. The

*Correspondence:
Wojciech Sulisz
sulisz@ibwpan.gda.pl
Institute of Hydro-Engineering, Polish Academy of Sciences, Gdańsk, Poland

results showed the superiority of the developed methods over the ANNs models. Daliakopoulos and Tsanis (2016) showed that the performance of ANN models in the simulation of ephemeral streamflow is higher than the performance of conceptual models (CM). Bomers et al. (2019) applied an ANN model for the reconstruction of historic floods. The results confirm the capability of ANN models to predict complex hydraulic phenomena. Linh et al. (2021) investigated the application of a wavelet neural network (WNN) for the prediction of river discharge. The results showed that the WNN is more accurate than ANNs. Gauch et al. (2021) applied long short-term memory (LSTM) and tree models (TR) for the prediction of streamflow and showed the superiority of the LSTM over the tree model.

All ML models have some coefficients that must be calculated and optimized during a training process. Recent studies have indicated that the application of meta-heuristic algorithms, as a training approach for optimizing part or all of these coefficients, increases the performance of original models. Azad et al. (2018) applied ANFIS integrated with ant colony optimization (ACOR), particle swarm optimization (PSO), and genetic algorithm (GA) to simulate river flow. The results demonstrated the ability of optimization algorithms to increase the performance of original ANFIS and showed the advantages of PSO over ACOR and GA. Yaseen et al. (2019) applied ANFIS integrated with PSO, GA, and differential evolution (DE) to forecast river flow. The results confirmed the advantages of PSO over GA and DE. Zounemat-Kermani et al. (2021) used several ML models including ANNs, ANFIS, LSTM, group method of data handling (GMDH), wavelet neural network (Wavenet), and ANN integrated with PSO and GA for forecasting river flow. The outcome indicated that the integrated models result in better forecasting than the original methods. Arora et al. (2021) applied ANFIS integrated with PSO, GA, and DE for flood susceptibility prediction mapping. The results showed the advantages of ANFIS-GA over ANFIS-PSO and ANFIS-DE.

In recent years various types of meta-heuristic algorithms have been introduced by researchers. Recently developed algorithms comprise moth-flame optimization (MFO) (Mirjalili 2015), whale optimization algorithm (WOA) (Mirjalili and Lewis 2016), butterfly optimization algorithm (BOA) (Arora and Singh 2019), black widow optimization (BWO) (Hayyolalam and Pourhaji Kazem 2020), carnivorous plant algorithm (CPA) (Ong et al. 2021), Poplar optimization algorithm (Chen et al. 2022), etc.

It has been claimed that the performance of new algorithms is better than the corresponding performance of widely recognized PSO or GA, especially in obtaining global solutions for benchmark functions. However, several approaches, especially applications of ML models integrated with optimization algorithms, indicated that the new algorithms are often overrated. Zounemat-Kermani and Mahdavi-Meymand (2019) indicated that the performance of the MFO as integrative method with ANFIS and ANN for predicting the piano key weir discharge is worse than PSO, and GA. Zounemat-Kermani and Mahdavi-Meymand (2021) compared the ability of GA, PSO, firefly algorithm (FA) with MFO and WOA as integrative method with ANFIS for predicting hydraulic jump length and height. The results indicated that PSO, GA, and FA act better than new algorithms (WOA and MFO). Memar et al. (2021) applied BWO and PSO as integrative methods with ANFIS and support vector regression (SVR) models to predict the maximum wave height of the Baltic Sea. The results indicated that PSO acts better than the BWO in high-dimensional problems (ANFIS).

Although some researches indicated that the new optimization algorithms are better than widely recognized traditional models (Fadaee et al. 2020; Milan et al. 2021; Aalimahmoody et al. 2021; Fattahi and Hasanipanah 2022), there is still a space for improving traditional models. Several modifications of PSO have been introduced in the literature. Kennedy (1999) tested the effect of different neighborhood topologies such as circles, wheels, stars, and random edges on PSO performance. The results indicated that the performance of PSO substantially depends on the neighborhood topology. Bergh and Engelbrecht (2002) investigated the stagnation weakness of PSO and proposed a different equation for the global best particle movement. Mendes et al. (2003) and Huang and Mohan (2005) suggested that the swarms receive information and knowledge only from the part of neighborhood populations. They modify PSO and the outcome indicated that the new approach results in better performance. Liang and Suganthan (2005) introduced a dynamic multi-swarm particle swarm optimizer. In this approach, the population is divided into several independent subgroups. The sub-swarms are regrouped using different operations. Lim and Isa (2014) purposed PSO with increased connectivity (PSO-ITC). In this approach, initially, each member of a population is connected to a randomly selected agent of a population. By increasing the number of iterations of the algorithm, the particle connections are increasing gradually till members are fully connected. Tsujimoto et al. (2012) analyzed the application of deterministic PSO (DPSO) to find a global solution. The DPSO approach was introduced to eliminate the stochastic factors. The DPSO improves traditional approaches to finding a global solution.

Bonyadi et al. (2014) introduced hybrid PSO variants with a time-adaptive approach. In this approach, the population is divided into several sub-swarms and optimization processes are applied. Suryanto et al. (2017) introduced multi-group particle swarm optimization with random redistribution (MGRR-PSO) algorithm. In the MGRR-PSO, the population is divided into two groups with different acceleration coefficients. Lv et al. (2018) introduced the improved eliminate particle swarm optimization (IEPSO) algorithm which works based on the last-eliminated principle. Song et al. (2021) purposed a new version of PSO and showed the superiority of improved PSO in benchmark functions and for the smooth path planning of mobile robots.

The PSO is a fast, recognized, and widely used meta-heuristic algorithm. The main weakness of the PSO is its tendency to trap in the local solutions (Rehman et al. 2019; Tu et al. 2020). In this study, an original particle swarm clustered optimization (PSCO) method is derived to overcome this weakness. First, the particle swarm clustered optimization (PSCO) technique is developed. Ten mathematical benchmark functions were selected to analyze the performance of PSCO. Each benchmark function predicts an applied or scientific problem. The results obtained by the application of PSCO are compared with the results of PSO and recently developed meta-heuristic algorithms including autonomous groups particles swarm optimization (AGPSO), dwarf mongoose optimization algorithm (DMOA), and weighted mean of vectors (INFO). Then, the derived method is used as a training algorithm to optimize ANFIS, MLPNN, linear equation (LE), and nonlinear equation (NE) for the prediction of river discharge. The tuning the ANFIS, MLPNN, and regression equations coefficients represent a high-, moderate-, and low-dimensional optimization problems, respectively. The performance of the derived algorithms is evaluated based on the outcome of 30 consecutive runs. Finally, the stabilities of the derived models in reaching a global solution are analyzed and conclusions are specified.

2 Materials and methods

2.1 Particle swarm optimization

Particle swarm optimization (PSO) is a population-based stochastic optimization algorithm introduced by Eberhart and Kennedy (1995). The PSO algorithm is inspired by the behavior of flocking birds or fishes. In general, the meta-heuristic algorithms are divided into two categories, namely swarm and evolutionary algorithms. The PSO is a swarm computation algorithm. The structure of evolutionary algorithms comprises crossover and mutation evolution operators. The swarm algorithms use another computation process to find a solution. In the PSO, each particle in a search space is identified by velocity and position vectors. At the first iteration of PSO, all particles in a search space are distributed randomly, which means that the particle position vector is randomly initialized. The velocity of particles at the first iteration is equal to zero. In each iteration, the values of particle position and velocity vectors are updated based on the following equations:

$$v_i^{t+1} = wv_i^t + C_1 \times r_1 \times \left(P_i^{\text{best}} - x_i^t\right) + C_2 \times r_2 \times \left(G^{\text{best}} - x_i^t\right) \tag{1}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{2}$$

where v is the velocity vector, x is the position vector, t is the iteration, r_1 and r_2 are the random number between 0 to 1, w is the inertial weight, P_i^{best} is the best position achieved by the particle i, G^{best} is the best position of the particles, and C_1 and C_2 are the personal learning and global learning coefficients, respectively.

2.2 Particle swarm clustered optimization

The PSO is a fast, recognized, and widely used meta-heuristic algorithm. The main weakness of the PSO is its tendency to trap in a local solution, and in consequence, it fails to reach the global solution (Rehman et al. 2019; Tu et al. 2020). Particle swarm clustered optimization (PSCO) is a novel technique derived in this study to overcome the weaknesses of PSO. In PSCO, the particles are divided into m clusters. Up to a specified iteration, I_m, each cluster follows the PSO procedure to find a solution. In the following iterations, the particles increase their knowledge using other particle knowledge. In other words, in the first stage particles move toward the cluster leader, and in the next stage, they move toward the best particle of the whole population. With this strategy, the clusters at I_m iteration are close to different local solutions. The strategy of moving particles from the local solutions to the best particle causes that the algorithm overcomes trapping at local solutions. The particle velocity vectors are updated based on the following equations:

$$\begin{aligned} v_{i,j}^{t+1} =& wv_{i,j}^t + C_1 \times r_1 \times \left(P_{i,j}^{\text{best}} - x_{i,j}^t\right) + C_2 \\ & \times r_2 \times \left(G_j^{\text{best}} - x_{i,j}^t\right) \quad \text{If } t \le I_m \end{aligned} \tag{3a}$$

$$\begin{aligned} v_{i,j}^{t+1} =& wv_{i,j}^t + C_1 \times r_1 \times \left(P_{i,j}^{\text{best}} - x_{i,j}^t\right) + C_2 \\ & \times r_2 \times \left(G^{\text{best}} - x_{i,j}^t\right) \quad \text{If } t > I_m \end{aligned} \tag{3b}$$

where $P_{i,j}^{\text{best}}$ is the best-observed position of ith particle of jth cluster, G_j^{best} is the position of the leader of jth cluster, and G^{best} is the best particle position. The position of particles is updated according to the following equation:

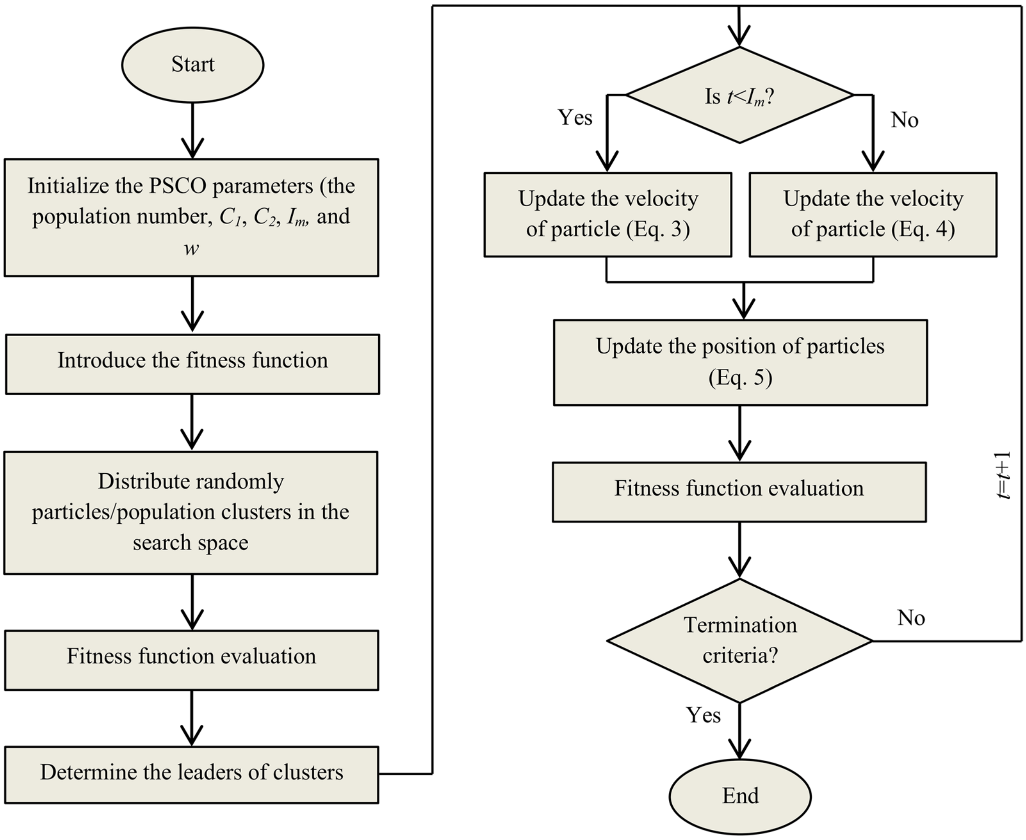
Start
Initialize the PSCO parameters (the population number, C_1, C_2, I_m, and w
Introduce the fitness function
Distribute randomly particles/population clusters in the search space
Fitness function evaluation
Determine the leaders of clusters
Is $t<I_m$?
Yes
No
Update the velocity of particle (Eq. 3)
Update the velocity of particle (Eq. 4)
Update the position of particles (Eq. 5)
Fitness function evaluation
Termination criteria?
No
Yes
End
$t=t+1$

Fig. 1 The flowchart of PSCO algorithm

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1} \tag{4}$$

Figure 1 shows the flowchart of PSCO algorithm. It is worth noting that PSCO topology, contrary to topologies applied by, e.g., Liang and Suganthan (2005) and Bonyadi, et al. (2014), assumes that the size or shape of groups/clusters is constant during the iterations. Moreover, Liang and Suganthan (2005) and Bonyadi, et al., (2014) use the G^{best} topology during the iterations to update the positions of particles, while PSCO uses leader topology up to a specified iteration, and then, it is switched to the G^{best} for the rest of iterations.

2.3 Other meta-heuristic algorithms

Three other algorithms, namely autonomous groups particles swarm optimization (AGPSO), dwarf mongoose optimization algorithm (DMOA), and weighted mean of vectors (INFO), are used to compare and verify the performance of PSCO. The AGPSO is a variant of PSO developed by Mirjalili et al. (2014). The AGPSO assumes that the individuals of the population are not similar. The population is divided into four groups. The individual learning and global learning coefficients, C_1 and C_2, are different and are updated in an iteration procedure by applying the following equations:

$$C_1 = 1.95 - 2\left(\frac{\text{It}}{\text{MaxIt}}\right)^{\frac{1}{3}}, \quad C_2 = 0.05 + 2\left(\frac{\text{It}}{\text{MaxIt}}\right)^{\frac{1}{3}} \quad \text{Group I} \tag{5}$$

$$C_1 = 2.5 - 2\left(\frac{\text{It}}{\text{MaxIt}}\right)^{3}, \quad C_2 = 0.5 + 2\left(\frac{\text{It}}{\text{MaxIt}}\right)^{3} \quad \text{Group II} \tag{6}$$

$$C_1 = 1.95 - 2\left(\frac{\text{It}}{\text{MaxIt}}\right)^{1/3}, \quad C_2 = 0.5 + 2\left(\frac{\text{It}}{\text{MaxIt}}\right)^{3} \quad \text{Group III} \tag{7}$$

$$C_1 = 2.5 - 2\left(\frac{\text{It}}{\text{MaxIt}}\right)^{3}, \quad C_2 = 0.05 + 2\left(\frac{\text{It}}{\text{MaxIt}}\right)^{1/3} \quad \text{Group IV} \tag{8}$$

where MaxIt is the maximum number of iterations and it denotes the current iteration. The INFO is a novel algorithm introduced by Ahmadianfar et al. (2022). The INFO is developed based on the weighted mean of vectors. In this algorithm, each member of population is a vector. At the first step of INFO, the vectors are distributed randomly in the search domain. The vectors are updated by using the weighted mean strategy. The weight of two vectors is calculated as follows:

$$w = \cos\big(f(x_1) - f(x_2) + \pi\big) \times \exp\left(-\left|\frac{f(x_1) - f(x_2)}{\omega}\right|\right) \tag{9}$$

where x_1 and x_2 are vectors, *f* denotes the objective function, and ω is a coefficient.

Three vectors of population are chosen randomly. Three values of w between vectors are calculated. The mean of weights is used to generate two new vectors. In the vector combining stage, the generated vectors are combined in one vector by applying a specific random strategy. Finally, in the local search stage the new value of vectors is obtained. This stage is designed to prevent the vectors to trap into local solutions. More details of INFO can be found in Ahmadianfar et al. (2022).

The DMOA is a swarm-based optimization algorithm developed by Agushaka et al. (2022). The DMOA is a nature-inspired algorithm that models the behavior of dwarf mongoose in foraging. In the DMOA, the population is divided to the alpha, scouts, and babysitters groups. The number of population, *N*, and babysitters, N_b, must be initialized before optimization process. The number of alpha members, *n*, is *N*-N_b. The alpha group members are selected by probability index:

$$a = \frac{F_i}{\sum_{i=1}^{n} F_i} \tag{10}$$

where F is the fitness of the population. The position of candidates is generated by applying the following formula:

$$x^{t+1} = x^t + \varphi \times \delta \tag{11}$$

where δ is the parameter that must be initialized before iterations and φ is a random number between -1 and 1 that is uniformly distributed. The members of the scout group select a new place for sleeping according to:

$$x^{t+1} = \begin{cases} x^t - \mathrm{CF} \times \varphi \times r\big(x^t - M\big) & \text{if } \varphi^{t+1} > \varphi^t \\ x^t + \mathrm{CF} \times \varphi \times r\big(x^t - M\big) & \text{else} \end{cases} \tag{12}$$

where r is the random number between 0 and 1, M is a vector that controls the movement of agents to new sleeping positions, and CF is the parameter obtained from:

$$\mathrm{CF} = \left(1 - \frac{t}{\mathrm{It}_{\max}}\right)^{\frac{2t}{\mathrm{It}_{\max}}} \tag{13}$$

where t is the current iteration and $\mathrm{It}_{\max}$ is the maximum number of iterations. In the last step, the babysitters are exchanged. This step is repeated until the algorithm reaches the global solution.

2.4 Multilayer perceptron neural network

Multilayer perceptron neural network (MLPNN) is a recognized and widely used neural network algorithm. The MLPNN is a robust machine learning method that can be applied in the modeling of complex scientific and engineering problems. In general, the MLPNN consists of three layers including the input layer, middle or hidden layers, and output layer. Each layer is made of several nodes/neurons. The parameters of a model enter into the network through the input layer and its neurons. The output of $l+1$th middle layer neurons is calculated by applying the following equation:

$$y_j^{l+1} = f\left(b_j + \sum_{i=1}^{N} w_i x_i\right) \tag{14}$$

where *f* is the activation function, *b* is the bias coefficient, *w* is the weight coefficient, and *N* is the number of neurons of *l*th layer. Different types of activation functions such as sigmoid, hyperbolic tangent, and log-sigmoid may be used in the middle and output layers (Orhan et al. 2011). Like other ML models, the differences between the predicted and actual values are used for training the MLPNN. The MLPNN will turn to a deep network by increasing the number of middle layers.

2.5 Adaptive neuro-fuzzy inference system

Adaptive neuro-fuzzy inference system (ANFIS) is a hybrid machine learning model that combines artificial neural networks (ANNs) with fuzzy logic techniques. This combination has created an attractive method capable of solving complex problems. The ANFIS was introduced by Jang (1993), and so far it has been applied in numerous scientific and engineering problems. In general, the ANFIS forms a fuzzy logic, if–then rules, and uses neural networks for altering these rules and train a network. The ANFIS structure consists of 5 layers. The first layer is the fuzzification layer. The fuzzification process is based on membership functions (MFs), e.g., the trapezoidal, bell, triangular, or Gaussian functions. For an ANFIS with two inputs, the outputs may be defined as:

$$\mathrm{O}_i^1 = \mu_{A_i}(x_1) \quad i = 1,2 \tag{15}$$

$$\mathrm{O}_i^1 = \mu_{B_i}(x_2) \quad i = 3,4 \tag{16}$$

where x_1 and x_2 are the inputs to the *i*th node, and μ_A and μ_B are the membership functions. In this study, the

Gaussian and linear MFs were used for inputs and output parameters, respectively. In the second layer, the firing strength of the rules is calculated as follows:

$$\mathrm{O}_i^2 = w_i = \mu_{A_i}(x_1).\mu_{B_i}(x_2) \quad i = 1, 2 \tag{17}$$

The third layer is the normalized layer. The output of this layer is defined as:

$$\mathrm{O}_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2} \quad i = 1, 2 \tag{18}$$

The fourth layer output is calculated by applying the following formula:

$$\mathrm{O}_i^4 = \overline{w}_i f_i = \overline{w}_i(p_i x_1 + q_i x_2 + r_i) \quad i = 1, 2 \tag{19}$$

The output of the network is obtained in the fifth layer from the following formula:

$$\mathrm{O}_i^5 = \sum_i \overline{w}_i f_i \quad i = 1, 2 \tag{20}$$

2.6 Regression equations

The linear and nonlinear regression equations are simple and useful tools for solving regression problems. Actually, the regression equations (REs) are simple machine learning models. The application of meta-heuristic algorithms in REs training process makes them more accurate. In this study, the following equations are used to predict a river discharge:

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_N x_N \tag{21}$$

$$y = w_0 + w_1 {x_1}^{w_2} + w_3 {x_2}^{w_4} + \cdots + w_{2N-1} {x_N}^{w_{2N}} \tag{22}$$

where x is the input vector, y is the output, and w is the weight vector. In this study, the performances of PSCO, PSO, and other algorithms in optimizing Eq. (21) and Eq. (22) are investigated.

2.7 Integrative machine learning models

All ML models have some coefficients that are optimized during a training process based on available data. It is possible to use meta-heuristic algorithms to optimize the part or all of the model coefficients. In this study, multilayer perceptron neural network (MLPNN), adaptive neuro-fuzzy inference system (ANFIS), and regression equations (REs) are integrated with particle swarm optimization (PSO), autonomous groups particles swarm optimization (AGPSO), dwarf mongoose optimization algorithm (DMOA), weighted mean of vectors (INFO), and particle swarm clustered optimization (PSCO) algorithms. The meta-heuristic algorithms are applied to optimize the weights and biases of MLPNN, the Gaussian and linear membership functions of ANFIS, and the weights/coefficients of REs. The optimization process of each model is similar. At the first stage, a cost function must be introduced. In this study, the root-mean-square error (*RMSE*) between predicted and observed values is considered as a cost function. In the next stage, the meta-heuristic algorithm parameters are initialized. Table 1 shows the values of initial parameters. These values were selected based on recommendations in the literature and a trial-and-error procedure (Mirjalili et al. 2014; Zounemat-Kermani and Mahdavi-Meymand 2019; Babanezhad et al. 2021; Agushaka et al. 2022). In the final stage, an iteration procedure is applied to find the best solution for the problem based on the training and validation data sets.

Table 1 The initial values of PSCO and PSO

Method	Parameter	
PSO	Population	ANFIS and MLPNN: 1000, REs: 100
	Iteration	1000
	Search space range	[− 10, 10]
	Momentum factor (*w*)	0.99
	Personal learning coefficient (c_1)	1
	Global learning coefficient (c_2)	2
PSCO	Population	ANFIS and MLPNN: 1000, REs: 100
	Iteration	1000
	Clusters iteration (I_m)	300
	Number of clusters (*m*)	10
	Search space range	[− 10, 10]
	Momentum factor (*w*)	0.99
	Personal learning coefficient (c_1)	1
	Global learning coefficient (c_2)	2
AGPSO	Population	ANFIS and MLPNN: 1000, REs: 100
	Iteration	1000
	Search space range	[− 10, 10]
	Momentum factor (*w*)	0.99
INFO	Population	ANFIS and MLPNN: 1000, REs: 100
	Iteration	1000
	Search space range	[− 10, 10]
DMOA	Population	ANFIS and MLPNN: 1000, REs: 100
	Number of baby sister	10
	Iteration	1000
	Search space range	[− 10, 10]
	δ (Eq. 11)	2

Table 2 Details of selected benchmark functions

Category	Test function	Bounds	Dimension
Unimodal	$F_1(x) = \sum_{i=1}^{D} x_i{}^2$	[− 100,100]	D = 30, 50
	$F_2(x) = (x_1 - 1)^2 + \sum_{i=2}^{D} i\left(2x_i{}^2 - x_{i-1}\right)^2$	[− 10,10]	D = 30, 50
	$F_3(x) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$	[− 100,100]	D = 30, 50
Multimodal	$F_4(x) = \sum_{i=1}^{D-1}(w_i - 1)^2\left[1 + 10\sin^2(\pi w_i + 1)\right] + (w_D - 1)^2\left[1 + 10\sin^2(2\pi w_D +)\right]$	[− 10,10]	D = 30, 50
	$F_5(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i{}^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{i^{0.5}}\right) + 1$	[− 600,600]	D = 30, 50
	$F_6(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{D}(x_i - 1)^2\left[1 + \sin^2(3\pi x_1)\right] + (x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{D} g(x_i, 5, 100, 4)$	[− 50,50]	D = 30, 50
	$g(x_i, 5, 100, 4) = \begin{cases} b(x_i - a)^c & x_i > a \\ 0 & -a < x_i > a \\ b(-x_i - a)^c & x_i < -a \end{cases}$		
	$F_7(x) = \sum_{i=1}^{D} x_i \sin(x_i) + 0.1x_i$	[− 10,10]	D = 30, 50
	$F_8(x) = \sum_{i=1}^{D} 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i{}^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i{}^2}$	[− 100,100]	D = 30, 50
	$F_9(x) = \left(x_1{}^6 + x_2{}^4 - 17\right)^2 + (2x_1 + x_2 - 4)^2$	[− 500,500]	D = 2
	$F_{10}(x) = -\frac{1}{30}\exp\left(1 - \frac{\sqrt{x_1{}^2 + x_2{}^2}}{\pi}\right)\cos^2(x_1)\cos^2(x_2)$	[− 10,10]	D = 2

2.8 Implementation

2.8.1 Benchmark functions

In this study, particle swarm clustered optimization (PSCO) algorithm is derived and then applied to predict a river water discharge. Before the PSCO is applied in real problems, it is reasonable to evaluate its performance using mathematical benchmark functions. Ten benchmark functions are selected to compare the performance of PSCO with particle swarm optimization (PSO), autonomous groups particles swarm optimization (AGPSO), dwarf mongoose optimization algorithm (DMOA), and weighted mean of vectors (INFO). These functions belong to two main categories comprising unimodal and multimodal functions. The unimodal functions have an extremum. Hence, finding the solution of unimodal function is not a difficult task and the convergence rate of algorithms is more important. Multimodal functions, in addition to a global solution, have several local extrema points. Thus, the ability of algorithms to deviate from local solutions may be evaluated. Table 2 shows the selected benchmark functions, the search space, and the considered dimensions. To obtain solutions for the benchmark functions, the population, c_1, c_2, I_m, m, and the number of iterations are assumed to be 500, 1, 2, 300, 10, and 1000, respectively.

2.8.2 Study area and model development

The Vistula is the 9th-longest river in Europe and the longest river in Poland. The Vistula rises at Barania Góra in the Beskidy Mountains in the south of Poland and empties into the Gdansk Bay of the Baltic Sea. The Vistula has a significant effect on the European residents, environment, and economy, so the prediction of its discharge is essential. In this study, data recorded at the Torun station were used to simulate Vistula discharge. In Fig. 2, the location of the Torun station is shown. The available data comprise daily water temperature (T), water surface level (WSL), and water discharge (Q_w) from January 1984 to December 2017. The number of data is 12419, which creates an attractive data set for simulating and comparing the performance of different machine learning models. The T and WSL are assumed to be the model input parameters to predict Q_w. The whole data set is randomly divided into three sub-data sets including the training (70%), validation (15%), and testing (15%) data sets. The training data set is used for training the models, the validation data set is used to overcome overfitting, and the testing data set was used for the evaluation of models. Details of applied data sets are presented in Table 3.

2.8.3 Evaluation criteria

The root-mean-square error ($RMSE$), the coefficient of determination (R^2), the mean absolute error (MAE), the Nash–Sutcliffe model efficiency (NSE), and the index of agreement (IA) indices were used to compare the performance of developed machine learning models. These parameters are calculated from the following formulas:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Q_{w}{}_{i}^{m} - Q_{w}{}_{i}^{o}\right)^2} \tag{23}$$

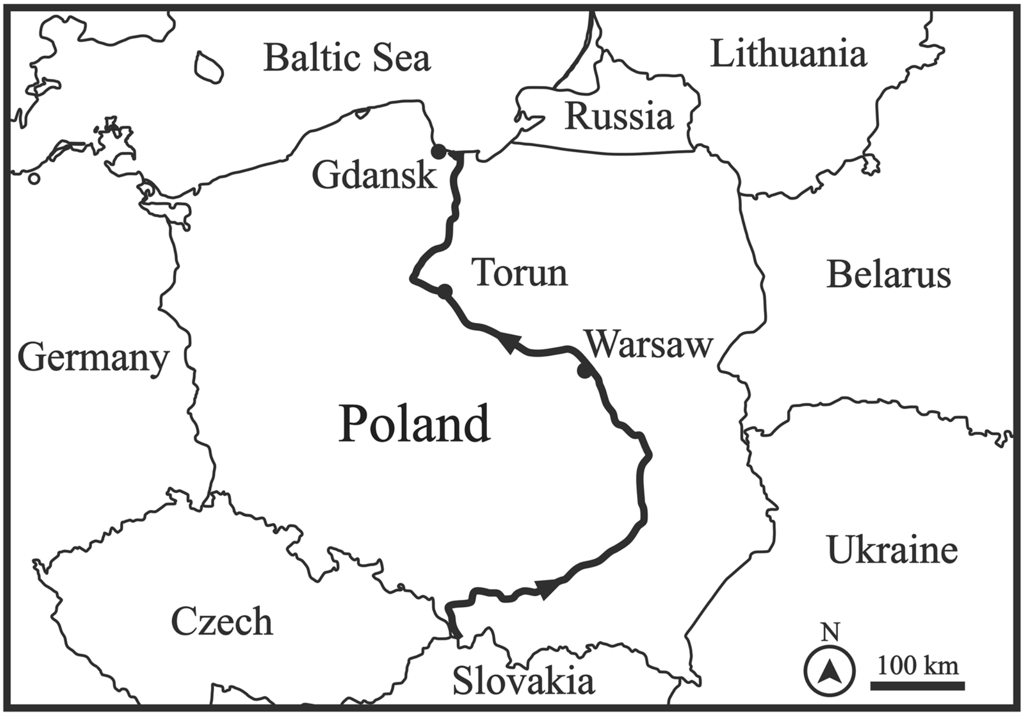

Fig. 2 The Vistula and the location of Torun station

Table 3 Statistical characteristics of used data sets

Parameter	Statistical parameter	Whole data set	Training data set	Validation data set	Testing data set
Q_w (m^3/s)	Max	6190	5850	5140	6190
	Min	218	218	228	229
	Avg	916.93	917.45	919.90	911.48
	SD	538.44	535.71	551.67	537.70
WLS (cm)	Max	836	816	771	836
	Min	94	94	94	94
	Avg	289.87	290.14	289.57	288.90
	SD	93.53	93.37	95.36	92.32
	R (vs. Q_w)	0.95	0.95	0.95	0.95
T (C)	Max	27	27	27	25.4
	Min	0	0	0	0
	Avg	10.58	10.62	10.51	10.48
	SD	7.79	7.79	7.81	7.77
	R (vs. Q_w)	− 0.14	− 0.14	− 0.11	− 0.15

Max, maximum; Min, minimum; Avg, average; SD, standard deviation; and R, correlation coefficient

$$R^2 = \frac{\left(\sum_{i=1}^{N}\left({Q_w}_i^m - \overline{{Q_w}^m}\right)\left({Q_w}_i^o - \overline{{Q_w}^o}\right)\right)^2}{\sum_{i=1}^{N}\left({Q_w}_i^m - \overline{{Q_w}^m}\right)^2\sum_{i=1}^{N}\left({Q_w}_i^o - \overline{{Q_w}^o}\right)^2} \tag{24}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{I=1}^{N}\left|{Q_w}_i^m - {Q_w}_i^o\right| \tag{25}$$

Table 4 Results obtained by the application of unimodal functions

Dimension	Result type	Result type	Algorithm				
			PSCO	**PSO**	**AGPSO**	**DMOA**	**INFO**
30	F_1	Best	1.15×10^{-150}	2.35×10^{-56}	1.39×10^{-18}	3.14×10^{-4}	6.45×10^{-57}
		Worst	5.69×10^{-140}	8.78×10^{-54}	2.11×10^{-14}	6.72×10^{-4}	4.19×10^{-56}
		Average	4.37×10^{-141}	1.90×10^{-54}	6.09×10^{-15}	4.78×10^{-4}	2.17×10^{-56}
		Rank	1	3	4	5	2
	F_2	Best	6.67×10^{-1}	6.67×10^{-1}	6.67×10^{-1}	5.9	6.67×10^{-1}
		Worst	6.67×10^{-1}	6.67×10^{-1}	6.67×10^{-1}	8.35	6.67×10^{-1}
		Average	6.67×10^{-1}	6.67×10^{-1}	6.67×10^{-1}	7.02	6.67×10^{-1}
		Rank	1	1	1	2	1
	F_3	Best	3.17×10^{-17}	4.22×10^{-7}	9.95×10^{-7}	$2.53 \times 10^{+4}$	2.54×10^{-55}
		Worst	2.33×10^{-13}	1.54×10^{-4}	1.41×10^{-6}	$4.45 \times 10^{+4}$	3.03×10^{-53}
		Average	2.91×10^{-14}	3.62×10^{-5}	3.27×10^{-7}	$3.48 \times 10^{+4}$	1.07×10^{-53}
		Rank	2	4	3	5	1
50	F_1	Best	1.03×10^{-76}	5.82×10^{-30}	4.16×10^{-18}	$8.67 \times 10^{+2}$	8.49×10^{-57}
		Worst	6.88×10^{-71}	1.5×10^{-27}	1.5×10^{-13}	$1.17 \times 10^{+3}$	2.09×10^{-55}
		Average	5.05×10^{-72}	4.63×10^{-28}	1.23×10^{-14}	$1.03 \times 10^{+3}$	6.72×10^{-56}
		Rank	1	3	4	5	2
	F_2	Best	6.67×10^{-1}	6.67×10^{-1}	6.67×10^{-1}	$2 \times 10^{+5}$	6.67×10^{-1}
		Worst	6.67×10^{-1}	6.67×10^{-1}	6.79×10^{-1}	$3.69 \times 10^{+5}$	6.67×10^{-1}
		Average	6.67×10^{-1}	6.67×10^{-1}	6.68×10^{-1}	$2.97 \times 10^{+5}$	6.67×10^{-1}
		Rank	1	1	1	2	1
	F_3	Best	2.29×10^{-3}	1.70	9.64×10^{-1}	$1.31 \times 10^{+5}$	1.2×10^{-53}
		Worst	6.37×10^{-1}	$7.20 \times 10^{+1}$	$5 \times 10^{+3}$	$2.11 \times 10^{+5}$	2.43×10^{-52}
		Average	7.14×10^{-2}	$2.30 \times 10^{+1}$	$3.38 \times 10^{+2}$	$1.76 \times 10^{+5}$	6.03×10^{-53}
		Rank	2	3	4	5	1
Final score			8	15	17	24	8
Final Rank			1	2	3	4	1

$$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{N}\left({Q_w}_i^m - {Q_w}_i^o\right)^2}{\sum_{i=1}^{N}\left({Q_w}_i^o - \overline{Q_w}^o\right)^2} \quad (26)$$

$$\mathrm{IA} = 1 - \frac{\sum_{i=1}^{N}\left({Q_w}_i^m - {Q_w}_i^o\right)^2}{\sum_{i=1}^{N}\left(\left|{Q_w}_i^m - \overline{Q_w}^o\right| + \left|{Q_w}_i^o - \overline{Q_w}^o\right|\right)^2} \quad (27)$$

where ${Q_w}_i^m$ and ${Q_w}_i^o$ are the simulated and measured water discharge and $\overline{Q_w}$ is the average value of water discharges.

3 Results

3.1 Benchmark functions

The performance of particle swarm clustered optimization (PSCO), particle swarm optimization (PSO), autonomous groups particles swarm optimization (AGPSO), dwarf mongoose optimization algorithm (DMOA), and weighted mean of vectors (INFO) is compared for unimodal and multimodal benchmark functions. The results obtained by the application of algorithms for unimodal functions and 30 consecutive runs are presented in Table 4.

Table 4 presents the rank of the applied algorithms for each function. The results show that the performance of PSCO is better than PSO, AGPSO, and DMOA. Both PSCO and INFO achieved the same excellent scores, which indicates that the performances of both methods are similar. However, the PSCO provides more accurate results than PSO and AGPSO. The DMOA is a newly developed algorithm that provides results of limited accuracy. The convergence rates for applied algorithms are presented in Fig. 3.

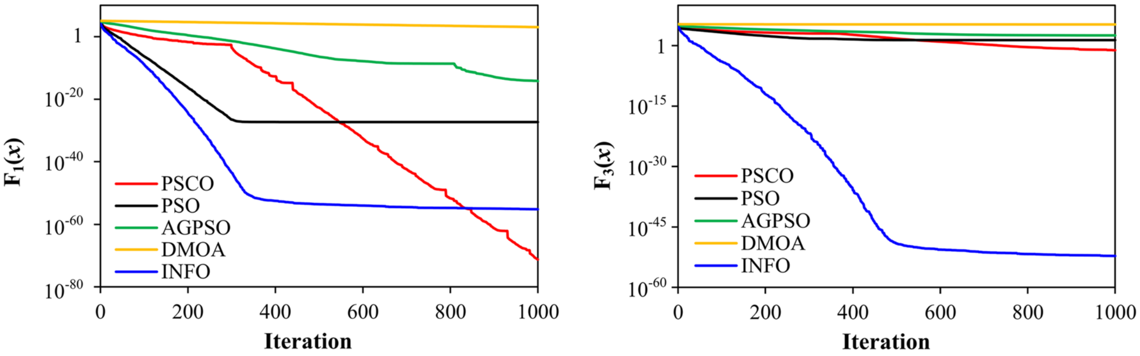

Fig. 3 The convergence rate for unimodal functions, $D=50$

The convergence rates presented in Fig. 3 indicate that PSCO is more efficient in finding a final solution than PSO, AGPSO, and DMOA. The convergence of PSCO is better than INFO for F_1, while the convergence of INFO is better than PSCO for F_3. It can be concluded that both of them are robust algorithms that can be recommended for unimodal functions. At the initial stage, the PSO convergence rate is better than the convergence rate of its counterpart as expected. This is because at the initial stage the PSCO population consists only of 10 clusters and each cluster is trying to find a solution based on cluster knowledge, while the PSO population is many times larger what implies a faster convergence. However, after the initial stage, the convergence rate of PSCO significantly improves. Multimodal functions, in addition to the global solution, have also several local solutions, so finding a final solution for these types of functions is more difficult. Table 5 presents the results obtained by the application of algorithms for multimodal functions and $D=30$, and $D=50$.

The results presented in Table 5 clearly show that there are significant differences between the best, the worst, and the average outcome of PSO. It means that PSO easily traps in local solutions. The results show that the approach applied in PSCO is successful and has no tendency to trap in local solutions. The PSCO obtained the lowest final score and the best rank among the tested algorithms. The performance of INFO for multimodal functions is not as good as for the unimodal functions. This shows the weakness of INFO and a tendency to trap in local solutions. The results also show that by increasing the dimension of a problem, the probability of trapping in local solution increases. This is because stochastic algorithms need more population for higher-dimensional problems. The results of PSCO, PSO, AGPSO, INFO, and DMOA for F_9 and F_{10} functions are the same. The dimensions of these two functions are low, so it is easy to find a solution. The convergence rate for multimodal functions and $D=50$ is presented in Fig. 4.

The plots in Fig. 4 indicate that the PSCO, contrary to PSO, has no tendency to trap in local solutions and finds a global solution with higher accuracy. The convergence rate of PSCO for multimodal functions is far better than for unimodal functions. At the initial stage, the convergence rate of PSO is better than PSCO as expected. After the initial stage, the convergence rate of PSCO rapidly increases. In practical problems, such as the prediction of river discharge, the accuracy of the final solution is more important than a convergence rate.

3.2 River discharge prediction

In this study, several machine learning models were used to predict Vistula water discharge. The applied models comprise multilayer perceptron neural network (MLPNN), adaptive neuro-fuzzy inference system (ANFIS) as well as ANFIS, MLPNN, linear equation (LE), and nonlinear equation (NE) integrated with particle swarm optimization (PSO), particle swarm clustered optimization (PSCO), autonomous groups particles swarm optimization (AGPSO), dwarf mongoose optimization algorithm (DMOA), and weighted mean of vectors (INFO). The average results obtained for 30 continuous runs in training and validations stages are presented in Table 6.

The results presented in Table 6 indicate that in the training and validation stages the performances of all applied models are acceptable. The average R^2 is higher than 0.90 which indicates, among others, that the applied models have the correct structure. The results show that the prediction of river discharge with high accuracy is possible with just two input parameters and randomized data sets (no time series). It is worth noting that most ML models provide different results in different runs. This may create a problem for users in real projects.

Table 5 Results obtained by the application of multimodal functions

Dimension	Result type	Result type	Algorithm				
			PSCO	**PSO**	**AGPSO**	**DMOA**	**INFO**
30	F_4	Best	1.5×10^{-32}	1.5×10^{-32}	1.5×10^{-32}	2.34×10^{-1}	8.95×10^{-2}
		Worst	6.39×10^{-30}	1.82	1.5×10^{-32}	6.97×10^{-1}	8.06×10^{-1}
		Average	1.17×10^{-30}	6.90×10^{-1}	2.16×10^{-32}	5.55×10^{-1}	3.88×10^{-1}
		Rank	2	5	1	5	4
	F_5	Best	0	0	0	1.14×10^{-2}	0
		Worst	1.09×10^{-14}	4.66×10^{-2}	1.72×10^{-2}	4.25×10^{-2}	0
		Average	1.99×10^{-15}	1.9×10^{-2}	4.27×10^{-3}	2.17×10^{-2}	0
		Rank	2	4	3	5	1
	F_6	Best	1.35×10^{-32}	1.35×10^{-32}	1.35×10^{-32}	$1.71 \times 10^{+1}$	1.35×10^{-32}
		Worst	4.69×10^{-31}	9.89×10^{-2}	5.99×10^{-31}	$2.43 \times 10^{+1}$	1.76×10^{-1}
		Average	4.83×10^{-32}	1.32×10^{-2}	5.94×10^{-32}	$2.13 \times 10^{+1}$	1.54×10^{-2}
		Rank	1	4	2	5	3
	F_7	Best	0	0	0	5.79×10^{-7}	4.94×10^{-8}
		Worst	0	1.11×10^{-16}	4.44×10^{-16}	2.75×10^{-4}	1.07×10^{-4}
		Average	0	1.11×10^{-17}	4.86×10^{-17}	7.52×10^{-5}	1.56×10^{-5}
		Rank	1	2	3	5	4
	F_8	Best	2×10^{-1}	2×10^{-1}	3×10^{-1}	6×10^{-1}	1.49×10^{-29}
		Worst	4×10^{-1}	5×10^{-1}	5×10^{-1}	8.95×10^{-1}	2.88×10^{-29}
		Average	3.33×10^{-1}	3.93×10^{-1}	3.39×10^{-1}	7.38×10^{-1}	2.15×10^{-29}
		Rank	2	3	3	4	1
50	F_4	Best	1.5×10^{-32}	4.28×10^{-18}	6.81×10^{-15}	$1.05 \times 10^{+2}$	6.27×10^{-1}
		Worst	5.94×10^{-29}	5.91	1.25×10^{-6}	$1.44 \times 10^{+2}$	1.79
		Average	8.57×10^{-30}	2.73	1.13×10^{-7}	$1.25 \times 10^{+2}$	1.02
		Rank	1	4	2	5	3
	F_5	Best	0	2.22×10^{-16}	0	8.62	0
		Worst	2.46×10^{-2}	4.41×10^{-2}	9.45×10^{-2}	$1.35 \times 10^{+1}$	0
		Average	4.92×10^{-3}	1.54×10^{-2}	1.43×10^{-2}	$1.1 \times 10^{+1}$	0
		Rank	2	4	3	5	1
	F_6	Best	1.97×10^{-32}	2.35×10^{-16}	4.03×10^{-13}	$1.51 \times 10^{+8}$	9.85×10^{-32}
		Worst	2.10×10^{-2}	2.13	1.1×10^{-2}	$2.85 \times 10^{+8}$	1.08×10^{-1}
		Average	2.87×10^{-3}	2.33	7.34×10^{-7}	$2.21 \times 10^{+8}$	1.67×10^{-2}
		Rank	2	4	1	5	3
	F_7	Best	0	0	0	9.46×10^{-6}	4.24×10^{-8}
		Worst	0	1.11×10^{-16}	0	4.93×10^{-4}	2.6×10^{-4}
		Average	0	7.86×10^{-18}	0	1.37×10^{-4}	4.4×10^{-5}
		Rank	1	2	1	4	3
	F_8	Best	5×10^{-1}	6×10^{-1}	5×10^{-1}	6.06	1.23×10^{-29}
		Worst	9×10^{-1}	1.20	8×10^{-1}	7.32	5.3×10^{-29}
		Average	6.33×10^{-1}	9.07×10^{-1}	7.13×10^{-1}	6.79	3.26×10^{-29}
		Rank	2	4	3	5	1
Final score			16	36	22	48	24
Final rank			1	4	2	5	3

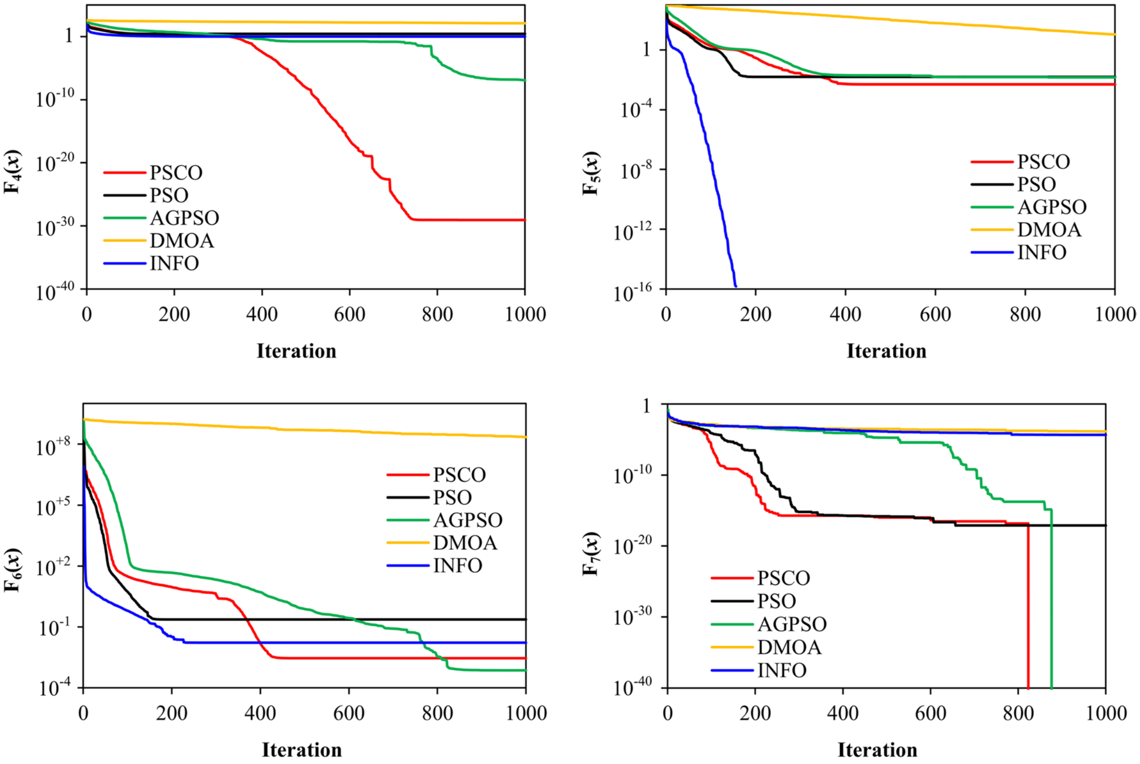

Fig. 4 The convergence rate for multimodal functions, $D = 50$

Table 6 Results obtained in training and validation phases

Method	Statistical index					
	Training			Validation		
	RMSE (m^3/s)	R^2	*NSE*	*RMSE* (m^3/s)	R^2	*NSE*
ANFIS	110.535	0.957	0.957	116.537	0.958	0.955
MLPNN	114.984	0.954	0.954	118.557	0.956	0.954
LE-PSCO	168.998	0.904	0.9	170.641	0.907	0.904
NE-PSCO	119.507	0.95	0.95	120.53	0.954	0.952
MLPNN-PSCO	116.915	0.952	0.952	115.478	0.959	0.956
ANFIS-PSCO	108.333	0.959	0.959	113.878	0.959	0.957
LE-PSO	169.778	0.904	0.899	171.508	0.907	0.903
NE-PSO	119.644	0.95	0.95	120.603	0.954	0.952
MLPNN-PSO	114.332	0.954	0.954	117.354	0.957	0.955
ANFIS-PSO	110.746	0.957	0.957	115.115	0.959	0.956
LE-AGPSO	168.022	0.904	0.902	169.287	0.908	0.906
NE-AGPSO	119.101	0.951	0.951	120.746	0.954	0.952
MLPNN-AGPSO	118.268	0.951	0.951	121.203	0.954	0.952
ANFIS-AGPSO	114.407	0.954	0.954	117.222	0.957	0.955
LE-INFO	167.65	0.904	0.902	169.769	0.908	0.905
NE-INFO	119.176	0.951	0.951	120.547	0.954	0.952
MLPNN-INFO	114.712	0.954	0.954	118.262	0.957	0.954
ANFIS-INFO	116.369	0.953	0.953	116.369	0.953	0.953
LE-DMOA	167.514	0.904	0.902	169.87	0.908	0.905
NE-DMOA	119.497	0.95	0.95	120.182	0.954	0.953
MLPNN-DMOA	168.948	0.917	0.901	190.611	0.918	0.881
ANFIS-DMOA	126.761	0.944	0.944	126.487	0.949	0.947

Table 7 Results obtained in testing phase

Method	**Statistical index**				
	RMSE **(m^3/s)**	***R^2***	***MAE*** **(m^3/s)**	***NSE***	***IA***
ANFIS	109.899	0.96	72.315	0.958	0.989
MLPNN	112.101	0.959	73.015	0.956	0.989
LE-PSCO	171.973	0.906	118.939	0.898	0.974
NE-PSCO	113.427	0.958	71.583	0.956	0.989
MLPNN-PSCO	109.963	0.96	72.724	0.958	0.989
ANFIS-PSCO	108.433	0.961	71.557	0.959	0.99
LE-PSO	172.53	0.905	119.245	0.897	0.974
NE-PSO	113.925	0.958	72.444	0.955	0.988
MLPNN-PSO	111.376	0.959	72.327	0.957	0.989
ANFIS-PSO	109.221	0.961	71.716	0.959	0.989
LE-AGPSO	172.824	0.906	120.234	0.897	0.973
NE-AGPSO	112.847	0.958	72.061	0.956	0.989
MLPNN-AGPSO	113.091	0.958	70.356	0.956	0.989
ANFIS-AGPSO	110.429	0.96	71.527	0.958	0.989
LE-INFO	172.198	0.905	119.756	0.897	0.974
NE-INFO	113.757	0.958	73.399	0.955	0.988
MLPNN-INFO	110.984	0.959	70.601	0.957	0.989
ANFIS-INFO	111.28	0.959	71.098	0.957	0.989
LE-DMOA	172	0.905	119.26	0.898	0.974
NE-DMOA	113.891	0.958	73.017	0.955	0.988
MLPNN-DMOA	159.048	0.929	86.81	0.913	0.974
ANFIS-DMOA	124.855	0.95	86.934	0.946	0.986

This is why the average from different consecutive runs is recommended to be used for the comparisons of the performances of different models. Table 6 shows that the nonlinear ML models are more accurate than linear ones in the training and validation stages. The ANFIS-PSCO, with the lowest *RMSE* and the highest R^2 and *NSE*, presents the best performance. In general, the results obtained in the training and validation stages show that the performance of the derived PSCO is better than PSO and its variant, AGPSO.

The results obtained in the training and validation stages provide important information regarding the performance of the derived models; however, more important for the evaluation of the models is the performance achieved in a testing stage. Table 7 presents the results obtained by the applied models in the testing stage.

The results presented in Table 7 indicate that in the testing stage the accuracies of applied ML models are acceptable ($R^2 > 0.90$). The performances of nonlinear models are better than their linear counterparts. On average, the nonlinear models are about 33.13% more accurate than linear models. The results show that the accuracies of all developed models are close to each other. However, ANFIS-PSCO, with the lowest *RMSE*, *MAE*, and the highest R^2, *NSE*, and *IA*, provides the best performance. The results show also that PSO, PSCO, AGPSO, and INFO increase the accuracy of ANFIS and MLPNN models. The DMOA decreases the ANFIS and MLPNN accuracies. The performance of DMOA applied for mathematical benchmark functions in high-dimensional problems was also low. The results also show that DMOA is an appropriate method for optimizing ML models with the low number of coefficients, as similar conclusion refers to LE and NE. Although the performances of PSCO and other algorithms are similar, the PSCO provides more accurate results in both high- and low-dimensional problems. The application of PSCO increases the accuracy of MLPNN and ANFIS by about 1.33% and 1.91%, respectively. Scatter plots are widely recognized visual techniques which may be applied for comparisons of the performances of different models. Figure 5 presents the average scatter plots for the testing stage.

The plots in Fig. 5 show that the applied models well predict the Vistula water discharge. The nonlinear models are more accurate than their linear counterparts. The points corresponding to linear methods are more scattered than points obtained by applying the nonlinear models. This means that the applications of advanced machine learning methods are attractive alternatives for the modeling of hydrological and hydraulic phenomena with high accuracy. The scatter plots of nonlinear models are located close to each other, which indicates that the performances of nonlinear models are similar. The ANFIS-PSCO provides the best performance among all developed models providing the best fitting of the predicted and observed results, the highest trendline slope coefficient ($m = 0.967$), and the lowest trendline intercept coefficient ($C = 54.431$).

4 Further discussion

In this study, particle swarm clustered optimization (PSCO) was developed to solve optimization problems. The performance of PSCO was compared with several recognized techniques available in the literature including particle swarm optimization (PSO), autonomous groups particles swarm optimization (AGPSO), dwarf mongoose optimization algorithm (DMOA), and weighted mean of vectors (INFO). In the first stage, the performances of the derived algorithms were evaluated by applying 10 complex mathematical benchmark functions. The applied benchmark functions describe applied scientific problems. The results indicate that the performance of algorithms depends on the type of problem. For some functions, the performances of the available algorithms are better than PSCO. However, in general, PSCO outperformed other algorithms. The PSCO developed in this study overcomes the weakness of PSO and a tendency to trap in local solutions. The DMOA results

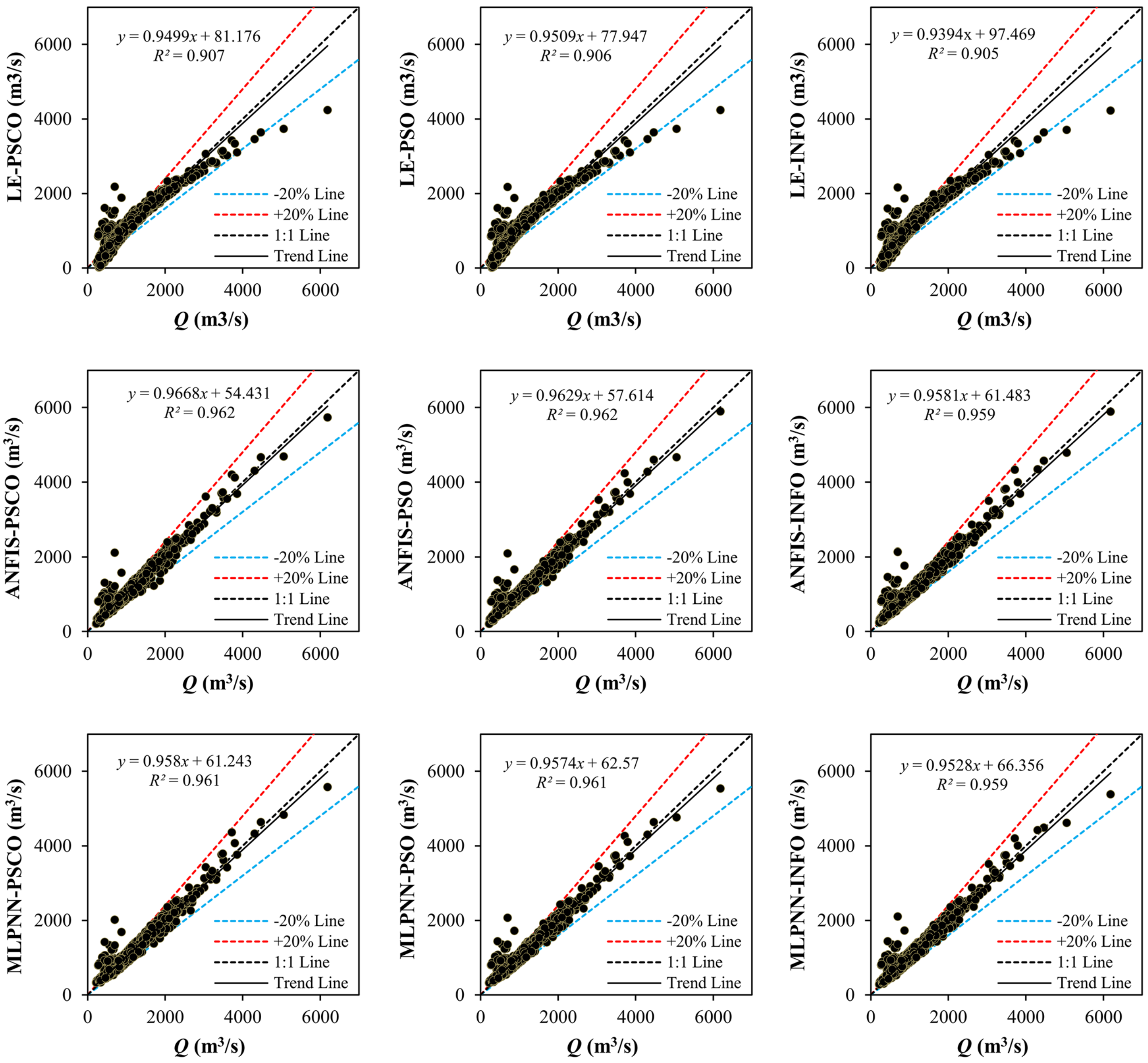

Fig. 5 The scatter plots for the testing stage

in high-dimensional problems are relatively poor. The application of DMOA may be recommended just for low-dimensional problems.

In the second stage, the performances of PSCO and other algorithms implemented to optimize machine learning (ML) models applied to predict Vistula river discharge were analyzed. As expected, the performance of DMOA applied in high-dimensional problems is far better than its performance in low-dimensional problems. The results of other algorithms are close to each other. However, the statistical indices indicate that ML models integrated with the developed PSCO are more accurate than other algorithms.

The inputs were selected based on data recorded by the Vistula river stations. This study indicates that the prediction of river discharge with high accuracy is possible with just two input parameters and randomized data sets (no time series). It was possible to use lags for both discharge and temperature and predict discharge by applying a time series modeling strategy (Lin et al. 2021; Zounemat-Kermani et al. 2021). The application of lags makes it possible to use more inputs. However, the predicted results for the second, third, or consecutive time steps are not accurate in such a modeling. In the literature, there are studies that report satisfying results obtained by applying two input parameters

(Heřmanovský et al. 2017; Hong et al. 2021; Song 2021). However, a more detailed analysis of the consequences of the application of more parameters may be recommended for future studies.

5 Conclusion

An original particle swarm clustered optimization (PSCO) method has been developed for the implementations in applied sciences. The performance of PSCO was compared with particle swarm optimization (PSO), autonomous groups particles swarm optimization (AGPSO), dwarf mongoose optimization algorithm (DMOA), and weighted mean of vectors (INFO). The derived PSCO, in contrary to particle swarm optimization technique (PSO) that is frequently used in many disciplines of applied sciences, does not trap in local solutions. The novel technique was applied as an integrative method with several machine learning (ML) models including multilayer perceptron neural network (MLPNN), adaptive neuro-fuzzy inference system (ANFIS), linear equation (LE), and nonlinear equation (NE) to predict river discharge. Ten benchmark functions were used to compare the performance of PSCO with frequently used traditional methods. The results show that PSCO provides the most accurate results. The PSCO provides the most accurate results for both high- and low-dimensional benchmark functions. The PSCO can escape from trapping in local solution and consistently and reliably reaches the global solution. The application of linear and nonlinear ML models in the prediction of river discharge shows that the nonlinear models act better. The performances of most algorithms are close to each other. However, the PSCO results are slightly more accurate than other algorithms in optimizing the MLPNN, ANFIS, and regression equations (REs). The average results of 30 continuous runs indicate that PSCO improves the performance of machine learning techniques and shows that the ANFIS-PSCO is the most accurate model.

Acknowledgements
Financial support for this study was partially provided by MuWin project,MarTERA4/1/9/MuWin/2023. The financial support is gratefully acknowledged.

Author contributions
A.M. and W.S. contributed to conceptualization and analysis and interpretation of results; A.M. was involved in methodology and writing—draft preparation; and W.S. contributed to writing—review and editing and supervision. Both authors read and approved the final manuscript.

Funding
Financial support for this study was partially provided by MuWin project,MarTERA4/1/9/MuWin/2023.

Availability of data and materials
Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interest.

Received: 24 October 2022 Accepted: 21 March 2023
Published online: 05 April 2023

References

Aalimahmoody N, Bedon C, Hasanzadeh-Inanlou N, Hasanzade-Inallu A, Nikoo M (2021) BAT algorithm-based ANN to predict the compressive strength of concrete—a comparative study. Infrastructures 6(6):80

Agushaka JO, Ezugwu AE, Abualigah L (2022) Dwarf mongoose optimization algorithm. Comput Methods Appl Mech Eng 391:114570

Ahmadianfar I, Heidari AA, Noshadian S, Chen H, Gandomi AH (2022) INFO: an efficient optimization algorithm based on weighted mean of vectors. Expert Syst Appl 195:116516

Arora A, Arabameri A, Pandey MM, Siddiqui MA, Shukla UK, Bui DT, Mishra VN, Bhardwaj A (2021) Optimization of state-of-the-art fuzzy-metaheuristic ANFIS-based machine learning models for flood susceptibility prediction mapping in the Middle Ganga Plain, India. Sci Total Environ 750:141565

Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput 23:715–734

Azad A, Farzin S, Kashi H, Sanikhani H, Karami H, Kisi O (2018) Prediction of river flow using hybrid neuro-fuzzy models. Arab J Geosci 11:718

Babanezhad M, Behroyan I, Taghvaie Nakhjiri A, Marjani A, Rezakazemi M, Heydarinasab A, Shirazian S (2021) Investigation on performance of particle swarm optimization (PSO) algorithm based fuzzy inference system (PSOFIS) in a combination of CFD modeling for prediction of fluid flow. Sci Rep 11:1505

Badrzadeh H, Sarukkalige R, Jayawardena AW (2013) Impact of multi-resolution analysis of artificial intelligence models inputs on multi-step ahead river flow forecasting. J Hydrol 507:75–85

Behzad M, Asghari K, Eazi M, Palhang M (2009) Generalization performance of support vector machines and neural networks in runoff modeling. Expert Syst Appl 36(4):7624–7629

Bomers A, Meulen BVD, Schielen RMJ, Hulscher SJMH (2019) Historic flood reconstruction with the use of an artificial neural network. Water Resour Res 55(11):9673–9688

Bonyadi MR, Li X, Michalewicz Z (2014) A hybrid particle swarm with a time-adaptive topology for constrained optimization. Swarm Evol Comput 18:22–37

Chen D, Ge Y, Wan Y, Deng Y, Chen Y, Zou F (2022) Poplar optimization algorithm: a new meta-heuristic optimization technique for numerical optimization and image segmentation. Expert Syst Appl 200:117118

Daliakopoulos IN, Tsanis IK (2016) Comparison of an artificial neural network and a conceptual rainfall–runoff model in the simulation of ephemeral streamflow. Hydrol Sci J 61(15):2763–2774

Dibike YB, Solomatine DP (2001) River flow forecasting using artificial neural networks. Phys Chem Earth Part B 26(1):1–7

Eberhart RC, Kennedy J (1995) Particle swarm optimization. In: Proceedings of the IEEE conference on neural network, IEEE, pp 1942–1948

Fadaee M, Mahdavi-Meymand A, Zounemat-Kermani M (2020) Suspended sediment prediction using integrative soft computing models: on the analogy between the butterfly optimization and genetic algorithms. Geocarto International

FajardoToro CH, Meire SG, Gálvez JF, Fdez-Riverola F (2013) A hybrid artificial intelligence model for river flow forecasting. Appl Soft Comput 13(8):3449–3458

Fattahi H, Hasanipanah M (2022) An integrated approach of ANFIS-grasshopper optimization algorithm to approximate flyrock distance in mine blasting. Eng Comput 38:2619–2631

Gauch M, Mai J, Lin J (2021) The proper care and feeding of CAMELS: how limited training data affects streamflow prediction. Environ Model Softw 135:104926

Hayyolalam V, Pourhaji Kazem AA (2020) Black Widow Optimization Algorithm: a novel meta-heuristic approach for solving engineering optimization problems. Eng Appl Artif Intell 87:103249

Heřmanovský M, Havlíček V, Hanel M, Pech P (2017) Regionalization of runoff models derived by genetic programming. J Hydrol 540:544–556

Hong J, Lee S, Lee G, Yang D, Bae JH, Kim J, Kim K, Lim KJ (2021) Comparison of machine learning algorithms for discharge prediction of multipurpose dam. Water 13(23):3369

Huang T, Mohan AS (2005) Significance of neighborhood topologies for the reconstruction of microwave images using particle swarm optimization. Asia-Pac Microw Conf Proc 2005:1–4

Jang JS (1993) Anfis: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23:665–685

Kennedy J (1999) Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406), vol 3, pp 1931–1938

Liang J.J, Suganthan P.N (2005) Dynamic multi-swarm particle swarm optimizer. In: Proceedings 2005 IEEE swarm intelligence symposium, SIS2005, pp 124–129

Lim WH, Isa NAM (2014) Particle swarm optimization with increasing topology connectivity. Eng Appl Artif Intell 27:80–102

Lin Y et al (2021) A hybrid deep learning algorithm and its application to streamflow prediction. J Hydrol 601:126636

Linh NTT, Ruigar H, Golian S, Bawoke GT, Gupta V, Rahman KU, Sankaran A, Pham QB (2021) Flood prediction based on climatic signals using wavelet neural network. Acta Geophys 69:1413–1426

Lv X, Wang Y, Deng J, Zhang G, Zhang L (2018) Improved particle swarm optimization algorithm based on last-eliminated principle and enhanced information sharing. Comput Intell Neurosci 2018:5025672

Memar S, Mahdavi-Meymand A, Sulisz W (2021) Prediction of seasonal maximum wave height for unevenly spaced time series by Black Widow Optimization algorithm. Mar Struct 78:103005

Mendes R, Kennedy J, Neves J (2003) Watch thy neighbor or how the swarm can learn from its environment. In: Proceedings of the 2003 IEEE swarm intelligence symposium, pp 88–94

Milan SG, Roozbahani A, Azar NA, Javadi S (2021) Development of adaptive neuro fuzzy inference system—evolutionary algorithms hybrid models (ANFIS-EA) for prediction of optimal groundwater exploitation. J Hydrol 598:126258

Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249

Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67

Mirjalili S, Lewis A, Safa Sadiq A (2014) Autonomous particles groups for particle swarm optimization. Arab J Sci Eng 39:4683–4697

Ong KM, Ong V, Sia CK (2021) A carnivorous plant algorithm for solving global optimization problems. Appl Soft Comput 98:106833

Orhan U, Hekim M, Ozer M (2011) EEG signals classification using the K-means clustering and a multilayer perceptron neural network model. Expert Syst Appl 38:13475–13481

Papacharalampous G, Tyralis H, Koutsoyiannis D (2019) Comparison of stochastic and machine learning methods for multi-step ahead forecasting of hydrological processes. Stoch Env Res Risk Assess 33:481–514

Rehman OU, Yang S, Khan S, Rehman SU (2019) A quantum particle swarm optimizer with enhanced strategy for global optimization of electromagnetic devices. IEEE Trans Magn 55:1–4

Shakti PC, Sawazaki K (2021) River discharge prediction for ungauged mountainous river basins during heavy rain events based on seismic noise data. Prog Earth Planet Sci 8:58

Solomatine DP, Dulal KN (2003) Model trees as an alternative to neural networks in rainfall—runoff modeling Arbres de modèles comme alternative aux réseaux de neurones en modélisation pluie—debit. Hydrol Sci J 48(3):399–411

Song B, Wang Z, Zou L (2021) An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve. Appl Soft Comput 100:106960

Song CM (2021) Data construction methodology for convolution neural network based daily runoff prediction and assessment of its applicability. J Hydrol 605:127324

Suryanto N, Ikuta C, Pramadihanto D (2017) Multi-group particle swarm optimization with random redistribution. In: International conference on knowledge creation and intelligent computing (KCIC), IEEE, p 17434080

Tsujimoto T, Shindo T, Kimura T, Jin'no K (2012) A relationship between network topology and search performance of PSO. In: 2012 IEEE congress on evolutionary computation, pp 1–6

Tu S, Rehman SU, Waqas M, Rehman OU, Yang Z, Ahmad B, Halim Z, Zhao W (2020) Optimisation-based training of evolutionary convolution neural network for visual classification applications. IET Comput 14(5):259–267

Van den Bergh F, Engelbrecht A.P (2002) A new locally convergent particle swarm optimizer. In: Proceedings of IEEE international conference on systems, man, and cybernetics 2002 (SMC 2002), pp 96–101

Yaseen ZM, Mohtar WHMW, Ameen AMS, Ebtehaj I, Razali SFM, Bonakdari H, Salih SQ, Al-Ansari N, Shahid S (2019) Implementation of univariate paradigm for streamflow simulation using hybrid data-driven model: case study in tropical region. IEEE Access 7:74471–74481

Zounemat-Kermani M, Mahdavi-Meymand A (2019) Hybrid meta-heuristics artificial intelligence models in simulating discharge passing the piano key weirs. J Hydrol 569:12–21

Zounemat-Kermani M, Mahdavi-Meymand A (2021) Embedded fuzzy-based models in hydraulic jump prediction. J Hydroinf 23(1):151–170

Zounemat-Kermani M, Mahdavi-Meymand A, Hinkelmann R (2021) A comprehensive survey on conventional and modern neural networks: application to river flow forecasting. Earth Sci Inf 14:893–911

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.