

# Predicting longitudinal dispersion coefficient using ANN with metaheuristic training algorithms

M. J. Alizadeh<sup>1</sup> · A. Shabani<sup>1</sup>  · M. R. Kavianpour<sup>1</sup>

Received: 10 March 2016/Revised: 9 February 2017/Accepted: 2 March 2017/Published online: 13 March 2017  
© Islamic Azad University (IAU) 2017

**Abstract** A reliable prediction of dispersion coefficient can provide valuable information for environmental scientists and river engineers as well. The main objective of this study is to apply intelligence techniques for predicting longitudinal dispersion coefficient in rivers. In this regard, artificial neural network (ANN) models were developed. Four different metaheuristic algorithms including genetic algorithm (GA), imperialist competitive algorithm (ICA), bee algorithm (BA) and cuckoo search (CS) algorithm were employed to train the ANN models. The results obtained through the optimization algorithms were compared with the Levenberg–Marquardt (LM) algorithm (conventional algorithm for training ANN). Overall, a relatively high correlation between measured and predicted values of dispersion coefficient was observed when the ANN models trained with the optimization algorithms. This study demonstrates that the metaheuristic algorithms can be successfully applied to make an improvement on the performance of the conventional ANN models. Also, the CS, ICA and BA algorithms remarkably outperform the GA and LM algorithms to train the ANN model. The results show superiority of the performance of the proposed model over the previous equations in terms of DR,  $R^2$  and RMSE.

**Keywords** Cuckoo search · Longitudinal dispersion · Neural networks · Rivers · Metaheuristic algorithms

## Introduction

Longitudinal dispersion coefficient ( $K$ ) is a key factor in pollution transport modeling in rivers. It usually happens when cross-sectional mixing is completed. However, accurate prediction of longitudinal dispersion coefficient is complicated due to nonuniformity of river bed and banks, irregularities of velocity, development of secondary flow, dead-zone, etc. Study of longitudinal dispersion coefficient has attracted the attention of many researchers. Therefore, a great number of research studies have been conducted to estimate longitudinal dispersion coefficient, based on different techniques (Azamathulla and Wu 2011; Disley et al. 2015; Elder 1959; Fischer et al. 1979; Kashefipour and Falconer 2002; Liu 1977; Najafzadeh and Tafarjoruz 2016; Sahay 2011; Seo and Cheong 1998; Tutmez and Yuceer 2013). Many different empirical equations have been developed based on the experimental and field measurements. In these expressions, channel width ( $W$ ), shear velocity, flow velocity and depth of water ( $H$ ) were mainly considered as key parameters affecting longitudinal dispersion coefficient. However, their predictions of longitudinal dispersion coefficient vary significantly (Etemad-Shahidi and Taghipour 2012). In addition, the extreme values of  $\frac{W}{H}$  and  $K$  were excluded in the model performance evaluation in order to enhance their accuracy. In other words, rivers with  $\frac{W}{H} > 50$  and  $K > 100$  were ignored (Etemad-Shahidi and Taghipour 2012; Sahay 2011). In such condition, the environmental issues and pollutant transportation are of great importance and, thus, accurate

**Electronic supplementary material** The online version of this article (doi:10.1007/s13762-017-1307-1) contains supplementary material, which is available to authorized users.

✉ A. Shabani  
a.shabani@mail.kntu.ac.ir

<sup>1</sup> Faculty of Civil Engineering, K.N. Toosi University of Technology, Tehran, Iran

prediction of longitudinal dispersion coefficient is essential.

Over the past two decades, artificial neural networks have been widely used for hydrological modeling and also for predicting longitudinal dispersion coefficient. Danandeh Mehr et al. (2015) applied different ANN algorithms for monthly streamflow forecasting. The results showed that the radial basis function (RBF)-based neural network provides better performance compared with the backpropagation feed forward (BPFF) neural network and the generalized regression neural network (GRNN). Jafarzadeh et al. (2016) conducted a comparative research study using ANN and support vector machine to predict soil cation exchange capacity. The results showed that the ANN model provides more accurate estimation than the support vector machine. Tayfur and Singh (2005) and Sahay (2011) developed ANN models for predicting longitudinal dispersion coefficient in natural streams. They have demonstrated that the ANN model can result in higher performance than the empirical regression-based predictive models. Different techniques can be applied to train ANN. Backpropagation (BP) technique is probably the most popular used ANN training technique due to its inherent simplicity and easy implementation. BP technique can apply different algorithms in training ANN (e.g., the gradient descent; the conjugate gradient; Bayesian; the Levenberg–Marquardt). These training algorithms may have one or more deficiencies such as convergence problem, sticking in local optimum and taking long time for training process. Moreover, in some cases their performance is not satisfactory (Asadnia et al. 2013; Cheng et al. 2015).

In recent years, evolutionary techniques such as the bee algorithm (BA), genetic algorithm (GA) and particle swarm optimization (PSO) algorithm have been successfully used for training ANN models. The results demonstrated that these techniques will improve the performance of the conventional neural networks (Chau 2004; Chen et al. 2015; Cheng et al. 2008; Dügenci et al. 2015; Kayarvizhy et al. 2013; Wang et al. 2012). Also, they overcome the above-mentioned drawbacks of different algorithms that can be applied in the BP training technique. Furthermore, some other optimization algorithms such as imperialist competitive algorithm (ICA) and cuckoo search (CS) algorithm have shown great efficiency for optimization problems in different fields of engineering (Gandomi et al. 2013; Kaveh and Talatahari 2010; Lucas et al. 2010).

To the best knowledge of the authors, this is the first effort which applies the metaheuristic algorithms to train ANN models for predicting longitudinal dispersion coefficient. Therefore, the main objective of the current research is to develop a more accurate model for predicting longitudinal dispersion coefficient in natural streams. In

this regard, ANN models with different training algorithms (GA, BA, ICA and CS algorithm) have been taken under investigation. The dimensionless ratios of width to depth ( $\frac{W}{H}$ ) and flow velocity to shear velocity ( $\frac{U}{U_*}$ ) were considered as input variables. These parameters are mainly applied in the predictive models for longitudinal dispersion coefficient because they are easily measurable, accessible and are consistent with the physics of the problem. Model performance is evaluated by using statistical error measures.

## Materials and methods

### Previous studies

Once the cross-sectional mixing is completed, the longitudinal dispersion becomes the most important process. Generally, two kinds of study can be recognized dealing with dispersion coefficient prediction. One method attempts to develop equations using different statistical approaches, and the other is based on constructing intelligence-based models to predict the coefficient. Each method contains some advantages. For example, the statistical methods that give equations are easy for computation of the coefficient. Anyhow, they are less accurate in comparison with those of the intelligence-based methods. A large number of studies have been carried out and presented some equations for predicting dispersion coefficient. Table 1 summarizes the results of some studies. In this table,  $C$  is cross-sectional average concentration ( $\text{kg}/\text{m}^3$ );  $U$  is average velocity (m/s) in the section;  $x$  is direction of the mean flow;  $t$  is time (s);  $K$  is longitudinal dispersion coefficient ( $\text{m}^2/\text{s}$ );  $H$  is the depth of flow,  $U_*$  is the bed shear velocity; and  $W$  is the channel width.

Regarding intelligence-based predictive models for longitudinal dispersion, several studies successfully applied different types of techniques for predicting  $K$  in natural streams. Noori et al. (2009) examined the ability of two models including support vector machine (SVM) and adaptive neuro-fuzzy inference system (ANFIS). They evaluated the performance of the models in terms of correlation coefficient ( $R$ ) and provided the  $R$  values of 0.7 and 0.71 for the SVM and ANFIS models, respectively. Sahay (2011) developed a backpropagation ANN model to forecast the dispersion coefficient. The results showed that the proposed model outperforms the practical equations. Also, it was found that the ANN model gives an accuracy (DR) of 65% for predicting dispersion coefficient with  $K < 100 \text{ m}^2/\text{s}$  and  $\frac{W}{H} < 50$ . Azamathulla and Wu (2011) developed a support vector machine approach for predicting dispersion coefficient in rivers. They concluded that the



**Table 1** Empirical equations for computation of dispersion coefficient

Nos.	References	Equations
1	Taylor (1953, 1954)	$\left(\frac{\partial C}{\partial t}\right) + U\left(\frac{\partial C}{\partial x}\right) = K\left(\frac{\partial^2 C}{\partial x^2}\right)$
2	Elder (1959)	$K = 5.93HU_*$
3	Fischer et al. (1979)	$\frac{K}{HU_*} = 0.011\left(\frac{U}{U_*}\right)^2\left(\frac{W}{H}\right)^2$
4	Seo and Cheong (1998)	$\frac{K}{HU_*} = 5.915\left(\frac{U}{U_*}\right)^{1.428}\left(\frac{W}{H}\right)^{0.62}$
5	Kashefipour and Falconer (2002)	$\frac{K}{U_*H} = \left[7.428 + 1.775\left(\frac{W}{H}\right)^{0.62}\left(\frac{U}{U_*}\right)^{0.572}\right]\left(\frac{U}{U_*}\right)^2$ ; if $\frac{W}{H} > 50$
6	Kashefipour and Falconer (2002)	$\frac{K}{U_*H} = 10.612\left(\frac{U}{U_*}\right)^2$ ; if $\frac{W}{H} > 50$
7	Sahay and Dutta (2009)	$\frac{K}{HU_*} = 2\left(\frac{U}{U_*}\right)^{1.25}\left(\frac{W}{H}\right)^{0.96}$
8	Etemad-Shahidi and Taghipour (2012)	$\frac{K}{HU_*} = 15.49\left(\frac{W}{H}\right)^{0.78}\left(\frac{U}{U_*}\right)^{0.11}$ ; if $\frac{W}{H} < 30.6$
9	Etemad-Shahidi and Taghipour (2012)	$\frac{K}{HU_*} = 8.36\left(\frac{W}{H}\right)^{0.61}\left(\frac{U}{U_*}\right)^{0.85}$ ; if $\frac{W}{H} > 30.6$
10	Li et al. (2013)	$\frac{K}{U_*H} = 2.282\left(\frac{W}{H}\right)^{0.7613}\left(\frac{U}{U_*}\right)^{1.4713}$
11	Zeng and Huai (2014)	$\frac{K}{HU_*} = 5.4\left(\frac{W}{H}\right)^{0.7}\left(\frac{U}{U_*}\right)^{1.13}$

proposed approach can be successfully used in order to provide an acceptable estimation of the dispersion coefficient. Noori et al. (2015) investigated the reliability of the intelligence-based models including ANN, ANFIS and SVM for predicting dispersion coefficient. The results indicated that a high uncertainty can be found in the models although they predicted the coefficient appropriately.

**Data and statistical analysis**

In this study, a comprehensive field dataset was used to develop the models. The data that include longitudinal dispersion coefficient, flow depth, flow velocity, shear velocity and channel width were obtained from the technical literature (Etemad-Shahidi and Taghipour 2012). These data have been widely employed in previous study to predict longitudinal dispersion due to their availability for a large number of rivers. Moreover, it was found that flow depth, flow velocity, channel width and shear velocity are the most effective parameters for predicting dispersion coefficient (Noori et al. 2015). Generally, 149 distinctive data records including different geometric and hydraulic parameters were used in model development (Etemad-Shahidi and Taghipour 2012). The statistical analysis of the data applied for this study is given in Table 2. These statistical measures of the datasets include minimum (Min), maximum (Max), average (Mean), standard deviation (SD) and skewness coefficient (C<sub>sk</sub>).

**Table 2** Statistical analysis of the data

Parameter	Min.	Max.	Mean	SD	C <sub>sk</sub>
W (m)	1.4	711.2	59.5	90.07	4.72
H (m)	0.14	19.94	1.53	2.1	5.34
U (m/s)	0.029	1.73	0.476	0.317	1.51
U <sub>*</sub> (m/s)	0.0016	0.553	0.084	0.0717	3.94
K (m <sup>2</sup> /s)	0.2	891.87	79.42	144.5	3.41
W/L	2.2	403.75	47.43	47	4.44
U/U <sub>*</sub>	0.77	20.25	7.04	4.67	1.19
K/U <sub>*</sub> H	3.08	5630.68	770.9	1028.4	2.83

All the data were normalized in a range of [-1, 1] applying Eq. (1) as (Xiao et al. 2014):

$$x_{\text{norm}} = \left(\frac{x_i - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \times 2\right) - 1 \tag{1}$$

Two main advantages of data normalization before applying ANN models are to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges and to avoid numerical difficulties during the calculation (Wang et al. 2009).

**Artificial neural networks (ANN)**

Over two past decades, ANNs as a ‘black box’ approach showed great ability dealing with nonlinear problems. The

ANNs that consist of an interconnected assembly of simple processing nodes are capable to find an appropriate relationship between input and target variables. A usual ANN is formed from three distinguished layers including input, output and hidden layer. Neurons of each layer are connected with neurons in the next layer. Strength of the connections is determined by a term called weight. Generally, mathematical structure of an ANN can be defined as:

$$y = \varphi \left( \sum_{i=1}^n w_i x_i - b \right) \quad (2)$$

In Eq. (2),  $w_i$  shows the weight of the input variable  $x_i$ ,  $b$  represents the bias,  $\varphi$  denotes the activation function, and  $n$  stands for number of input variables (Bishop 1995). Sigmoid function and a linear activation function were used as transfer functions in the hidden layer and output layer, respectively. These activation functions can be formulated as Eqs. (3) and (4).

$$f(y_j) = 1 / \left( 1 + \exp \left( - \left( \sum_{i=1}^n w_{ij} x_i - b_j \right) \right) \right), \quad (3)$$

$$j = 1, 2, \dots, h$$

$$y_k = \sum_{j=1}^h w_{jk} f(y_j) - b \quad (4)$$

where  $y_j = \sum_{i=1}^n w_{ij} x_i - b_j$ ,  $n$  is the number of input nodes or variables,  $b_j$  is the bias for the  $j$ th node in the hidden layer, and  $b$  is the bias for the output layer.  $h$ ,  $w_{ij}$ ,  $w_j$  and  $y_k$  stand for the number of neurons in the hidden layer, the connection weight from the  $i$ th node in the input layer to the  $j$ th node in the hidden layer, the connection weight from the  $j$ th node in the hidden layer to the output node and the final output, respectively.

Training ANN models play an important role on the ANN performances. The training procedure is viewed as a process to find the optimum weights and biases. In other words, it tunes the ANN model in a way to minimize the difference between observed and predicted target variable. Different algorithms such as gradient descent, backpropagation and heuristic/metaheuristic can be employed in the training phase. In the present study, the Levenberg–Marquardt backpropagation algorithm and also some metaheuristic algorithms including GA, ICA, BA and CS are employed to train the ANN models. The Levenberg–Marquardt BP algorithm, which is a simplified version of Newton method, has been widely used because it is usually faster and more reliable than any other backpropagation techniques (Hagan and Menhaj 1994; Ham and Kostanic 2000). The metaheuristic algorithms will be described in the following

subsections. More information related to ANNs can be found in Hagan and Menhaj (1994) and Ham and Kostanic (2000).

## Optimization algorithms

### Genetic algorithm (GA)

GA is a heuristic/metaheuristic, stochastic, combinatorial, optimization technique that mimics the process of natural selection. Main components of GA include bit, gene, chromosome and gene pool. Gene illustrating a decision variable consists of bits (0 and 1) which stands for a model parameter to be optimized. The combination of genes forms the chromosome, and the gene pool is comprised of a set of chromosomes (Tayfur 2009). Each chromosome can be considered as a possible solution for each decision variable. Reproduction, crossover and mutation are three of the most important operations of the algorithm. Reproduction contains elite and selection of chromosomes. Characteristics of the elite which is defined as a chromosome with the best fitness do not change in the next generation. During the selection process, some of the chromosomes which are not elite but have a better fitness than others are chosen as tentative genes to be utilized in the crossover stage. Crossover is a process in which chromosomes of the older generation are combined with each other to make a new generation of chromosomes. In this stage, parents exchange their genes and create new members. In mutation, gene values in a chromosome can be changed in which it can be resulted to omit a gene from chromosome or to add a new gene to the chromosome. Therefore, the present solution can be different from the previous solution and subsequently the solution is changed in a way to obtain a better solution through mutation process.

The parameters of an optimization problem are first encoded as an artificial chromosome (Chen and Chang 2009). The algorithm starts the optimization process with a number of randomly initialized chromosomes. Afterward, a cost function (objective function) is defined in order to evaluate the fitness of each chromosome. The reproduction, crossover and mutation continue to find the optimal solution up to maximum iteration number or until the stop criteria are satisfied. One can find more information related to the algorithm in Goldberg (1989). In this study, we investigated different values of population size, mutation probability and crossover probability to achieve the optimum values through trial-and-error procedure.

### Imperialist competitive algorithm (ICA)

The ICA is a new socio-politically motivated global search strategy that is used to solve different types of optimization



problems (Atashpaz-Gargari and Lucas 2007). Dealing with the ICA, a large number of parameters should be set appropriately in order to achieve optimized solutions. Some of these parameters include the number of countries, number of imperialists and colonies, number of decades, revolution rate, assimilation coefficient and assimilation angle coefficient. The algorithm is initiated with the initial world which has a primary population. Individuals form countries. Countries with the best objective functions (powerful countries) constitute imperialists, and the remains form colonies of the imperialists. Based on the power of the imperialists, they absorb the colonies in which the imperialist with more power tries to have more colonies. The position of the imperialist and a colony is changeable if the imperialist becomes less powerful than a colony. Afterward, the ICA updates with the new position of the imperialists and colonies.

Empires compete with each other in order to get the control of colonies belonging to the others (not themselves). Subsequently, some empires will be more powerful by getting the control of more colonies from the weaker empires. The empires that lost their power may collapse. There is a tendency among colonies to remain in their imperialists through competition and collapse processes which led them to form a state (Ardalan et al. 2015). The new situation provides an equal position and power for colonies and the imperialist. Dealing with ICA, there are a number of parameters (e.g., number of initial countries, number of initial imperialists, assimilation coefficient and revolution rate) which should be tuned appropriately in order to get optimum solutions.

#### *Bee algorithm (BA)*

The bee algorithm is a global search algorithm which inspired by foraging strategy of honey bees. In nature, bees conduct a waggle dance including some useful information such as the distance from the hive and the quality of food source by dance duration and frequency, respectively. In this way, they help the colony to send its bees precisely. The BA introduced by Pham et al. (2005), as a metaheuristic algorithm, is organized to achieve the optimum solution. Some important parameters affecting the efficiency of the algorithm are:  $n$  (number of scout bees),  $m$  (number of sites selected out of  $n$  visited sites),  $e$  (number of best sites out of  $m$  selected sites),  $ngh$  (initial size of patches),  $nep$  (number of bees recruited for best  $e$  sites),  $nsp$  (number of bees recruited for the other). Bee algorithm begins with  $n$  scout bees being randomly placed in the search space. After return, suitability of the visited places via scout bees is assessed and stored as arrays in descending order. In other words, it initializes the population with random solutions and evaluates fitness of the

population. The  $m$  best sites having the best fitness are selected randomly, and then, the best  $e$  sites out of  $m$  will be chosen. Using the information derived from  $m$  bees in the preceding step, the  $ngh$  is determined or updated. In the recruitment procedure, number of  $nep$  and  $nsp$  ( $nsp \leq nep$ ) are sent to  $e$  sites and  $m-e$  sites, respectively. In other words, more foragers are assigned for places with better solution recruited. Therefore, patches can be visited by more bees or may be abandoned considering their fitness. Afterward, the best bee from each site (the highest fitness) is selected to form the next bee population. It continues by forming new population and repeating the above-mentioned procedures for a specified iteration number or until the stop criteria be obtained. As it was mentioned, these values were obtained through trial-and-error procedure.

#### *Cuckoo search algorithm*

Cuckoo search (CS) is a recent metaheuristic algorithm inspired by the breeding behavior of some species of cuckoo birds. These species have an interesting behavior which lay their eggs in the nests of other species. If the host birds detect the cuckoo egg, they may throw it away or leave their nests and construct a new nest in other places. The algorithm initializes with a population of cuckoos. It acts in a way that each egg and cuckoo egg stand for a solution and a new solution, respectively. Cuckoo eggs are considered as an improved solution that are going to be substituted with those of worse solutions in the nest. It should be pointed out that in the algorithm implementation, one egg is placed in each nest. A usual cuckoo search algorithm composed of three rules including (Yang and Deb 2009): (1) At a time, each cuckoo lays one egg only and dumps it in a randomly chosen nest; (2) the best nests with the best solutions are remained to be carried over to the next generations; (3) the probability ( $p_a$ ) that the host birds can detect the cuckoo egg in their nests varies from  $[0, 1]$ ; also it is assumed that number of their nests are unchanged. The nest is substituted with a new nest containing random solutions when the host bird detects the cuckoo egg. At each iteration, the best solution (nests with the highest fitness) is memorized and the process will be repeated to reach the requirements. Similar to other optimization metaheuristic algorithms, cuckoo search comprises of two local and global search spaces which  $p_a$  represents its local search capability and  $1 - p_a$  denotes its global search strength. The parameters of the algorithm have been tried for different values, and the best values have been found. It was found that  $P_a = 0.25$  resulted in best performance which is in a complete accordance with that proposed by Yang and Deb (2009). Details of the CS algorithm can be found in Gandomi et al. (2013), Yang and Deb (2009) and Yildiz (2013).

### ANN training with optimization algorithms

Once a network has been structured for a particular application, training process can be started. In this stage, optimization algorithms are applied to tune weights and biases accordingly. It was found that 10 neurons in the hidden layer for all the ANN models applied in this study give a higher performance. The metaheuristic algorithms can be employed for training ANNs in two ways. They can be applied to minimize error by finding optimum values of weights and biases and obtain a suitable ANN structure. In this research, the metaheuristic algorithms have been applied to train ANN models by finding optimum weights and biases. In other words, role of training algorithm is to minimize the error which is defined as difference between real and estimated values of the target values. To do this, the training algorithm through an iterative process for the applied algorithms assigns different values to weights and biases and measures the error. In each algorithm (Metaheuristic and LM), the fitness function is evaluated for 500,000 times at most.

Through continuous iterations, the network is set to have minimum error. For each iteration, the algorithm tunes the weights and biases and computes the error in a way to improve the model performance or decrease the error for next iteration. Assuming  $m$  input–output sets,  $y_k - t_k$  for  $k = 1, 2, \dots, m$ ;  $y_k$  and  $t_k$  represent forecasted and real values of  $k$ th output unit. Therefore, the error is defined as follows:

$$E = \sqrt{\frac{\sum_{k=1}^m E_k}{m}} \quad \text{where } E_k = (t_k - y_k)^2 \quad (5)$$

The optimization algorithms are aimed to minimize the error which is considered as cost function. The minimum of cost function is achieved if the nodes in the layers connect to each other with optimized weights and biases. The weights and biases which should be optimized are  $\theta = (iw.hw.hb.ob)$  which are arranged as a single vector of real numbers.  $iw$  and  $hw$  represent weights for input to hidden layer connections and hidden to output layer connections, respectively. Also,  $hb$  and  $ob$  are the biases imposed to the neurons in the hidden and output layers. The weights and biases of ANN for every solution in evolutionary algorithms can be encoded in three methods including matrix, vector and binary (Zhang et al. 2007). In the current research, the vector encoding method has been taken under consideration. However, it consists of a decoding phase in which particles' vectors are applied to form weights and biases matrix. In this study, the weights and biases matrix are formed as follows:

$$iw = \begin{bmatrix} wa1 & wb1 \\ wa2 & wb2 \\ wa3 & wb3 \\ wa4 & wb4 \\ wa5 & wb5 \\ wa6 & wb6 \\ wa7 & wb7 \\ wa8 & wb8 \\ wa9 & wb9 \\ wa10 & wb10 \end{bmatrix} \cdot hb = \begin{bmatrix} b1 \\ b2 \\ b3 \\ b4 \\ b5 \\ b6 \\ b7 \\ b8 \\ b9 \\ b10 \end{bmatrix} \cdot hw = \begin{bmatrix} w1c \\ w2c \\ w3c \\ w4c \\ w5c \\ w6c \\ w7c \\ w8c \\ w9c \\ w10c \end{bmatrix} \cdot ob = [b11]$$

In brief, 41 variables including weights and biases have been resulted following the ANN model for this study. As we have two input variable, therefore, we have two input nodes in the ANN structure. Also, through a trial-and-error procedure, it was found that 10 neurons in the hidden layer give the optimum performance. Figure 1 shows a schematic layout of the ANN model with a 2–10–1 structure in

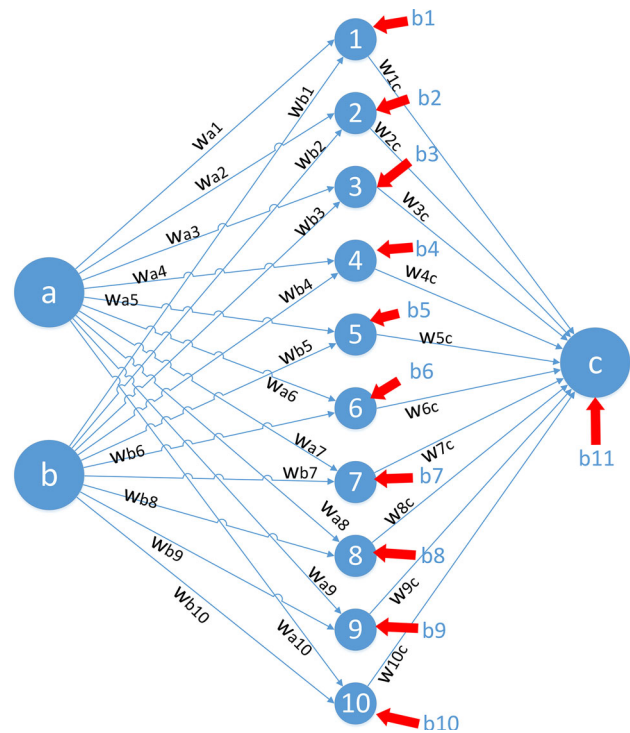


Fig. 1 A schematic layout of the ANN with a 2–10–1 structure

which 2, 10 and 1 represent number of neurons in the input layer, hidden layer and output layer.

To have a fair comparison between the performance of the different algorithms in training the ANN models, the models’ parameters were tuned in a way the number of fitness evaluation and also number of recall to change the weights and biases for all the algorithms be equal. In this regard, as the number of parameters is different and for high dimensionality of the problem, the 500,000 has been selected as the number of neural network (fitness evaluation) modifications (e.g., for the cuckoo search:  $5000 \times 50 \times 2 = 500,000$ , where 5000, 50 and 2 stand for number of iterations, number of nests and fitness evaluation for each CS algorithm iteration, respectively).

Through a trial-and-error procedure and following the equal number of fitness evaluation for each model, the following parameters have been achieved for the meta-heuristic training algorithms (Table 3).

In order to increase the speed of the MATLAB software for such large number of parameters, the computer codes were originally developed in C language and recalled with MATLAB software.

In the developed models, two variables of  $\frac{W}{L}$  and  $\frac{U}{U_s}$  were applied to predict  $\frac{K}{U_s H}$ . In this regard, different optimization algorithms including GA, ICA, BA and CS algorithms as well as Levenberg–Marquardt algorithm were applied for training the ANN. In all the developed models, 80 and 20% of the data were used for training and testing ANN, respectively. In the data selection, it has been tried to select test data in a way that follow the normal distribution of the training data. It is noteworthy, for the ANN model trained with Levenberg–Marquardt algorithm, we applied a validation dataset including 20% of data. Therefore, for the LM-ANN model, the dataset was divided into three subsets of training data (60%), validation data (20%) and testing data (20%), whereas for the other ANN models, only two sets of data including training (80%) and testing (20%) datasets. The validation dataset in gradient-based algorithm for training ANN models is applied in order to overcome network overfitting problems. Figure 2 shows the data in a 3D graph in which blue star and red circles represent training and testing data, respectively.

**Table 3** Summary of the algorithm parameters

GA		ICA		BA		CS	
Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
Population size	200	Initial countries	250	$n$	100	$P_a$	0.25
Mutation probability	0.05	Initial imperialists	20	$m$	40	Nests	50
Crossover probability	0.7	Assimilation coefficient	1.5	$e$	10		
		Revolution rate	0.3	$ngh$	0.1		
				$nep$	30		
				$nsp$	10		

### Performance evaluation criteria

In this study, the performance of different predictive models is evaluated by using statistical criteria including the coefficient of determination ( $R^2$ ), the root-mean-square error (RMSE) and the discrepancy ratio (DR).  $R^2$  measures the degree of determination among the actual and predicted values. RMSE indicates the discrepancy between the observed and forecasted values. Positive and negative values of discrepancy ratio (DR) indicate an overestimation and an underestimation, respectively. The percentage of DR values which are in a range between  $-0.3$  and  $0.3$  is defined as accuracy (Kashefipour and Falconer 2002; Seo and Cheong 1998). In brief, the models’ predictions are optimum if  $R^2$ , RMSE and DR are found to be close to 1, 0 and 0, respectively. These indices ( $R^2$ , RMSE and DR) are defined as follows:

$$R^2 = \frac{[\sum_{i=1}^n (K_{i(m)} - \bar{K}_{(m)})(K_{i(p)} - \bar{k}_{(p)})]^2}{\sum_{i=1}^n (K_{i(m)} - \bar{K}_{(m)})^2 \sum_{i=1}^n (K_{i(p)} - \bar{k}_{(p)})^2} \tag{6}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (K_{i(m)} - K_{i(p)})^2}{n}} \tag{7}$$

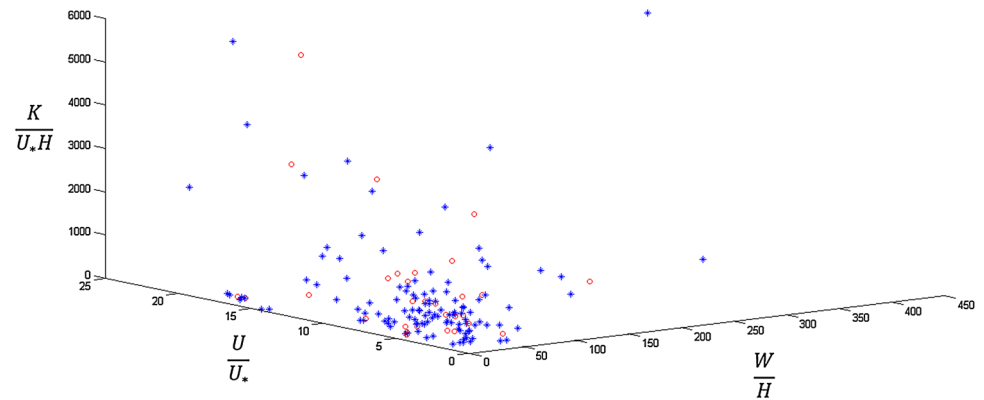
$$DR = \log \frac{K_{(p)}}{K_{(m)}} \tag{8}$$

where  $K$  and  $n$  stand for dimensionless longitudinal dispersion ( $\frac{K}{U_s H}$ ) and the number of data, respectively. In Eq. (8),  $\bar{k}_{(m)}$  and  $\bar{k}_{(p)}$ , respectively, stand for the mean values of measured and predicted dimensionless longitudinal dispersion coefficient.

### Results and discussion

Relative ratios of  $\frac{W}{H}$  and  $\frac{U}{U_s}$  are imposed as input variables to predict the dimensionless values of  $\frac{K}{U_s H}$ . Four different metaheuristic optimization algorithms including GA, ICA, BA and CS and the Levenberg–Marquardt (LM) algorithm were applied to train ANN models. Results of the best

**Fig. 2** The training and testing data



ANN models for each algorithm and during testing period are presented in Table 4. The table shows that applying optimization algorithms to train neural network remarkably improves the ANN performance. All the optimization algorithms are superior over the traditional LM algorithm for training ANN. The performance evaluation criteria (DR,  $R^2$  and RMSE) improve when metaheuristic algorithms are used instead of the LM algorithm for training process. Generally, the performance of the ANN models trained with the metaheuristic algorithms is close to each other regarding DR and RMSE. The best model in terms of accuracy is the ICA-ANN model, while the CS-ANN model has the highest performance in terms of  $R^2$  and RMSE and the GA shows the lowest efficiency. Anyway, the GA-ANN model outperforms the LM-ANN model considering the accuracy,  $R^2$  and RMSE.

It is concluded that the proposed models can be applied successfully to estimate longitudinal dispersion coefficient in natural streams. High values of  $R^2$  and the accuracy ( $-0.3 < DR < 0.3$ ) and also small values of RMSE for testing period of the developed models demonstrate their ability. The models including CS, BA and ICA have relatively low values of RMSE. As the values of dispersion coefficient applied for this study change in a wide variety, great values of RMSE reveal the failure of the model to predict the extreme values of the target parameter. Regarding the performance evaluation criteria, the CS, BA, ICA and GA are the most efficient algorithms for training

the ANN models for this study. Figures 3 and 4 show the predicted values of  $\frac{K}{U_*H}$  versus observed values for testing set which illustrate the ability of the models.

The figures show that the proposed models are able to give an acceptable prediction of the target variable and provide a great performance in estimation of extreme values. Except LM and ICA algorithms with a deficiency in prediction of extreme values, other algorithms offer better prediction. Therefore, BA, GA and cuckoo search algorithms are recognized as efficient algorithms for training ANN models to give an acceptable prediction of longitudinal dispersion coefficient. Generally, BA-based ANN model overestimates the  $\frac{K}{U_*H}$  values, while GA underestimates the target values. Regarding the scatter diagram, the highest correlation between observed and predicted values is derived for cuckoo and BA algorithms. ANN with cuckoo search algorithm gains  $R^2 = 0.693$  during testing period, which demonstrate a relatively high correlation between measured and predicted values. The CS-ANN model reasonably predicts the extreme values. Therefore, providing an acceptable prediction for high values of  $K$  is one of the advantages of the ANN models. Moreover, the correlation coefficient derived for the ANN models trained with BA and CS algorithms is relatively high. According to Fig. 4, the CS-ANN and BA-ANN models provide a relatively high value of the coefficient of determination compared with the other models.

**Table 4** Results of the ANN models trained with different algorithms during testing phase

Algorithm	DR			Testing period	
	DR < -0.3	-0.3 < DR < 0.3	DR > 0.3	$R^2$	RMSE
LM	13	47	40	0.428	893
GA	20	77	3	0.519	887
ICA	13	83	7	0.543	779.9
BA	13	77	10	0.661	771
CS	12	77	10	0.693	734





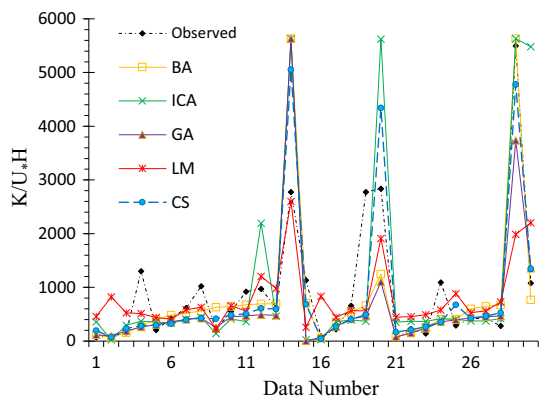


Fig. 3 Predictions of the different models developed in this study

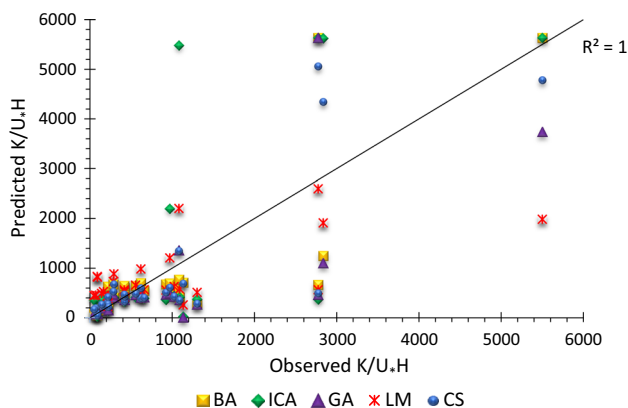


Fig. 4 Scatter plots of the predicted values versus observed values

A comparison between the LM-ANN models with those of the GA-, BA-, ICA- and CS-based ANN models demonstrated that the metaheuristic algorithms are superior over the LM-ANN model. Besides from the intrinsic characteristics of metaheuristic algorithms which are superior over the gradient-based algorithms (the metaheuristics are advantages due to not sticking in local minimum, higher speed and divergence), they have been trained with larger number of datasets which results in better performance. The ANN model trained with metaheuristic employed 80% of the whole dataset for the model training, while the LM based applied 60% of the data for training. It is because the LM-ANN model needs a separate dataset for validation to prevent the network overfitting.

Based on the practical formulas in the literature, the best model performances achieved so far provide an accuracy of 66% and correlation coefficient of 0.6, which are lower than those of CS-ANN and BA-ANN models. Noori et al. (2009) developed SVM and ANFIS models and achieved correlation coefficient of 0.7 and 0.71, respectively. Sahay (2011) achieved an accuracy of 65% by using ANN models. Therefore, the CS-ANN model constructed in this study with, respectively, an accuracy and correlation

coefficient of 77% and 0.83 outperforms those of the previous studies.

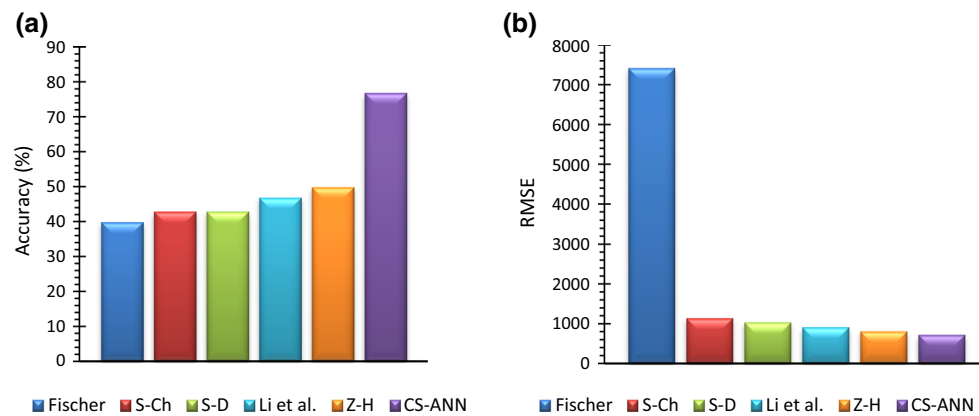
To provide more comparisons of the efficiency of the proposed model (CS-ANN model), results of the CS-ANN model have been compared with some of the empirical equations in terms of the accuracy, RMSE and  $R^2$ . The results are illustrated in Figs. 5 and 6.

Figure 5 illustrates the results of the different models in terms of accuracy and RMSE, respectively. As it can be observed, the cuckoo search algorithm-based ANN model significantly outperforms the other models in terms of accuracy. The CS-ANN model has the accuracy of 77% in which it is higher than the best empirical-based model (Zeng and Huai 2014) with the accuracy of 50%. Also, the CS-ANN model has the lowest value of RMSE equal to 734 which demonstrates the model superiority over the compared equations considering the RMSE criterion. Regarding the scatter plot (Fig. 6), the predictions obtained by the CS-ANN model which suggested in this study represent the highest correlation with the observed values of  $\frac{K}{U_*H}$ . As illustrated in the figure, the coefficient of determination for the CS-ANN model is higher than those of the empirical equations remarkably. The highest value of the coefficient of determination for the empirical equations has been achieved as 0.5 which is much lower than the  $R^2$  obtained for the CS-ANN model.

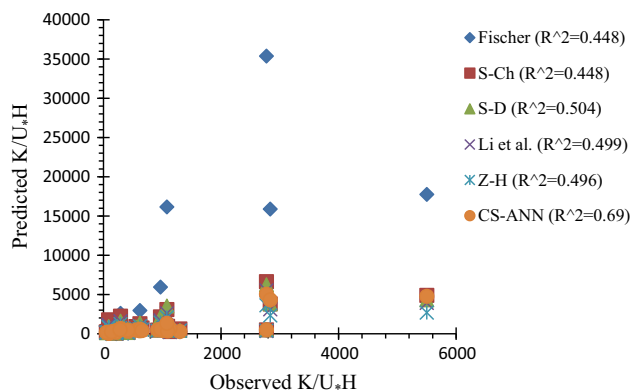
Results of previous studies indicated that black box models such as ANN models are superior over empirical-based models for predicting longitudinal dispersion coefficient. The black box models outperform the statistical methods including regression-based models because of their great ability dealing with nonlinear problems. However, the ANN models applied for the prediction of dispersion coefficient were trained via gradient-based algorithms. Also, it was found that the performance of the ANN models can be improved significantly when metaheuristic algorithms are employed for different applications. In this regard, the proposed models developed for this study are expected to have a higher performance than those of the previous studies.

### Conclusion

A reliable prediction of longitudinal dispersion coefficient in rivers can bring useful information for environmental scientists and river engineers. Anyhow, it is not an easy task due to nonlinear relationship between input and target variables and inherent complexity of the phenomenon. Recently, ANNs have been applied as an efficient approach to deal with nonlinear problems. Concerning ANNs, the training process is known as one of the most important stages to construct an accurate ANN model. In this regard, different training



**Fig. 5** Results of different models for **a** the accuracy, **b** the RMSE



**Fig. 6** Scatter plot of the predicted versus observed values for different models. *Note* Fischer: Fischer et al. (1979); S-Ch: Seo and Cheong (1998); S-D: Sahay and Dutta (2009); Li et al.: Li et al. (2013); and Z-H: Zeng and Huai (2014), which are represented in Table 1 with the row no. of 3, 4, 7, 10 and 11, respectively

algorithms can be utilized in which backpropagation algorithms are probably the most common type of it due to their simplicity. However, they have some disadvantages such as premature and slow convergence. To overcome this deficiency of the BP algorithms, various optimization algorithms such as metaheuristic algorithms have been investigated for ANN training and interesting results have been obtained. In the present study, an attempt was made to examine the ability of four different metaheuristic algorithms (genetic algorithm, imperialist competitive algorithm, bee algorithm and cuckoo search algorithm) for training ANNs. Expansive field data were applied to derive longitudinal dispersion coefficient in natural rivers.

The results obtained through the study were compared with the LM-ANN model. It was observed that ANN models successfully predict longitudinal dispersion coefficient in natural streams. Training ANNs with metaheuristic algorithms increased the model performance significantly. A remarkable improvement in the model performance was observed when BA, ICA and CS algorithms were applied

for training ANNs. These models have a great ability for predicting extreme values of dispersion coefficient in rivers. The proposed models are applicable for a wide range of river geometric and hydraulic characteristics.

Considering the model performance evaluation criteria including discrepancy ratio, coefficient of determination and RMSE, the CS-ANN model is realized as the most efficient model for predicting dispersion coefficient in a wide range of river geometric and hydraulic characteristics. The results of this study indicated that applying CS-ANN model provides a relatively high accurate prediction of dispersion coefficient with accuracy,  $R^2$  and RMSE of 77%, 0.693 and 734  $m^2/s$ , respectively.

Findings of this study indicated that the proposed ANN models provide more accurate predictions compared with the previous studies and those of the empirical equations. The CS-ANN model developed in this study has the highest values of the accuracy (equals to 77%) and correlation coefficient (equals to 0.83) and the lowest value of RMSE as 734 compared with the equations applied for predicting longitudinal dispersion coefficient.

This study presented a new application of cuckoo search algorithm which rarely applied to train ANN models. As the results of the CS-ANN models were superior over the other metaheuristic algorithms, it is suggested to investigate more applications of this algorithm to examine its efficiency in other problems and compare the results with other metaheuristics and optimization algorithms as well.

**Acknowledgments** We thank Ali Afghantoloe for assistance with the training algorithms.

## References

- Ardalan Z, Karimi S, Poursabzi O, Naderi B (2015) A novel imperialist competitive algorithm for generalized traveling salesman problems. *Appl Soft Comput* 26:546–555

- Asadnia M, Chua LH, Qin X, Talei A (2013) Improved particle swarm optimization-based artificial neural network for rainfall-runoff modeling. *J Hydrol Eng* 19:1320–1329
- Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: *Evolutionary computation, 2007. CEC 2007. IEEE Congress on. IEEE*, pp 4661–4667
- Azamathulla HM, Wu F-C (2011) Support vector machine approach for longitudinal dispersion coefficients in natural streams. *Appl Soft Comput* 11:2902–2905
- Bishop CM (1995) *Neural networks for pattern recognition*. Oxford University Press, Oxford
- Chau K (2004) River stage forecasting with particle swarm optimization. In: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, Heidelberg, pp 1166–1173
- Chen Y-h, Chang F-J (2009) Evolutionary artificial neural networks for hydrological systems forecasting. *J Hydrol* 367:125–137
- Chen X, Chau K, Busari A (2015) A comparative study of population-based optimization algorithms for downstream river flow forecasting by a hybrid neural network model. *Eng Appl Artif Intell* 46:258–268
- Cheng C-T, Wang W-C, Xu D-M, Chau K (2008) Optimizing hydropower reservoir operation using hybrid genetic algorithm and chaos. *Water Resour Manag* 22:895–909
- Cheng C-t, Niu W-j, Feng Z-k, Shen J-j, Chau K-w (2015) Daily reservoir runoff forecasting method using artificial neural network based on quantum-behaved particle swarm optimization. *Water* 7:4232–4246
- Danandeh Mehr A, Kahya E, Şahin A, Nazemosadat M (2015) Successive-station monthly streamflow prediction using different artificial neural network algorithms. *Int J Environ Sci Technol* 12:2191–2200
- Disley T, Gharabaghi B, Mahboubi A, McBean E (2015) Predictive equation for longitudinal dispersion coefficient. *Hydrol Process* 29:161–172
- Düğenci M, Aydemir A, Esen İ, Aydın ME (2015) Creep modelling of polypropylenes using artificial neural networks trained with Bee algorithms. *Eng Appl Artif Intell* 45:71–79
- Elder J (1959) The dispersion of marked fluid in turbulent shear flow. *J Fluid Mech* 5:544–560
- Etemad-Shahidi A, Taghipour M (2012) Predicting longitudinal dispersion coefficient in natural streams using M5' model tree. *J Hydraul Eng* 138(6):542–554
- Fischer HB, List JE, Koh CR, Imberger J, Brooks NH (1979) *Mixing in inland and coastal waters*. Elsevier, Amsterdam
- Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput* 29:17–35
- Goldberg DE (1989) *Genetic algorithm in search, optimization and machine learning*. Addison Wesley Publishing Company, Reading, pp 1–9
- Hagan MT, Menhaj MB (1994) Training feedforward networks with the Marquardt algorithm. *IEEE Trans Neural Netw* 5:989–993
- Ham FM, Kostanic I (2000) *Principles of neurocomputing for science and engineering*. McGraw-Hill Higher Education, New York
- Jafarzadeh A, Pal M, Servati M, FazeliFard M, Ghorbani M (2016) Comparative analysis of support vector machine and artificial neural network models for soil cation exchange capacity prediction. *Int J Environ Sci Technol* 13:87–96
- Kashefipour SM, Falconer RA (2002) Longitudinal dispersion coefficients in natural channels. *Water Res* 36:1596–1608
- Kaveh A, Talatahari S (2010) Optimum design of skeletal structures using imperialist competitive algorithm. *Comput Struct* 88:1220–1229
- Kayarvizhy N, Kanmani S, Uthariaraj R (2013) Improving Fault prediction using ANN-PSO in object oriented systems. *Int J Comput Appl* 73:0975–8887
- Li X, Liu H, Yin M (2013) Differential evolution for prediction of longitudinal dispersion coefficients in natural streams. *Water Resour Manag* 27:5245–5260
- Liu H (1977) Predicting dispersion coefficient of streams. *J Environ Eng Div* 103:59–69
- Lucas C, Nasiri-Gheidari Z, Tootoonchian F (2010) Application of an imperialist competitive algorithm to the design of a linear induction motor. *Energy Convers Manag* 51:1407–1411
- Najafzadeh M, Tafarjoruz A (2016) Evaluation of neuro-fuzzy GMDH-based particle swarm optimization to predict longitudinal dispersion coefficient in rivers. *Environ Earth Sci* 75:1–12
- Noori R, Karbassi A, Farokhnia A, Dehghani M (2009) Predicting the longitudinal dispersion coefficient using support vector machine and adaptive neuro-fuzzy inference system techniques. *Environ Eng Sci* 26:1503–1510
- Noori R, Deng Z, Kiaghadi A, Kachoosangi FT (2015) How reliable are ANN, ANFIS, and SVM techniques for predicting longitudinal dispersion coefficient in natural rivers? *J Hydraul Eng* 142:04015039
- Pham D, Ghanbarzadeh A, Koc E, Otri S, Rahim S, Zaidi M (2005) *The bees algorithm*. Technical note Manufacturing Engineering Centre, Cardiff University, UK, pp 1–57
- Sahay RR (2011) Prediction of longitudinal dispersion coefficients in natural rivers using artificial neural network. *Environ Fluid Mech* 11:247–261
- Sahay R, Dutta S (2009) Prediction of longitudinal dispersion coefficients in natural rivers using genetic algorithm. *Hydrol Res* 40(6):544–552
- Seo IW, Cheong TS (1998) Predicting longitudinal dispersion coefficient in natural streams. *J Hydraul Eng* 124:25–32
- Tayfur G (2009) GA-optimized model predicts dispersion coefficient in natural channels. *Hydrol Res* 40(1):65–78
- Tayfur G, Singh VP (2005) Predicting longitudinal dispersion coefficient in natural streams by artificial neural network. *J Hydraul Eng* 131:991–1000
- Taylor G (1953) Dispersion of soluble matter in solvent flowing slowly through a tube. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol 1137. The Royal Society, pp 186–203
- Taylor G (1954) The dispersion of matter in turbulent flow through a pipe. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol 1155. The Royal Society, pp 446–468
- Tutmez B, Yuceer M (2013) Regression Kriging analysis for longitudinal dispersion coefficient. *Water Resour Manag* 27:3307–3318
- Wang W-C, Chau K-W, Cheng C-T, Qiu L (2009) A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *J Hydrol* 374:294–306
- Wang W-C, Cheng C-T, Chau K-W, Xu D-M (2012) Calibration of Xinjiang model parameters using hybrid genetic algorithm based fuzzy optimal model. *J Hydroinform* 14:784–799
- Xiao Z, Liang S, Wang J, Chen P, Yin X, Zhang L, Song J (2014) Use of general regression neural networks for generating the GLASS leaf area index product from time-series MODIS surface reflectance. *IEEE Trans Geosci Remote Sens* 52:209–223



- Yang X-S, Deb S (2009) Cuckoo search via Lévy flights. In: Nature and biologically inspired computing, 2009. NaBIC 2009. World Congress on, 2009. IEEE, pp 210–214
- Yildiz AR (2013) Cuckoo search algorithm for the selection of optimal machining parameters in milling operations. *Int J Adv Manuf Technol* 64:55–61
- Zeng Y, Huai W (2014) Estimation of longitudinal dispersion coefficient in rivers. *J Hydro-environ Res* 8:2–8
- Zhang J-R, Zhang J, Lok T-M, Lyu MR (2007) A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Appl Math Comput* 185:1026–1037

