Water Supply

Water Supply Vol 22 No 2, 1533 doi: 10.2166/ws.2021.333

Prediction of longitudinal and transverse profiles of pressure flushing cones using artificial intelligence and data pre-processing

Mehdi Daryaee [a,*], Farshad Ahmadi [b], Peyman Peykani[a] and Mohammadreza Zayeri[a]

[a] Department of Water Structures, Faculty of Water and Environmental Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran
[b] Department of Hydrology and Water Resources, Faculty of Water and Environmental Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran
*Corresponding author. E-mail: m.daryaee@scu.ac.ir

MD, 0000-0003-4304-0240; FA, 0000-0001-7387-0224

ABSTRACT

One of the most critical issues in dam reservoir management is the determination of sediment level after flushing operation. Artificial intelligence (AI) methods have recently been considered in this context. The present study adopts four AI approaches, including the Feed-Forward Neural Network (FFNN), Cascade Feed-Forward Neural Network (CFFNN), Gene Expression Programming (GEP), and Bayesian Networks (BNs). Experimental data were exploited to train and test the models. The results revealed that the models were able to estimate the post-flushing sediment level accurately. FFNN outperformed the other models. Furthermore, the importance of model inputs was determined using the τ-Kendall (τ–k), Random Forest (RF), and Shannon Entropy (SE) pre-processing methods. The initial level of sediment was found to be the most important input, while the orifice output flow rate was observed to have the lowest importance in modeling. Finally, inputs of higher weights were introduced to the FFNN model (as the best predictive model), and the analysis of the results indicated that the exclusion of less important input variables would have no significant impact on model performance.

Key words: flushing, pre-processing methods, sediment depth

HIGHLIGHTS

- Using artificial intelligence for reservoir operation.
- Prediction of flushing half-cone dimensions.
- Determination of effective parameters in pressure flushing.

INTRODUCTION

Dams are constructed for different purposes across the world, e.g., water storage, floodwater control, and energy generation. The construction of dams on rivers reduces the water stream velocity in the reservoir to nearly zero. This leads to the settlement of stream-carried sediments near the dam body within the reservoir. These sediments disturb the functions of dam components, such as the turbine, embedded intake, and bottom outlet. Also, a significant portion of the reservoir is occupied by the sediments over time, reducing the service life of the dam. According to the International Commission on Large Dams (ICOLD), sedimentation reduces 0.5–0.75% of the total reservoirs of dams on average every year. Several methods have been proposed to remove dam sediments and revive the sedimentation-destroyed reservoir capacities, including bypass systems, turbidity current venting, dredging, and flushing (Schleiss *et al.* 2016). The most frequent sediment removal technique is flushing, in which sediments are washed away by water flowing through the bottom outlet. Flushing operations are defined in two types, free flushing and pressure flushing. The former completely drains the reservoir to the bottom output level, while the latter keeps the reservoir water level almost unchanged (Shen 1999). In practice, pressure flushing is more often employed than free flushing since free flushing may cause environmental damage downstream due to the complete drainage of the reservoir. Free flushing is commonly applied to small reservoirs. The removal of sediment in pressure flushing is dependent on several factors, including the reservoir water level, sediment size and material, bottom outlet size and shape, and the initial level of sediments. Many field and experimental studies have been conducted on the effects of these factors on the output

sediment of the reservoir in pressure flushing (Emamgholizadeh *et al.* 2006; Meshkati Shahmirzadi *et al.* 2010; Powell & Khan 2012; Emamgholizadeh & Fathi-Moghadam 2014).

The sediments settled near the dam body begin moving when the bottom outlet is opened and pressure flushing is started. The movement of the sediments is initially driven by the shear stress on the bottom and then the vortexes forming outlet upstream (Powell & Khan 2012). Once pressure flushing is completed, a semi-conical hole forms in the orifice upstream, which is known as the flushing half-cone.

The water flow-induced movement of sediments on the bed is dependent on a number of factors. It is impossible to empirically relate the influential parameters due to complexities. Thus, researchers have evaluated the feasibility of utilizing artificial intelligence (AI) models in this respect. However, AI is not a new idea (Vaghefi *et al.* 2020). Kaya (2010) modeled the observed pattern of the scour hole arising from sediment movement around piers by using an Artificial Neural Network (ANN). Ramezani *et al.* (2015) predicted the quantity of water-transported sediments on the river bed by using ANNs. Spiliotis *et al.* (2018) studied the initiation threshold of sediment movement based on flow turbulence and a fuzzy model. Kisi *et al.* (2009) adopted a neuro-fuzzy approach and predicted the suspending sediment load of rivers with high accuracy. Najafzadeh & Oliveto (2021) investigated the scour pattern around a pier group by using experimental data and an AI model. Rathod & Manekar (2020) developed a Gene Expression Programming (GEP) model for estimating local scour around bridge piers using laboratory and field data. Sajedi Hosseini *et al.* (2020) reported high performance of flash-flood modeling using three machine learning methods (i.e. boosted generalized linear model (GLMBoost), random forest (RF), and Bayesian generalized linear model (BayesGLM)).

To achieve the reservoir's restored capacity and adopt steps to minimize downstream environmental effects, it is important to estimate the flushing removal of sediments. Hence, researchers have sought to develop empirical equations based on experimental and field data to estimate the removal of sediments based on the aforementioned factors (Fathi-Moghadam *et al.* 2010; Meshkati Shahmirzadi *et al.* 2010). Since pressure flushing involves a complex mechanism of sediment movement and water-sediment interaction, equations developed based on regression methods may not be accurate enough. As a result, AI can be useful as a practical instrument in this respect. Few studies have investigated the utilization of AI models in flushing applications. Emamgholizadeh *et al.* (2013) evaluated the flushing cone volume and length by using the ANN and Adaptive Neuro-Fuzzy Inference System (ANFIS) techniques and experimental data. They reported that the ANN and ANFIS models had high predictive accuracy. Li *et al.* (2016) simulated flushing in the reservoir of the Three Gorges Dam, China, using an ANN model. They concluded that the ANN model was significantly capable of relating the prediction of the output sediment to the influential parameters.

A review of the literature indicated that flushing studies mostly focused on the prediction of the discharged sediments after flushing. As mentioned, increasing of the sediment level in the vicinity of the dam disturbs the functions of dam components. Thus, it is essential to not only predict the output sediment quantity but also the post-flushing sediment level; i.e. the level of sediment at different distances from the bottom outlet after flushing. It should be noted that the collection of topographic data of the reservoir bed to measure the sediment level would be expensive and time-consuming and require equipment. Also, the numerical simulation of pressure flushing in order to predict the flushing cone size using computational fluid dynamics (CFD) would be significantly time-consuming due to large reservoir sizes. The present work aims to study the feasibility of using AI models to predict the sediment level of reservoirs after pressure flushing and identify the influential parameters through data pre-processing methods, i.e. τ-Kendall (τ–k), Random Forest (RF), and Shannon Entropy (SE).

MATERIALS AND METHODS

Data collection

To simulate the pressure flushing in laboratory settings and collect the required data, a flume with a length of 250 cm, a width of 120 cm, and a height of 100 cm was employed. Figure 1 illustrates a schematic of the flume. The variables included the flow rate (Q), the water level in the flume to the orifice center (H_w), sediment level from the flume bottom (H_s), and sediment size (D_{50}). Also, a fixed orifice diameter D_o was applied to the tests. The values and ranges of the variables were determined based on the flume size and laboratory conditions. Sediments were leveled at a given level. Then, water flowed into the flume at a specific flow rate. The test began once the water reached the predefined level and started exiting through the orifice. The flow rate and water level were kept unchanged during the test. The tests continued until the semi-cone size remained unchanged. Once each test had been completed, the scour depth was measured in the longitudinal and transverse profiles at five points Z_x

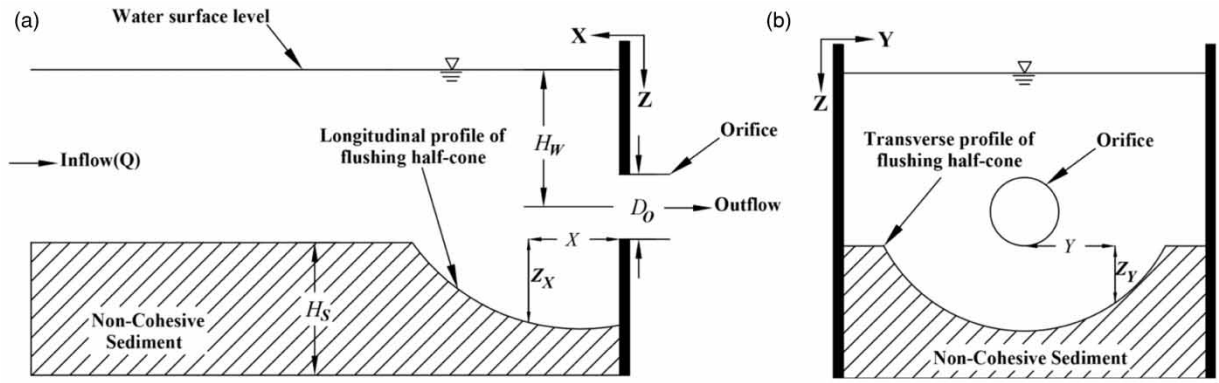

Figure 1 | Laboratory flume from the (a) side and (b) front views.

in the x-direction and 13 points Z_y in the y-direction, as shown in Figure 1. A total of 168 and 462 measurements were performed in the x- and y-directions, respectively. Table 1 reports the input and output variables of the models along with statistical indexes.

Seventy percent of the data were used as the training dataset (i.e. 117 data points for the longitudinal profile and 323 data points for the transverse profile), and the remaining 30% were utilized as the testing dataset (i.e. 51 data points for the longitudinal profile and 139 data points for the transverse profile). To compare the models in terms of accuracy in the training and testing stages, the same inputs were introduced to the models. It should be mentioned that the testing dataset to examine model performance was not used in the training stage.

Artificial neural networks (ANNs)

ANNs have been a popular AI approach in recent decades. It has been employed in several applications, including engineering. An ANN is a computational instrument that can relate parameters influencing a complex phenomenon. It consists of a large number of simple processing components and is inspired by the human brain (Raikar *et al.* 2004). A set of input data is employed to train the ANN model, and then the model can make predictions. In summary, an ANN operation begins with the processing of data in neurons (nodes), and signals are transferred through links between nodes. Each link has a specific weight based on the importance of its nodes. Each node utilizes a nonlinear activation function to determine the output signal from the input signal (Raikar *et al.* 2004). Most multi-layer ANNs consist of an input layer, one or more hidden layers, and an output layer, where each layer can involve several neurons. Many ANN techniques have been introduced. There are different classifications of ANN techniques. In general, ANNs are classified into feed-forward (perceptron) networks, competitive networks, and recurrent networks (Vaghefi *et al.* 2020). The present study employed the Feed-Forward Neural Network (FFNN) and Cascade Feed-Forward Neural Network (CFFNN) techniques to predict the longitudinal and transverse profiles of the flushing cone. FFNN and CFFNN have similar network structures. These two approaches differ in signal transfer between neurons; FFNN transfers neurons only from the input layer toward the output layer, whereas CFFNN is not limited to one-way transfers, and each layer is linked with both previous and next layers (Warsito *et al.* 2018).

Table 1 | Statistical indexes of the measurements

	Input variables						Output variables	
Statistical parameters	Q (lit/s)	H_S (cm)	H_W (cm)	D_{50} (mm)	Longitudinal profile X (cm)	Transverse profile Y (cm)	Longitudinal profile Z_X (cm)	Transverse profile Z_Y (cm)
Min	2.000	20.000	15.000	0.250	0.000	0.000	0.100	0.100
Max	4.000	25.000	56.000	0.750	12.000	34.000	9.600	9.600
Mean	3.000	22.500	36.500	0.500	4.130	17.820	3.789	3.454
Std. deviation	0.670	1.829	15.468	0.200	3.165	8.136	2.456	2.493

Gene expression programming (GEP)

In recent years, new genetic algorithm systems have been developed. These powerful algorithms are inspired by natural evolution and can be employed in a wide range of applications. These algorithms are structurally classified into three groups (Ferreira 2002): (1) algorithms with individuals that have linear chromosomes with a fixed length and non-complex expression. Chromosomes maintain their individual characteristics based on superiority, which is known as Genetic Algorithms (GAs); (2) algorithms with individuals of branched structures with different sizes and shapes and the capability of accepting more factors, which is referred to as Genetic Programming (GP); and (3) individuals are programmed in the form of expressible linear chromosomes with a fixed length, branching structures, and a variety of sizes and forms. Chromosomes are maintained based on the superiority of causal factors on the phenotype (branched structures), which is known as Genetic Expression Programming (GEP).

This classification shows the relationship between GEP and GP, which are both utilized in the form of branched structures in the evolution of computer programs (Mehdizadeh *et al.* 2020). As with GA and GP, GEP is a genetic-based algorithm. It uses a population of individuals, selects them based on their fitness, and applies genetic changes by one or more genetic operators. As mentioned, the natures of individuals are the main difference between these three classes of algorithms. A GA has individuals as linear strings with a length fixed (i.e. chromosomes), GP has nonlinear bodies with different sizes and shapes (i.e. parsing trees), and GEP codes individuals as linear strings with a fixed length (genomes or chromosomes) and describes them as nonlinear bodies with different sizes and shapes (i.e. simple diagram depiction or tree expression) (Ferreira 2002). GEP involves five stages in solving a problem: (1) selecting the terminal set (independent variables and input variables), (2) selecting the function set, including arithmetic operators, testing functions, and Boolean functions, (3) accuracy measurement indexes, which demonstrate the ability of a model to solve a given problem, (4) control components (quantitative and qualitative variables used to control the execution of programs), and (5) a discontinuance criterion.

Bayesian networks (BNs)

A Bayesian network (BN) is a graphical probabilistic model that depicts a set of variables and their probabilities. The BN is a direct acyclic graph where nodes represent problem variables. The BN is grounded on the calculations of dependent probabilities (Bayes' Rule) as:

$$P(b|a) = P(a|b) \times \frac{P(b)}{P(a)} \tag{1}$$

where $P(a)$ denotes the probability of event a, $P(b)$ is the probability of event b, $P(b|a)$ is the probability of event b provided that event a has occurred, and $P(a|b)$ is the probability of event a provided that event b has happened (McCann *et al.* 2006). A BN consists of three main components: (1) a set of nodes serving as management system variables. These nodes could be continuous or discontinuous variables, fixed numbers, or continuous functions. Each node is divided into several classes. Nodes are either parents or children in general. Several parent nodes can produce a child node (McCann *et al.* 2006); (2) a set of connections. Causal relationships between the variables are created by connections. These relationships are represented by arrows. Nodes are linked by arrows. The lack of an arrow between nodes represents the independence of the variables. Nodes that do not have an arrow are the input parent nodes. Also, if an arrow enters and exits a node, it indicates the status child node, and nodes from which no arrow exits represent the output child nodes. The logical meaning of the arrow that goes from the variable x to the variable y is that the variable x has a direct effect on the variable y (Pollino & Hart 2007); (3) a set of probabilities. Each probability determines decision conditions for variables. These conditions are applied to the nodes through the variables that directly affect the corresponding node (i.e. the parents). Nodes with prior nodes are defined by conditional probability distributions. Otherwise, they are expressed by initial probabilities. The influence of an input node on a particular node in the network is represented by a conditional probability, whereas the initial probability denotes the likelihood of an input variable in a certain class. The initial probability is obtained by the initial data of the variables. The probabilities of the lowest section of the BN are calculated using the total probability rule, and the probabilities of the upper sections are found based on Bayes' rule. In general, node x is unconditional if it has no parent; otherwise, it is conditional (Kuikka & Varis 1997).

Model development

FFNN and CFFNN

As mentioned, there are as many neurons in the input and output layers as input and output variables in the FFNN and CFFNN methods. The present study required a network for the longitudinal profile and a network for the transverse one. Based on the collected data, a network with five neurons in the input layer and a neuron in the output layer was used for each profile (Z_x or Z_y), as shown in Figure 2.

Model accuracy is a function of the number of hidden layers, the number of neurons in the hidden layer, activation function, and training algorithm (Vaghefi *et al.* 2020). There is no solid technique to determine these parameters. Thus, they are calculated by trial and error (Mahmoodi *et al.* 2017). Earlier works demonstrated the Levenberg–Marquardt algorithm to be among the most rapid and acceptable training algorithms among other traditional training techniques (Huang *et al.* 2006). Thus, the present study adopted the Levenberg–Marquardt training algorithm. Also, a sigmoid transfer function was employed in the hidden layer, while a linear transfer function was applied to the output layer. According to Bishop (1995), it is typically not necessary to apply more than one hidden layer. However, the present study changed the number of hidden layers from 1 and 5, with 1–10 neurons in each layer, using trial and error. The highest accuracy was obtained for three hidden layers and eight neurons in each layer. Also, it was observed that the iteration of the model for a given number of hidden layers and neurons would alter accuracy. This was reported in earlier works (Vaghefi *et al.* 2020). Hence, the 50 iterations of the model were executed for each change in the number of hidden layers and neurons to obtain the highest accuracy.

GEP model

As mentioned, the selection of different initial populations affecting a given phenomenon (i.e. training dataset) in order to train the governing mechanism of the phenomenon would not only further complicate the model and increase the occupied memory but also reduce the accuracy of the model. Therefore, to estimate the longitudinal and transverse profiles, it is required to select the most effective observed data as the training dataset. The present study selected the inputs of the other techniques for the GEP model. Table 2 describes the GEP model.

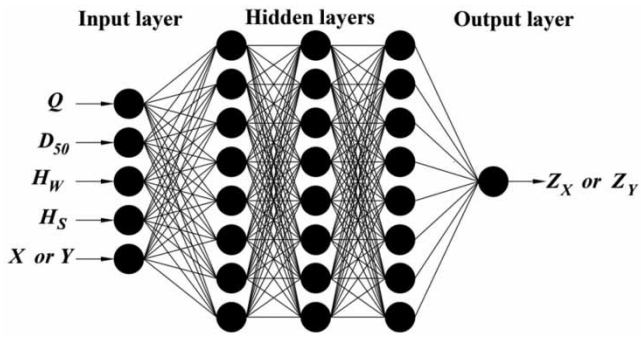

Figure 2 | FFNN structure.

Table 2 | GEP description in longitudinal and transverse profile estimation

Parameter	Value
Head size	8
Number of chromosomes	30
Number of genes	3
Fitness function	*RMSE*
Linking function	Addition

BN model

The structure of a BN is constructed by using the variables and defining a conceptual model to relate the variables. Thus, the network structure was established based on the number of variables. Also, in order to define the conceptual model, it was assumed that Z_x and Z_y in the longitudinal and transverse profile directions, respectively, were influenced by the other variables, as shown in Figure 3. Then, the training dataset was utilized to relate the variables and determine the mathematical form of the relationship. Finally, the performance of the model would be evaluated using the mathematical formulation.

Selection of optimal modeling variables

To employ AI models, it is important to select suitable influential inputs (training dataset) in order to train the governing mechanism. It is expensive and time-consuming to measure all the input variables in real-life settings. Therefore, based on the importance evaluation of parameters, one can apply the most influential parameters to estimation and significantly save time and cost. In order to choose the most valuable inputs for estimating longitudinal and transverse profiles, the Shannon Entropy, τ −Kendall, and Random Forest pre-processing methods were employed.

Shannon entropy (SE)

Entropy is a concept in information theory and refers to the information received from each message. Entropy was introduced by Boltzmann (1896) in thermodynamics. Shannon & Weaver (1949) utilized entropy to express uncertainty. In information theory, entropy is expressed by a given probability distribution P_i. Uncertainty is measured as (Shannon 2001):

$$E_j = S(P_1, P_2, \ldots, P_n) = -K\sum_{i=1}^{m} P_i \mathrm{Ln} P_i \quad i = 1, \ldots, m \tag{2}$$

where K is a constant. The above equation is employed as the entropy of probability distribution P_i in statistical calculations. $P_i = 1/n$ when P_i values are equal for given i and j values.

p_{ij} can be used to evaluate alternatives in a decision matrix. A decision matrix with m alternatives and n indexes (criteria) is formulated as (Shannon 2001):

$$D = \begin{pmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \tag{3}$$

The results of the decision matrix for index j (p_{ij}) is represented as (Shannon 2001):

$$P_{ij} = \frac{a_{ij}}{\sum_{i=1}^{m} r_{ij}}; j = 1, \ldots, n \forall_{ij} \tag{4}$$

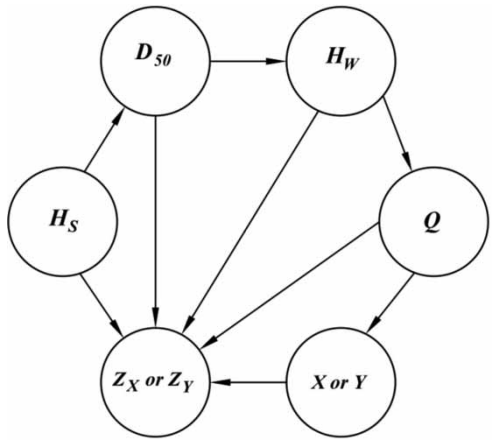

Figure 3 | BN structure.

Entropy E_j is calculated as (Shannon 2001):

$$E_j = -K\sum_{i=1}^{m} P_{ij}\mathrm{Ln}P_{ij}; \forall_{ij} \tag{5}$$

where $K = 1/\mathrm{Ln}m$ is a constant. The deviation degree d_j represents how much useful information is provided by index j to decision-makers and is found as (Shannon 2001):

$$d_j = 1 - E_j; \forall_j \tag{6}$$

Finally, to select the best weight, the weights are calculated as (Shannon 2001):

$$W_j = \frac{d_j}{\sum_{i=1}^{n} d_j}; \forall_j \tag{7}$$

τ–Kendall (τ–k)

The regression coefficient is commonly used to measure the correlations of variables. It is a parametric method. Thus, to employ the regression coefficient method, the dataset needs to meet a number of conditions, including a normal distribution of data. However, this may not be always the case. To solve this problem, the nonparametric $(\tau - \mathrm{k})$ technique can be used. Nonparametric methods are insensitive to data distribution and outliers and thus are more flexible to data (Mehdizadeh *et al.* 2020).

$(\tau - \mathrm{k})$ is a rank correlation coefficient varying in the range of $[-1, 1]$. For a random sample with n pair observations- (x_1, y_1), (x_2, y_2), …, (x_n, y_n), the $(\tau - \mathrm{k})$ estimator is calculated as (Mehdizadeh *et al.* 2020):

$$\hat{\tau} = \binom{n}{2}^{-1} \sum_{1\le i<j\le n} \mathrm{sgn}[(x_i - x_j)(y_i - y_j)] \tag{8}$$

$$\mathrm{sgn}(\psi) = \begin{cases} 1 & \text{if} \quad \psi > 0 \\ 0 & \text{if} \quad \psi = 0 \\ -1 & \text{if} \quad \psi < 0 \end{cases} \tag{9}$$

where $i, j = 1, 2, \ldots, n$, and sgn [.] denotes the sign function.

Random forest (RF)

The RF algorithm is among the latest ensemble learning approaches. It is an extension of the Bagging algorithm, with the difference that feature selection is performed randomly. A set of features is randomly selected in each node. Then, the feature selection steps continue in the feature space. K controls the contribution of randomness in this algorithm. A single feature is selected in each node when K = 1. When K is equal to the total number of features, the decision tree is the deterministic decision tree. The value of K is proposed to be the logarithm of the number of features. Also, a number of works assumed K to be equal to the squared root of the number of features (Gislason *et al.* 2006).

Evaluation of the models

In order to evaluate the performance and accuracy of the models, the Regression coefficient (R), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Kling-Gupta efficiency (KGE), and Willmott's Index (WI) were utilized as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_{mi} - X_{pi})^2} \tag{10a}$$

$$R = \frac{\sum_{i=1}^{N}(X_{mi} - \bar{X}_m)(X_{pi} - \bar{X}_p)}{\sqrt{\sum_{i=1}^{N}(X_{mi} - \bar{X}_m)^2 \sum_{i=1}^{N}(X_{pi} - \bar{X}_p)^2}} \tag{10b}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|X_{mi} - X_{pi}| \tag{10c}$$

$$KGE = 1 - \sqrt{(R-1)^2 + (\alpha - 1)^2 + (\beta - 1)^2} \tag{10d}$$

$$WI = \left|1 - \left[\frac{\sum_{i=1}^{N}(X_{mi} - X_{pi})^2}{\sum_{i=1}^{N}(|X_{mi} - \bar{X}_m| + |X_{pi} - \bar{X}_p|)^2}\right]\right|, 0 \le WI \le 1 \tag{10e}$$

where X_{mi} is the observed value, X_{pi} is the predicted value, $\bar{X}_m$ is the mean observed value, $\bar{X}_m$ is the mean predicted value, R is the correlation coefficient of observed and predicted values, α is the standard deviation ratio (SDR) of X_{mi} and X_{pi}, β is the mean ratio of X_{mi} and X_{pi}, and N is the number of data points. The model with the highest R, KGE, and WI and the lowest RMSE and MAE would be selected as the best model.

RESULTS AND DISCUSSION

The performance evaluation of the models

Table 3 compares the accuracy of the models in the estimation of the longitudinal and transverse profiles. As can be seen, all the models had good performance and accuracy in the longitudinal and transverse profile estimation of the flushing cone. Based on the indexes in the testing stage, FFNN had the lowest estimation error, whereas the BN showed the highest estimation error. In the testing stage, FFNN exhibited an RMSE of 0.245 cm in the estimation of the longitudinal profile. The RMSE of the FFNN model was 43.54, 54.71, and 64.94% lower than those of CFFNN, GEP, and BN, respectively. Furthermore, in transverse profile estimation, FFNN showed a KGE of 0.975 in the testing stage, which was 2.67, 13.02, and 15.89% higher than those of CFFNN, GEP, and BN, respectively. This suggests that FFNN outperformed CFFNN, GEP, and BN.

Figure 4 depicts the measured and predicted longitudinal and transverse profiles in the testing stage. As can be seen, the models had $R > 0.950$ and were able to explain the measurements with good accuracy. FFNN had the highest correlations

Table 3 | Performance comparison of the models

	RMSE (cm)		*MAE* (cm)		*KGE* (–)		*WI* (–)	
Model	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Transverse profile								
CFFNN	0.487	0.612	0.363	0.465	0.942	0.949	0.911	0.891
FFNN	0.276	0.384	0.216	0.310	0.991	0.975	0.947	0.927
BNs	0.465	0.735	0.380	0.820	0.894	0.820	0.907	0.863
GEP	0.702	0.695	0.556	0.582	0.851	0.848	0.864	0.854
Longitudinal profile								
CFFNN	0.327	0.434	0.242	0.340	0.973	0.982	0.941	0.918
FFNN	0.199	0.245	0.139	0.192	0.995	0.979	0.966	0.954
BNs	0.665	0.699	0.532	0.584	0.956	0.955	0.871	0.859
GEP	0.551	0.541	0.451	0.441	0.901	0.909	0.891	0.893

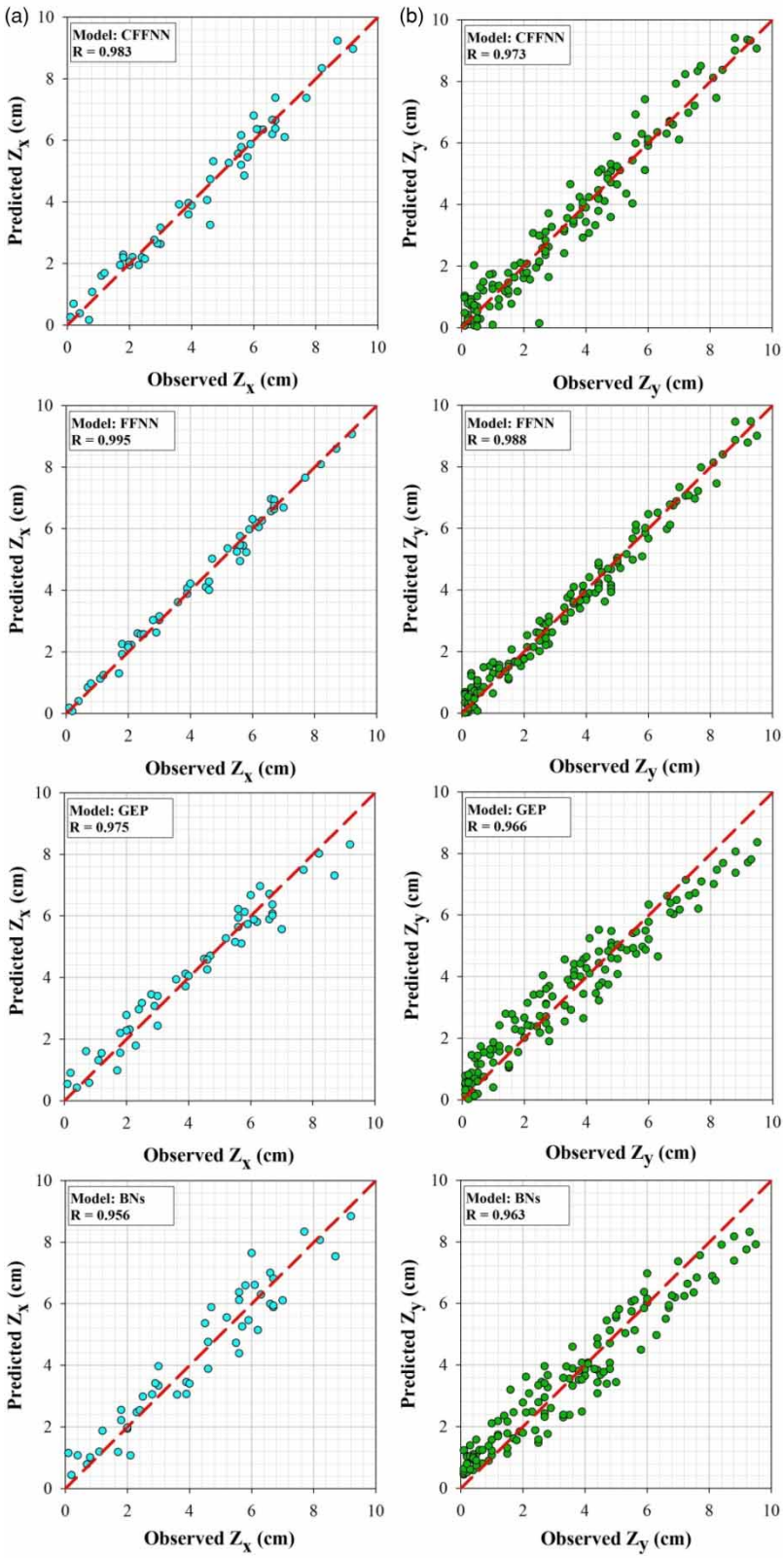

Figure 4 | Scatter plots for the observed versus modeled data for the (a) longitudinal and (b) transverse profiles.

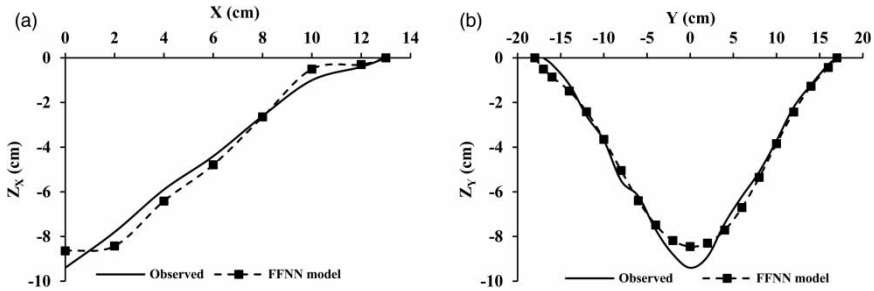

Figure 5 | Comparison of observations and FFNN predictions of the (a) longitudinal and (b) transverse profiles.

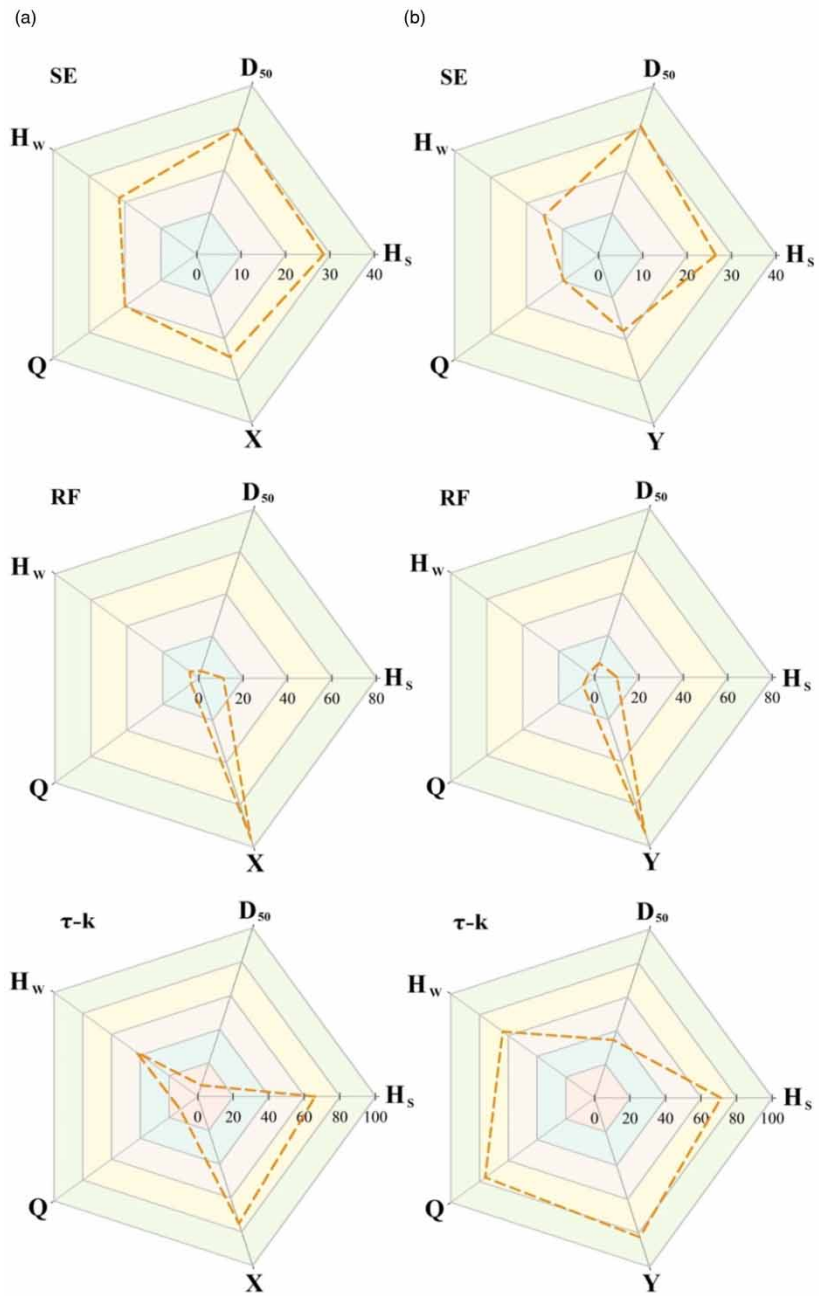

Figure 6 | Pre-processing results for the (a) longitudinal and (b) transverse profiles.

of R = 0.995 in longitudinal profile estimation and R = 0.988 in transverse profile estimation. On the other hand, BN showed the lowest correlations of 0.956 in longitudinal profile estimation and R = 0.963 in transverse profile estimation.

Figure 5 illustrates the FFNN-predicted longitudinal and transverse profiles and testing dataset for one of the tests. As can be seen, the FFNN-predicted profiles are in good agreement with the observations.

Pre-processing results

As mentioned, the present study employed the (τ – k), RF, and SE pre-processing methods to identify the most important inputs, as shown in Figure 6. The longitudinal and transverse profiles are direct functions of inputs X and Y, respectively. Thus, these two parameters were incorporated as influential variables into the model by the pre-processing methods. The pre-processing of inputs revealed that H_s had the highest importance among others in the estimation of the longitudinal and transverse profiles. This is consistent with Emamgholizadeh *et al.* (2013). It can be said that more sediments can be removed through the bottom outlet for a higher initial level of sediment in pressure flushing. As a result, the shapes of the longitudinal and transverse profiles would be affected. On the other hand, Q had the lowest importance, based on the pre-processing results. Table 4 shows the optimal inputs based on the pre-processing results.

According to the results, FFNN had the highest performance and lowest errors. Therefore, the optimal inputs (Table 4) were introduced to the FFNN model. Table 5 represents the evaluation indexes of FFNN under the optimal inputs. According to the testing results, the (τ–k) and SE methods yielded the highest accuracy of FFNN in the estimation of the transverse and longitudinal profiles, respectively, whereas the RF method showed the lowest performance of the model in the estimation of both profiles.

Figure 7 compares the accuracy of the FFNN model in the testing stage between (a) using the entire variables as inputs versus (b) using the optimal inputs. It was observed that the SE and (τ–k) methods led to maximum RMSE values of 0.031 and 0.234 cm, respectively. Thus, it can be concluded that model performance declines as some variables are excluded. However, satisfactory results could be obtained in the initial estimation of the profiles by measuring fewer parameters under an increased acceptable error range.

Additionally, the pre-processing methods proposed different variables in modeling. The SE method proposed the highest number of inputs (except for Q) and yielded better results in the initial estimations. Although the (τ–k) method excluded

Table 4 | Pre-processing-selected input variables

	Transverse profile	Longitudinal profile
SE	H_S, D_{50}, H_W, Y	H_S, D_{50}, X
RF	H_S, Y	H_S, X
τ – k	H_S, H_W, Q, Y	H_S, H_W, X

Table 5 | FFNN evaluation results under the preprocessed inputs

	RMSE (cm)		*MAE* (cm)		*KGE* (–)		*WI* (–)	
Model	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Transverse profile								
SE	0.315	0.415	0.245	0.328	0.988	0.983	0.940	0.923
τ – k	0.528	0.611	0.436	0.521	0.966	0.934	0.893	0.878
RF	0.906	1.089	0.712	0.851	0.894	0.892	0.826	0.800
Longitudinal profile								
SE	0.600	0.741	0.476	0.582	0.957	0.944	0.885	0.859
τ – k	0.408	0.479	0.332	0.370	0.980	0.979	0.919	0.911
RF	0.743	0.820	0.587	0.639	0.932	0.927	0.858	0.845

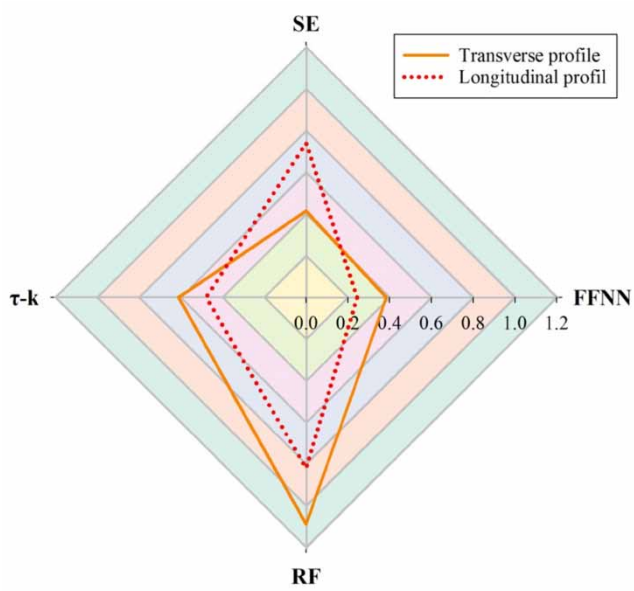

Figure 7 | Radar diagrams for the RMSE values of the FFNN and pre-processed boosted models for the longitudinal and transverse profiles during the test phase.

Q and D_{50}, modeling accuracy did not significantly change. Therefore, it is suggested that studies which may have limited access to D_{50} data exploit the (τ–k)-selected variables.

CONCLUSION

The present study employed the FFNN, CFFNN, GEP, and BN techniques to predict the longitudinal and transverse profiles of the pressure flushing cone. The models were trained using physical modeling outcomes in the laboratory. The evaluation of the models revealed that FFNN and BN had the highest and lowest performance, respectively. Also, the proposed models had higher performance in the estimation of the longitudinal profile than the transverse one. The importance of input variables was measured using pre-processing methods. It was found that H_s and Q had the highest and lowest importance. Then, based on the importance evaluation results, the least important variables were excluded. Finally, the FFNN model was applied to predict the longitudinal and transverse profiles under the preprocessed variables, evaluating the results. It was found that FFNN had the highest performance in the estimation of the transverse and longitudinal profiles under the SE and (τ–k) methods, respectively, and that the exclusion of the least important variables had no significant impact on the accuracy of the FFNN model.

ACKNOWLEDGEMENT

We are grateful to the Research Council of Shahid Chamran University of Ahvaz for financial support (GN: SCU.WH99.31370). The authors are also grateful to the Center of Excellence for the Improvement and Maintenance of the Irrigation and Drainage Networks for provision a part of data.

DATA AVAILABILITY STATEMENT

All relevant data are included in the paper or its Supplementary Information.

REFERENCES

Bishop, C. M. 1995 *Neural Networks for Pattern Recognition*. Oxford University Press, New York.

Boltzmann, L. 1896 Entgegnung auf die wärmetheoretischen Betrachtungen des Hrn. E. Zermelo. *Annalen der Physic* **293** (4), 773–784.

Emamgholizadeh, S. & Fathi-Moghdam, M. 2014 Pressure flushing of cohesive sediment in large dam reservoirs. *Journal of Hydrologic Engineering* **19** (4), 674–681.

Emamgholizadeh, S., Bateni, S. M. & Jeng, D. S. 2013 Artificial intelligence-based estimation of flushing half-cone geometry. *Engineering Applications of Artificial Intelligence* **26** (10), 2551–2558.

Emamgholizadeh, S., Bina, M., Fathi-Moghadam, M. & Ghomeyshi, M. 2006 Investigation and evaluation of the pressure flushing through storage reservoir. *ARPN Journal of Engineering and Applied Sciences* **1** (4), 7–16.

Fathi-Moghadam, M., Emamgholizadeh, S., Bina, M. & Ghomeshi, M. 2010 Physical modelling of pressure flushing for desilting of non-cohesive sediment. *Journal of Hydraulic Research* **48** (4), 509–514.

Ferreira, C. 2002 Genetic representation and genetic neutrality in gene expression programming. *Advances in Complex Systems* **5** (4), 389–408.

Gislason, P. O., Benediktsson, J. A. & Sveinsson, J. R. 2006 Random forests for land cover classification. *Pattern Recognition Letters* **27** (4), 294–300.

Huang, G. B., Zhu, Q. Y. & Siew, C. K. 2006 Extreme learning machine: theory and applications. *Neurocomputing* **70** (1–3), 489–501.

Kaya, A. 2010 Artificial neural network study of observed pattern of scour depth around bridge piers. *Computers and Geotechnics* **37** (3), 413–418.

Kisi, O., Haktanir, T., Ardiclioglu, M., Ozturk, O., Yalcin, E. & Uludag, S. 2009 Adaptive neuro-fuzzy computing technique for suspended sediment estimation. *Advances in Engineering Software* **40** (6), 438–444.

Kuikka, S. & Varis, O. 1997 Uncertainties of climatic change impacts in Finnish watersheds: a Bayesian network analysis of expert knowledge. *Boreal Environment Research* **2**, 109–109.

Li, X., Qiu, J., Shang, Q. & Li, F. 2016 Simulation of reservoir sediment flushing of the three gorges reservoir using an artificial neural network. *Applied Sciences* **6** (5), 148.

Mahmoodi, K., Ghassemi, H. & Nowruzi, H. 2017 Data mining models to predict ocean wave energy flux in the absence of wave records. *Zeszyty Naukowe Akademii Morskiej w Szczecinie* **49** (121), 119–129.

McCann, R. K., Marcot, B. G. & Ellis, R. 2006 Bayesian belief networks: applications in ecology and natural resource management. *Canadian Journal of Forest Research* **36** (12), 3053–3062.

Mehdizadeh, S., Ahmadi, F., Mehr, A. D. & Safari, M. J. S. 2020 Drought modeling using classic time series and hybrid wavelet-gene expression programming models. *Journal of Hydrology* **587**, 125017.

Meshkati Shahmirzadi, M. E., Dehghani, A. A., Sumi, T., Naser, G. & Ahadpour, A. 2010 Experimental investigation of local half-cone scouring against dam. *River Flow* **2010**, 1267–1276.

Najafzadeh, M. & Oliveto, G. 2021 More reliable predictions of clear-water scour depth at pile groups by robust artificial intelligence techniques while preserving physical consistency. *Soft Computing* **25** (7), 5723–5746.

Pollino, C. A. & Hart, B. T. 2007 Bayesian network model in natural resources management. Information sheet prepared by the Integrated Catchment Assessment and Management Centre, the Australian National University.

Powell, D. N. & Khan, A. A. 2012 Scour upstream of a circular orifice under constant head. *Journal of Hydraulic Research* **50** (1), 28–34.

Raikar, R. V., Kumar, D. N. & Dey, S. 2004 End depth computation in inverted semicircular channels using ANNs. *Flow Measurement and Instrumentation* **15** (5–6), 285–293.

Ramezani, F., Nikoo, M. & Nikoo, M. 2015 Artificial neural network weights optimization based on social-based algorithm to realize sediment over the river. *Soft Computing* **19** (2), 375–387.

Rathod, P. & Manekar, V. L. 2020 Gene expression programming to predict local scour using laboratory and field data. *ISH Journal of Hydraulic Engineering*. https://doi.org/10.1080/09715010.2020.1846144.

Sajedi Hosseini, F., Choubin, B., Mosavi, A., Nabipour, N., Shamshirband, S., Darabi, H. & Haghighi, A. T. 2020 Flash-flood hazard assessment using ensembles and Bayesian-based machine learning models: application of the simulated annealing feature selection method. *Science of the Total Environment* **711**, 135–161.

Schleiss, A. J., Franca, M. J., Juez, C. & De Cesare, G. 2016 Reservoir sedimentation. *Journal of Hydraulic Research* **54** (6), 595–614.

Shannon, C. E. 2001 A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review* **5** (1), 3–55.

Shannon, C. & Weaver, W. 1949 *A Mathematical Theory of Communication*. University of Illinois Press, Champaign, IL.

Shen, H. W. 1999 Flushing sediment through reservoirs. *Journal of Hydraulic Research* **37** (6), 743–757.

Spiliotis, M., Kitsikoudis, V., Kirca, V. O. & Hrissanthou, V. 2018 Fuzzy threshold for the initiation of sediment motion. *Applied Soft Computing* **72**, 312–320.

Vaghefi, M., Mahmoodi, K., Setayeshi, S. & Akbari, M. 2020 Application of artificial neural networks to predict flow velocity in a 180 sharp bend with and without a spur dike. *Soft Computing* **24**, 8805–8821.

Warsito, B., Santoso, R. & Yasin, H. 2018 Cascade forward neural network for time series prediction. *Journal of Physics: Conference Series* **1025** (1), 012097. IOP Publishing.

First received 30 June 2021; accepted in revised form 15 September 2021. Available online 29 September 2021