ORIGINAL ARTICLE

Simulation of discharge coefficient of side weirs placed on convergent canals using modern self-adaptive extreme learning machine

Reza Gharib[1] · Majeid Heydari[1] · Saeid Kardar[2] · Saeid Shabanlou[3]

Abstract

Side weirs are broadly used in irrigation channels, drainage systems and sewage disposal canals for controlling and adjusting the flow in main channels. In this study, a new artificial intelligence model entitled "self-adaptive extreme learning machine" (SAELM) is developed for simulating the discharge coefficient of side weirs located upon rectangular channels. Also, the Monte Carlo simulations are implemented for assessing the abilities of the numerical models. It should be noted that the k-fold cross-validation approach is used for validating the results obtained from the numerical models. Based on the parameters affecting the discharge coefficient, six artificial intelligence models are defined. The examination of the numerical models exhibits that such models simulate the discharge coefficient valued with acceptable accuracy. For instance, mean absolute error and root mean square error for the superior model are computed 0.022 and 0.027, respectively. The best SAELM model predicts the discharge coefficient values in terms of Froude number (F_d), ratio of the side weir height to the downstream depth (w/h_d), ratio of the channel width at downstream to the downstream depth (b_d/h_d) and ratio of the side weir length to the downstream depth (L/h_d). Based on the sensitivity analysis results, the Froude number of the side weir downstream is identified as the most influencing input parameter. Lastly, a matrix is presented to estimate the discharge coefficient of side weirs on convergent channels.

Keywords Side weir · Discharge coefficient · Convergent channel · Self-adaptive extreme learning machine · Sensitivity analysis

Introduction

Side orifices, side slide gates and side weirs are divert structures usually installed on the main channel wall for diverting and controlling the flow into the divert channel. This type of hydraulic structures have many applications in environmental and engineering and irrigation practices and also used in wastewater treatment plants, drainage lands, sedimentation tanks, aeration ponds, irrigation networks and flocculation units. Numerous studies have been conducted by different researchers on divert structures. In addition, several researchers such as Ghodsian (2003), Kra and

Merkley (2004), Lewis (2011), Granata et al. (2013) Azimi et al. (2014, 2015), Nezami et al. (2015), Maranzoni et al. (2017) and Karami et al. (2018) have carried out some laboratory, analytical and numerical investigations on hydraulic behavior of side weirs. Hussein et al. (2010) experimentally studied the parameters affecting the passing discharge through circular sharp-crested side orifices. They concluded a relationship as a function of the Froude number and also the ratio of the orifice diameter to the main channel width for computing the discharge coefficient of this type of side orifices. Hussein et al. (2011) conducted an experimental study on the properties of the flow passing through a main channel with rectangular sharp-crested side orifices. Moreover, they also examined the parameters affecting the flow passing through rectangular side orifices. They presented an equation for calculating the discharge coefficient of such side orifices as a function of the Froude number and the ratio of the width of the rectangular side orifice to the main channel width.

Recently, artificial intelligence techniques such as the artificial neural network and fuzzy logic have been utilized as

✉ Majeid Heydari
  mheydari@basu.ac.ir

[1] Department of Water Engineering, Faculty of Agriculture, Bu-Ali Sina University, Hamadan 6517838695, Iran

[2] Department of Architecture, Science and Research Branch, Islamic Azad University, Tehran, Iran

[3] Department of Water Engineering, Kermanshah Branch, Islamic Azad University, Kermanshah, Iran

powerful tools in modeling, pattern-cognition and solving non-linear complex problems (Zambrano et al. 2019; Gholami et al. 2019; Parsaie and Haghiabi 2019). Emiroglu et al. (2010) proposed an equation for calculating the discharge coefficient of triangular labyrinth side weirs placed on a rectangular straight channel in subcritical flow conditions using ANFIS. Bilhan et al. (2010) compared the experimental discharge coefficient of sharp-crested side weirs located on the wall of a rectangular channel in subcritical flow conditions with the discharge coefficient predicted by different neural network techniques. Dursun et al. (2012) concluded the discharge coefficient of semi-elliptical side weirs located on a rectangular straight channel using ANFIS.

As discussed above, side weirs are widely used for conducting and diverting exceeded flows in urban sewage disposal systems and are utilized as flow control structures in flood control systems.

Furthermore, Ebtehaj et al. (2015) predicted discharge coefficient of rectangular side weirs within a main flume by employing gene expression programming (GEP). In addition, a meta-heuristic hybrid artificial intelligence approach was developed by Khoshbin et al (2016) to estimate discharge coefficient of side weirs located on rectangular canals. Besides, Akhbari et al. (2017) applied the radial base neural networks (RBNN) and M5' approaches to model discharge coefficient of triangular weirs in a rectangular conduit. Azimi and Shabanlou (2018) simulated the flow turbulence and water surface nearby a side weir on circular channels in supercritical flow regime through renormalization group (RNG) k-e turbulence model and volume of fluid (VOF) method. By using support vector machine (SVM), discharge coefficients of side weirs installed on trapezoidal channels were modeled by Azimi et al (2019).

Furthermore, usage of various artificial intelligence (AI) techniques is increasing every day so as to model multifarious fields, ranging from hydrological issues to hydraulic problems. It should be stated that discharge coefficient is one of the most important factor to design a side weir properly. Thus, simulation of the discharge coefficient of side weirs located on convergent channels through a new AI approach so-called "self-adaptive extreme learning machine" has enough novelty for both scholar and engineers. This means that, for the first time, the discharge coefficient of rectangular weirs located on convergent channels is simulated by the novel and robust self-adaptive extreme learning machine in this study.

Materials and methods

Extreme learning machine (ELM)

The extreme learning machine (ELM) which is a single-layer feed-forward neural network was provided for the first time by Huang et al. (2004). The ELM randomly determines input weights and also analytically specifies output weights. The only difference between the ELM and the single-layer feed-forward neural network (SLFFNN) is the lack of using biases for the output neuron. Input layer neurons are related with all hidden layer neurons. Hidden layer neurons are created by a bias. The activation function of hidden neurons can be in the form of piecewise continuous function, while this function is linear for output layer neurons. The ELM model uses different algorithms for calculating weights and biases; thus, it leads to decrease the learning time of the network. The mathematical expression of the feed-forward neural network with n hidden node is as follows Huang et al. (2004):

$$f_n(x) = \sum_{i=1}^{n} \beta_i G(a_i, b_i, x) \tag{1}$$

where β_i is the weight between the ith hidden node and the output node, a_i ($a_i \in R^n$) and b_i are the learning factors of hidden nodes and $G(a_i, b_i, x)$ is the ith node output for the input x. The activation function $g(x)$ (with many types) for the additive hidden node $G(a_i, b_i, x)$ can be rewritten as follows Huang et al. (2004):

$$G(a_i, b_i, x) = g(a_i.x + b_i) \tag{2}$$

Activation functions are used for calculating the response of neurons. The behavior of neurons is composed of two parts including the total weighted of inputs and the activation function. When a set of weighted input signals is applied, activation functions are used to achieve the response. Also, the same activation functions are used for the same layer neurons which might be linear and nonlinear. In linear functions, a straight linear graph is drawn while a curved line is drawn for nonlinear. Since the number of input and output variables is not the same in nonlinear functions, classification issues are common among them (Pandey and Govind 2016). Nonlinear ELM activation functions discussed in this paper are: step function (hardlim), sigmoid (sig), sinusoidal (sin), triangular bias (tribas) and radial bias (radbas). In the ELM, weights and biases between hidden and output layer neurons are allocated randomly. The activation of hidden layer neurons for each learning sample in an ELM network with j neurons in the hidden layer, i input neurons and k learning samples are computed as follows Huang et al. (2004):

$$H_{jk} = g\left(\sum (W_{ji}X_{ik}) + B_j\right) \tag{3}$$

where $g(.)$ can be any nonlinear continuous activation function, W_{ji} is the ith input weight and the neuron of the jth hidden layer, B_j is the bias of the jth hidden layer neuron, X_{ik} is the input neuron for the kth learning sample and H_{ik} is the activation matrix of the jth neuron of the hidden layer for the kth learning sample so that the activation of all hidden layer

neurons for samples used in the learning are presented by this matrix. In this matrix, j and k represent the column and row, respectively. The matrix H is expressed as the matrix of the output hidden layer of the neural network. Weights between hidden and output layer neurons are utilized by the fitness of the least square for objective values in the learning mode versus outputs of hidden layer neurons for each learning sample. The mathematical equivalent is expressed as follows Huang et al. (2004):

$$H\beta = T \tag{4}$$

$$\beta = \left(\beta_1, \dots, \beta_j\right)_{j \times 1} \tag{5}$$

where β denotes the weight between the output layer neuron and hidden layer neurons and the vector T represents objective values for learning samples which is expressed as follows Huang et al. (2004):

$$T = \left(T_1, \dots, T_k\right)_{k \times 1} \tag{6}$$

Finally, weights are calculated as follows Huang et al. (2004):

$$\beta = H'T \tag{7}$$

where

$$H(\tilde{a}, \tilde{b}, \tilde{x}) = \begin{bmatrix} G(a_1, b_1, x_1) & G(a_L, b_L, x_L) \\ G(a_1, b_1, x_N) & G(a_L, b_L, x_N) \end{bmatrix}_{L \times N} \tag{8}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad \text{and} \quad T = \begin{bmatrix} T_1^T \\ \vdots \\ T_L^T \end{bmatrix}_{L \times m} \tag{9}$$

In these equations, $\tilde{a} = a_1, \dots, a_L$; $\tilde{b} = b_1, \dots, b_L$; $\tilde{x} = x_1, \dots, x_L$, β is the weight vector between hidden and output layer neurons and H' is the Moore–Penrose inverse of the matrix H. T is the vector between weights of learning samples. It can be said the ELM learning process includes two stages: (1) the random allocation of weights and biases to hidden layer neurons and the calculation of the matrix H hidden layer output and (2) the calculation of output weights using the matrix H Moore–Penrose inverse and objective values for different learning samples. The learning process in finding the Moore–Penrose inverse of the hidden layer matrix (H) is fast. This procedure is faster than common iteration-based algorithms such as Levenberg–Marquardt which does not include any step of nonlinear optimization. Therefore, the learning time of the network is noticeably reduced (Huang et al. 2006). The ELM model works with a large number of nonlinear random input space simulations so that each neuron is related to a single random sample.

Differential evolution

The differential evolution (DE) method is one of the relatively new techniques in the field of meta-search optimization proposed by Storn and Price (1997). In recent years, differential evolution algorithm has been introduced as a powerful and fast method for optimization problems in continuous spaces and has a good ability to optimize nonlinear indifferentiable functions. Like other evolutionary algorithms, this algorithm starts by creating an initial population. Then, by applying operators such as combinations, mutations and intersections, the offspring is formed and in the next step called selection stage, the offspring of the offspring is compared with the offspring for the amount of competency measured by the target function. Then the best members move on to the next stage as the next generation. This process continues until the desired results are achieved.

Self-adaptive extreme learning machine

Using the differential evolution algorithm as self-adaptive has the ability to overcome limitations such as control parameters in the algorithm, choosing the trial vector strategy. Therefore, the robust self-adaptive learning machine algorithm (SAELM) is proposed by Cao et al. (2012) to optimize network input weights and hidden node bias. Having the training dataset, L hidden nodes and the activation function $g(x)$, the SAELM algorithm can be formulated. To this end, the initial population is first represented by population vectors (NPs) which include hidden nodes.

Physical model

In the current study, two groups of data including the experimental values obtained by Bagheri et al. (2014) and Maranzoni et al. (2017) are used for validating the numerical results. Bagheri et al.'s (2014) model is composed of a prismatic channel with a side weir installed on its sidewall. Additionally, Maranzoni et al.'s (2017) experimental model consists of a rectangular convergent channel with a side weir attached to the convergence point of the channel and on the sidewall. The layout of the experimental models is illustrated in Fig. 1.

Discharge coefficient of side orifices located on convergent channels

Maranzoni et al. (2017) considered discharge of side weirs placed on convergent channels (Q_w) as a function of the main channel width before convergence (B), the main channel width after convergence (b_d), the weir height (w), the weir length (L), the crest thickness of the side weir (s), the channel bed slope (S_0), the roughness coefficient of the
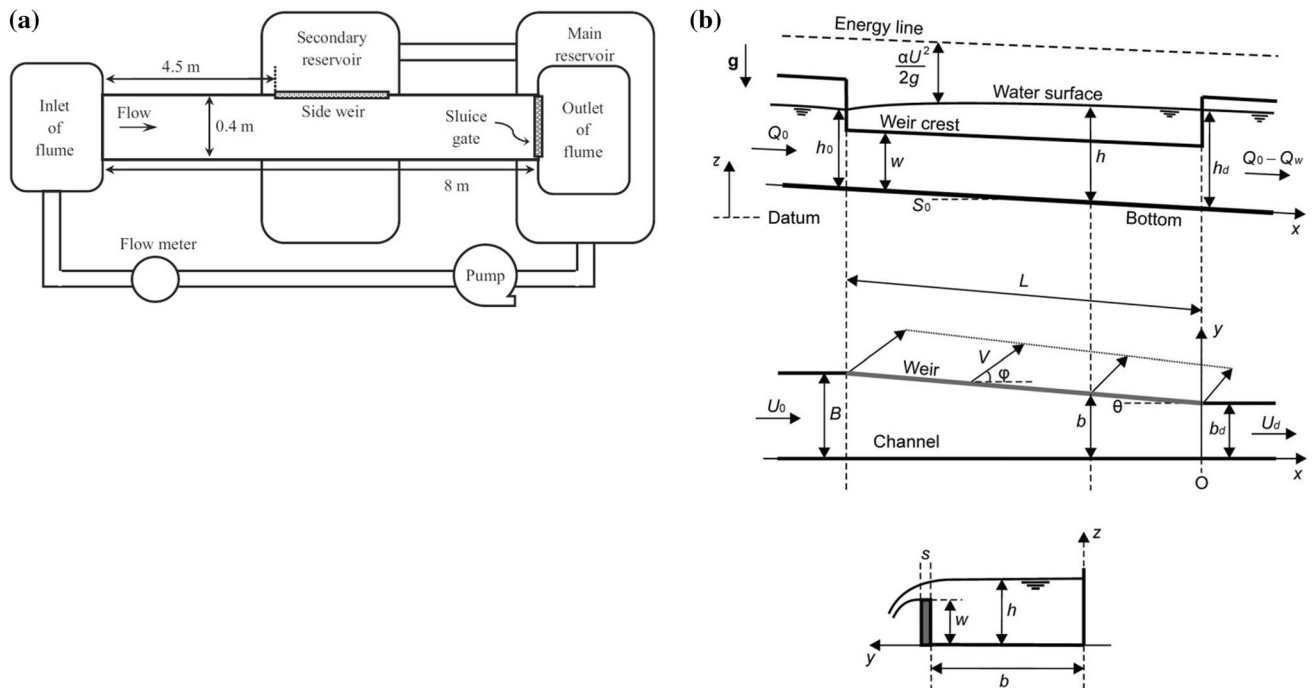
(a)

(b)

Fig. 1 Experimental models **a** Bagheri et al. (2014) and **b** Maranzoni et al. (2017)

channel (ε), the downstream depth of the side weir (h_d), discharge at the side weir downstream (Q_d), viscosity of water (ρ), specific weight (γ), dynamic viscosity (μ) and surface stress tension (σ) (Maranzoni et al. 2017):

$$Q_\mathrm{w} = f_1\left(B, b_\mathrm{d}, w, L, s, S_0, \varepsilon, h_\mathrm{d}, Q_\mathrm{d}, \rho, \gamma, \mu, \sigma\right) \qquad (10)$$

By introducing six groups, they considered the discharge coefficient of side weirs placed upon convergent channels as follows (Maranzoni et al. 2017):

$$Q_\mathrm{w} = f_1\left(\theta, \frac{b_\mathrm{d}}{h_\mathrm{d}}, \frac{w}{h_\mathrm{d}}, \frac{L}{h_\mathrm{d}}, F_\mathrm{d}\right) \qquad (11)$$

In this relationship, θ is the convergence factor and F_d denotes the Froude number at the downstream of the side weir. Thus, the dimensionless parameters of this equation are taken into account as the input parameters of the numerical models. In this study, six different numerical models are defined and the combinations of input parameters are shown in Fig. 2.

In this paper, Monte Carlo simulations (MCs) are used to investigate the capability of numerical models. The Monte Carlo method is a class of computational algorithms that rely on random iterative sampling to calculate their results. Monte Carlo methods are often used when simulating a mathematical or physical system. Because of their reliance on duplicate calculations and random or random numbers, Monte Carlo methods are often configured to be run

by a computer. The tendency to use Monte Carlo methods becomes even more difficult when it is impossible to calculate the exact response using deterministic algorithms. Monte Carlo simulation methods are especially useful in studying systems where there are many variables associated with pair related degree of freedom. In the present study, the algorithms of ANFIS, genetic, and particle swarm optimization have a set of coefficients that provide one response at a time. In general, the purpose of the Monte Carlo method is to adjust the coefficients in a fixed range, which is then
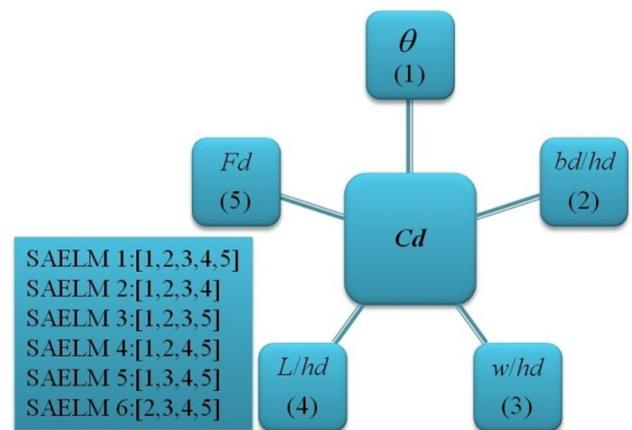

Fig. 2 Combinations of input parameters for self-adaptive extreme learning machine models

presented as the average of all performances (e.g., 1000 runs) as optimized values. If the Monte Carlo method is not used at each step, coefficient values change and do not follow the stump pattern. In general, Monte Carlo is an algorithm that is implemented to optimize the coefficients used in other algorithms in the programming language environment and the statistical distribution is not used. In addition, the multilayer validation method is used for performance evaluation. The aforementioned models are used. In the multilayer validation method, the original sample is randomly divided into k subsamples of equal size. Among subsamples k, one subsample is used as the validation data and the remainder as the test data of this model. The k-fold cross-validation approach is then repeated k times (equal to the number of layers), each of the k subsamples being used exactly once as validation data. The results of the k-layer are averaged and presented as an estimate. The advantage of this method is the random repetition of subsamples in the test and training process for all observations and each observation is used exactly once for validation. In this study, k is assumed to be 4. The outline of the k-fold cross-validation method can also be seen in Fig. 3.

Results and discussion

Criteria for examining accuracy of numerical models

In this study, to study the accuracy of the self-adaptive extreme learning machine models, the statistical indices including mean absolute error (MAE), root mean square error (RMSE), the coefficient of determination (R^2) and mean absolute percent error (MAPE) are used as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| R_{(\text{Pridicted})i} - R_{(\text{Observed})i} \right| \tag{12}$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(R_{(\text{Pridicted})i} - R_{(\text{Observed})i} \right)^2} \tag{13}$$

Iteration 1
Iteration 2
Iteration 3
Iteration k
Train   Test

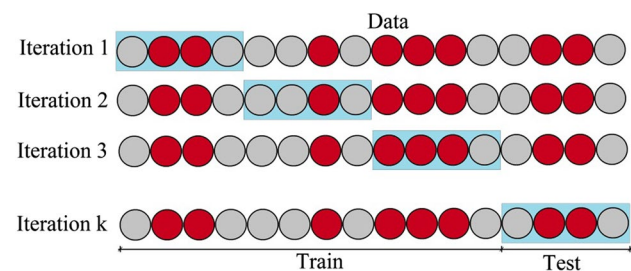**Fig. 3** Dealing of k-fold cross-validation method with experimental values

$$R = \frac{\sum_{i=1}^{n} \left(R_{(\text{Observed})i} - \bar{R}_{(\text{Observed})} \right) \left(R_{(\text{Pridicted})i} - \bar{R}_{(\text{Pridicted})} \right)}{\sqrt{\sum_{i=1}^{n} \left(R_{(\text{Observed})i} - \bar{R}_{(\text{Observed})} \right)^2 \sum_{i=1}^{n} \left(R_{(\text{Pridicted})i} - \bar{R}_{(\text{Pridicted})} \right)^2}} \tag{14}$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\left| R_{(\text{Pridicted})i} - R_{(\text{Observed})i} \right|}{R_{(\text{Observed})i}} \right) \times 100 \tag{15}$$

In the above equations, $R_{(\text{Observed})i}$, $R_{(\text{Pridicted})i}$, $\bar{R}_{(\text{Observed})i}$ and n are experimental dats, results predicted by numerical models, the average of experimental data and the number of experimental measurements, respectively. Desirable value of the MAE, RMSE and MAPE closes to zero, while closeness of the R index to one is ideal.

The presented statistical indices do not provide a simultaneous comparison of average and variance of models. Thus, the Akaike information criterion (AIC) is introduced for comparing predicted and experimental discharge coefficients as follows:

$$\text{AIC} = n \cdot \log \left[\frac{1}{n} \sum_{i=1}^{n} \left(C_{d(\text{Observed})i} - C_{d(\text{Pridicted})i} \right)^2 \right] + 2k \tag{16}$$

where k is the number of estimated parameters used in the numerical model. The parameter ACI is taken into account as a criterion for suitable agreement of a statistical model. In addition, this criterion is used as a tool for choosing them model which simultaneously describes the complexity and accuracy of the numerical model. The best model has the lowest value of the AIC index.

Number of hidden layer neurons

In this section, the number of hidden layer neurons is evaluated. Generally, by increasing the number of hidden layer neurons, the modeling accuracy increases. In this paper, two hidden layer neurons are utilized for modeling the discharge coefficient of side weirs placed on convergent channels. Then, by increasing the number of neurons, the error of each step is calculated and compared with the previous step. This procedure continues until the numerical model error reaches an acceptable range. In Fig. 4, the changes of different statistical indices versus the changes of the number of hidden layer neurons are depicted. As shown in Fig. 4, after reaching the number of the optimized hidden layer neurons to 6, the numerical model error is calculated in an acceptable range. For example, the RMSE and MAPE values of the numerical model in the case where the number of hidden layer neuron is equal to 6 are computed 0.034 and 5.135, respectively. Also, the MAE for this number of hidden layer neurons is calculated 0.027. The coefficient of determination calculated for this

condition is approximated 0.963. Therefore, the number of the hidden layer neurons is considered 6 for the rest of the modeling process.
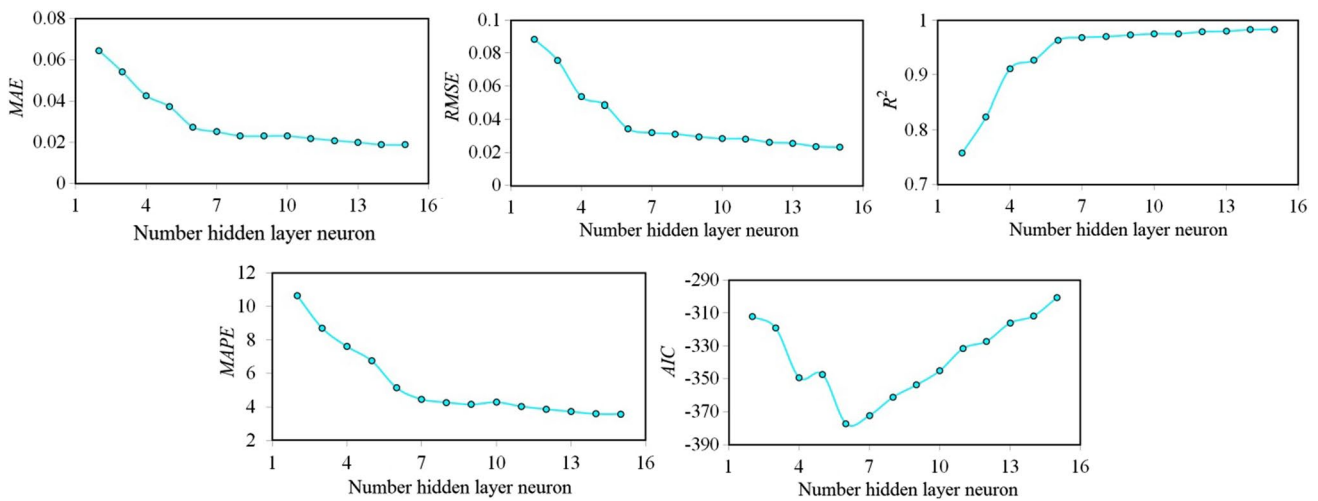
Activation functions

In the next section, the results of the activation functions of the self-adaptive extreme learning machine model are examined. As discussed previously, the self-adaptive extreme learning machine model has five activation functions entitled sigmoid, sine, hardlimit, triangle basis and radial basis. In Fig. 5 and Table 1, different values of statistical indices for the SAELM model activation functions are arranged. Also, the discharge coefficient values simulated by sigmoid, sine, hardlimit, tribas and radbas are given in Fig. 6. For

Table 1 Results of statistical indices for different activation functions of self-adaptive extreme learning machine

Activation functions	R^2	MAE	RMSE	MAPE	AIC
Sig	0.963	0.027	0.034	4.135	377.259
Sin	0.818	0.056	0.077	9.248	275.205
Hardlim	0.812	0.056	0.078	9.012	273.169 −
Tribas	0.584	0.094	0.118	16.915	216.645
Radbas	0.868	0.050	0.065	8.317	296.206

example, the values of R^2, RMSE and MAPE for the sigmoid activation function are computed 0.963, 0.034 and 5.135, respectively. Furthermore, MAE for this activation function

Fig. 4 Changes of different statistical indices versus changes of number of hidden layer neurons
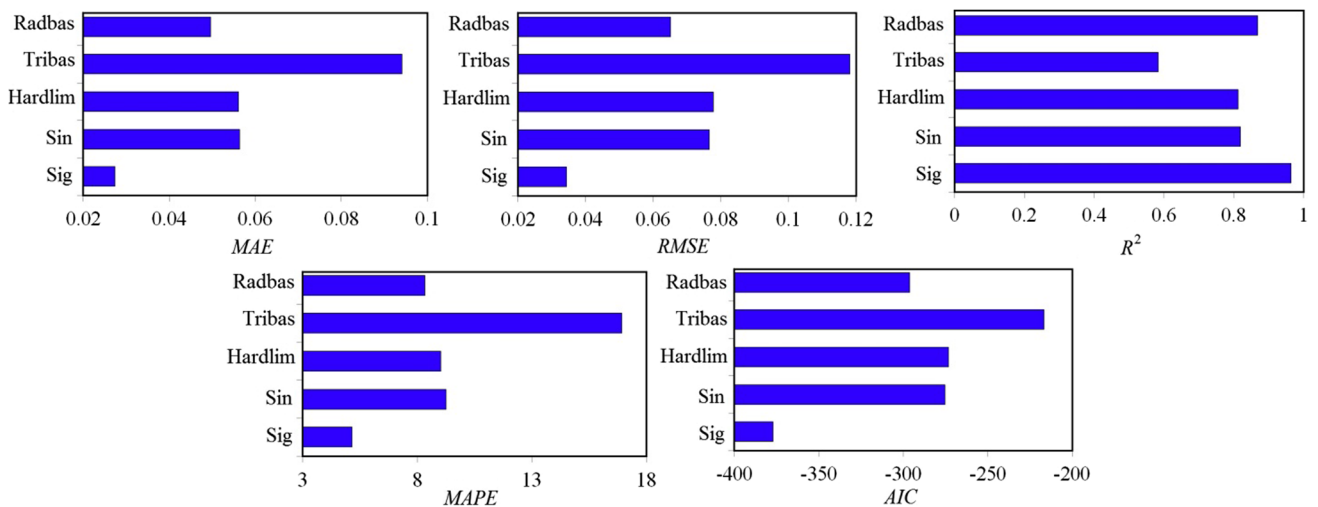
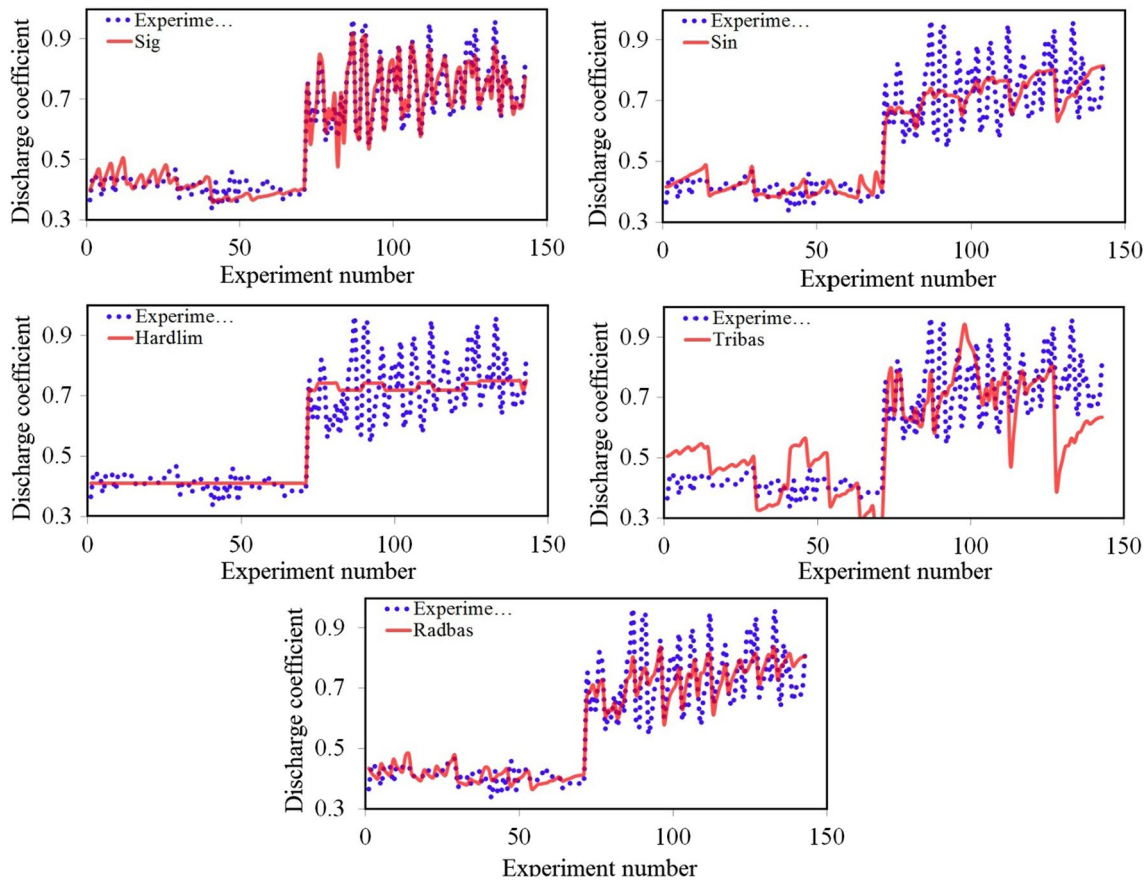**Fig. 5** Comparison of statistical indices for different activation functions of SAELM

Fig. 6 Comparison of observed and experimental discharge coefficients with different activation functions of SAELM

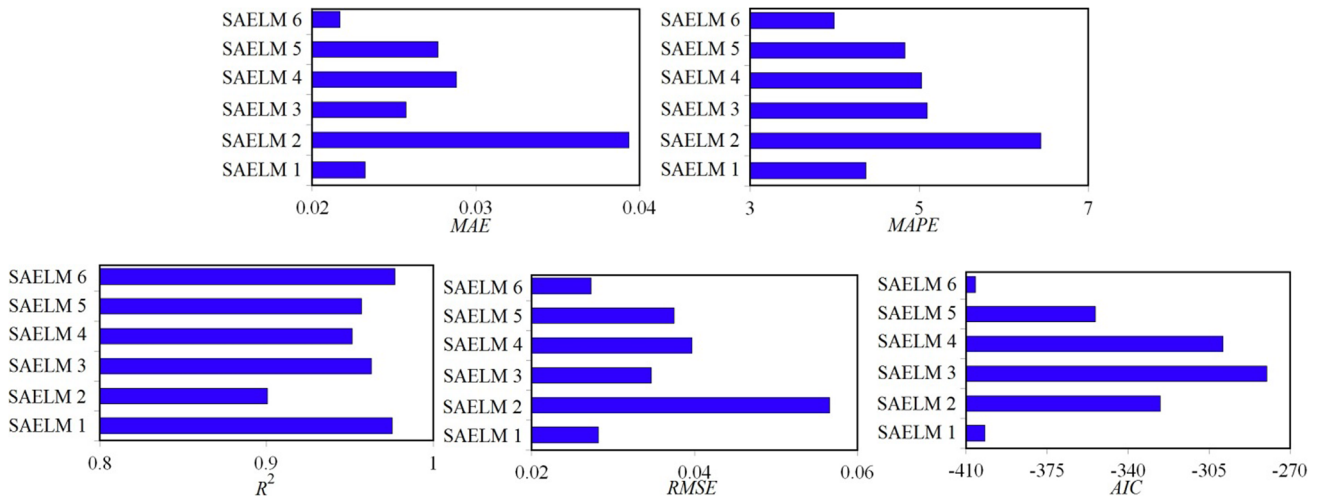**Fig. 7** Changes of different statistical indices for SAELM 1 to SAELM 6 models

**Table 2** Results of different statistical indices of SAELM models

Model	R^2	MAE	RMSE	MAPE	AIC
SAELM 1	0.975	0.023	0.028	4.369	41.826
SAELM 2	0.901	0.039	0.057	6.438	326.181
SAELM 3	0.963	0.026	0.035	5.059	279.956
SAELM 4	0.951	0.029	0.040	5.030	299.019
SAELM 5	0.957	0.028	0.037	4.831	354.203
SAELM 6	0.977	0.022	0.027	3.997	406.018

is estimated to be 0.027. For the sine activation function, the MAE is calculated almost equal to 0.056, while the sine function predicts the ACI and R^2 values -275.205 and 0.818, respectively. It is worth noting that the MAE, R^2 and ACI values for the hardlim activation function are estimated 0.056, 0.812 and -273.169, respectively. The triangle basis activation function calculates the RMSE and MAPE values equal to 0.118 and 16.915, respectively. Furthermore, the value of R^2 for the radial basis activation function is equal to 0.548. Also, the MAE, MAPE and ACI values for this function are calculated 0.050, 8.317 and -296.206, respectively. As can be seen, the sigmoid activation function predicts the discharge coefficient values with higher accuracy than the other activation functions.

Results of sensitivity analysis

In order to conduct the sensitivity analysis, six SAELM models are introduced by combining the input parameters. The results of the statistical indices calculated for different models are shown in Fig. 7. Also, the results of the statistical indices are given in Table 2. For example, the SAELM 1 is a function of all input parameters. For this model, the R^2, MAE and MAPE are estimated 0.975, 0.023 and 4.369, respectively. The ACI for the SAELM 1 is obtained equal to -401.826. Moreover, the scatter plots of different SAELM models are depicted in Fig. 8. In the following, the influence of each input parameter is removed to identify the most effective input parameter. For example, the SAELM 2 model is a function of the parameters L/h_d, w/h_d, b_d/h_d and $.\theta$ In other words, the influence of the Froude number at the downstream of the side weir (F_d) is neglected for this model. For this model, the values of MAE and RMSE are calculated 0.039 and
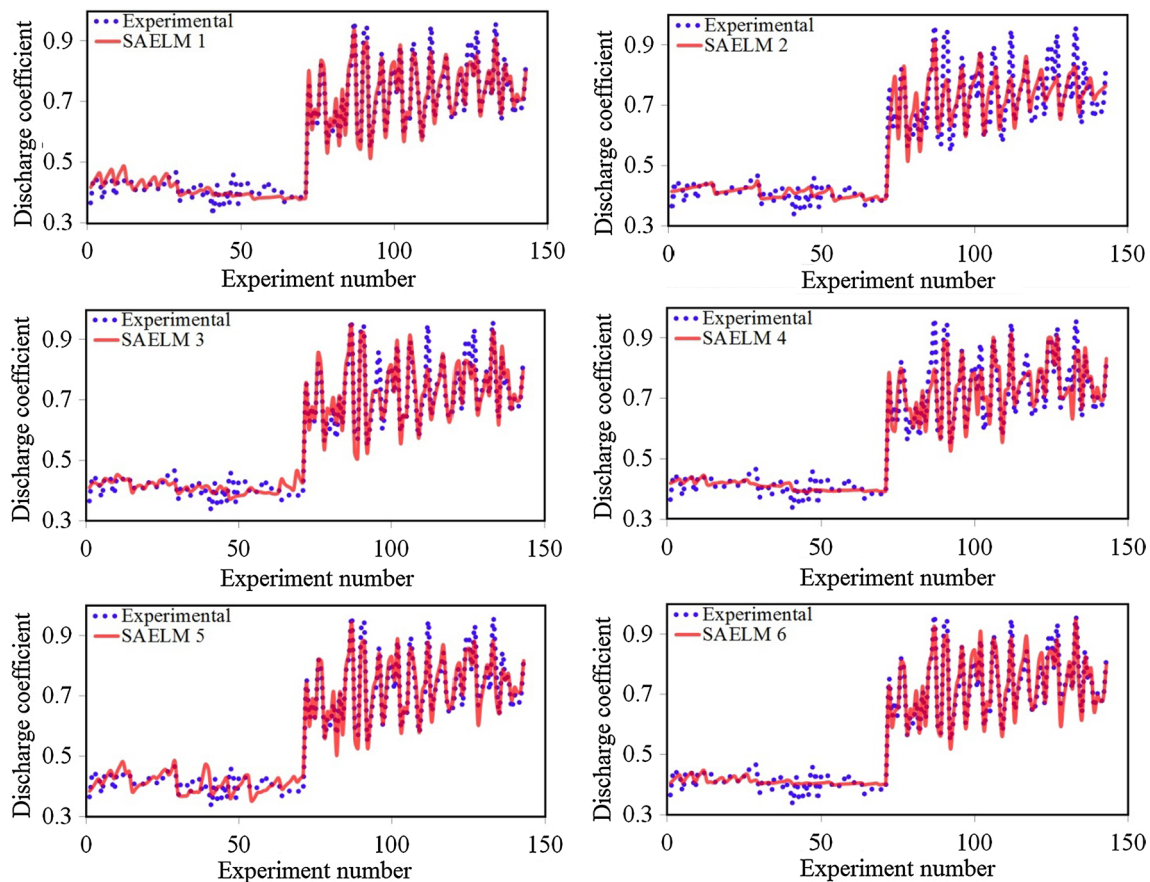
Fig. 8 Comparison of simulated and experimental discharge coefficients for different SAELM models

0.057, respectively. Furthermore, the R^2, MAPE and ACI values for the mentioned model are 0.901, 6.438 and − 326.181, respectively. Among all SAELM model, the SAELM 2 model has the lowest accuracy. Therefore, by eliminating the flow Froude number, the numerical model accuracy is significantly decreases. For the SAELM 3 model, the influence of the input parameter L/h_d is neglected. In other words, this model simulates the discharge coefficient values in terms of F_d‘, w/h_d, b_d/h_d and θ. For instance, the R^2 value for this model is calculated 0.962. Also, the values of MAE and RMSE for the mentioned model are obtained 0.026 and 0.035, respectively. Among the models with four input parameters, this model has the highest correlation with the experimental values. In the following, the SAELM 4 model is evaluated. This model simulates the objective function values in terms of F_d, L/h_d, b_d/h_d and θ. The influence of the parameter w/h_d is removed for this model. This model estimated the R^2 and ACI values equal to 0.951 and − 299.019, respectively. In addition, the RMSE, MAPE and MAE values are 0.040, 5.030 and 0.029, respectively. Also, the RMSE, MAPE and MAE values for the SAELM 5 model are calculated 0.037, 4.831 and 0.028, respectively. The values of R^2 and ACI for the SAELM 5 model are 0.957 and − 354.203, respectively. This model simulates the discharge coefficient values as a function of F_d, L/h_d, w/h_d and θ. It should be noted that for the mentioned model the influence of the dimensionless parameter b_d/h_d is eliminated. The SAELM 6 model predicts the discharge coefficient values in terms of F_d, w/h_d, b_d/h_d and L/h_d and the influence of θ is neglected. Based on the modeling results, the R^2, MAPE and RMSE values for the SAELM 6 model are approximated 0.977, 3.997 and 0.027, respectively. In addition, the ACI value for the mentioned model is computed − 406.018. According to the modeling results, the SAELM 6 model is introduced as the superior model. This model estimates the discharge coefficient values in terms of F_d, w/h_d, b_d/h_d and L/h_d. It is worth noting that the Froude number at the downstream of the side weir (F_d) and w/h_d are, respectively, identified as the most influencing input parameters.

Superior model

In the following, the superior model (SAELM 6) which estimates the discharge coefficient values in terms of F_d, w/h_d, b_d/h_d and L/h_d. The mentioned model has higher accuracy compared to the other artificial intelligence models. Next, a formula is proposed for calculating the discharge coefficient of labyrinth weirs for the superior model as follows:

$$C_d = \left[\frac{1}{(1 + \exp(\text{In } W \times \text{In } V + \text{BHN}))} \right]^T \times \text{Out } W \qquad (17)$$

where In W, In V, BHN and Out W are the matrices of input weights, input variables, the bias of hidden neurons and output weights, respectively. Values of each of these matrices are presented as follows:

$$\text{In } V = \begin{bmatrix} F_d \\ w/h_d \\ b_d/h_d \\ L/h_d \end{bmatrix} \qquad (18)$$

$$\text{In } W = \begin{bmatrix}
-0.152 & -0.655 & 0.269 & 0.674 & 0.585 & -0.243 \\
-0.613 & 0.179 & -1.032 & -0.391 & -0.247 & -0.012 \\
-0.081 & 0.585 & 0.381 & -0.240 & 0.390 & -0.024 \\
-0.354 & -0.261 & -0.062 & 0.456 & -1.334 & -1.026 \\
0.424 & 0.261 & -0.746 & 0.679 & 0.428 & 0.461 \\
0.447 & -0.550 & -1.058 & 0.157 & -0.029 & 0.332 \\
-0.063 & -0.539 & -0.577 & 0.547 & -0.524 & -0.923 \\
-0.643 & -0.847 & 0.639 & 0.550 & -0.370 & -0.842 \\
0.427 & 0.055 & -0.532 & 0.536 & 0.054 & 0.267 \\
-0.288 & -0.265 & 0.299 & 0.199 & 0.388 & -0.768 \\
0.681 & -0.069 & -1.041 & 0.182 & 0.503 & -0.221 \\
0.0737 & -0.338 & -0.907 & 0.113 & 0.915 & -0.177 \\
-0.675 & -0.986 & 0.325 & 0.294 & 0.105 & -0.383 \\
-0.329 & 0.307 & 0.636 & 0.293 & 0.824 & 0.691 \\
0.617 & 0.501 & 0.025 & -0.377 & 0.538 & -0.178 \\
-0.093 & -0.432 & -0.873 & 0.943 & 0.176 & 0.082 \\
0.727 & 0.433 & 0.059 & -0.043 & 0.537 & 0.715 \\
-0.338 & 0.463 & 0.935 & -0.700 & -0.429 & -0.782 \\
-0.330 & -0.255 & 0.558 & -0.235 & 0.056 & 0.136 \\
-1.119 & 0.011 & 0.394 & -0.282 & 0.160 & 0.304 \\
-0.290 & 0.531 & -0.918 & 0.052 & -0.054 & -0.360 \\
0.583 & 0.269 & 0.269 & -0.042 & -0.544 & -0.488 \\
0.236 & 0.365 & -0.581 & -0.562 & -0.198 & 0.085 \\
-0.356 & 0.013 & -0.030 & -1.005 & -0.417 & 0.557 \\
-0.359 & 0.199 & -0.246 & -0.648 & 0.192 & 0.066 \\
0.229 & 0.183 & -0.567 & -0.202 & 0.120 & -0.052 \\
0.079 & -0.516 & 0.654 & 0.551 & 0.468 & 0.235 \\
0.201 & -0.485 & -0.702 & -0.365 & -0.079 & 0.513 \\
-0.489 & 0.001 & 0.264 & 0.224 & -0.520 & -0.627 \\
-0.341 & 0.078 & 0.725 & -0.896 & -0.652 & 0.057
\end{bmatrix}$$

$$\qquad (19)$$

$$
\text{BHN} = \begin{bmatrix}
-0.598 \\
-0.042 \\
-0.456 \\
-0.397 \\
0.354 \\
-0.001 \\
0.650 \\
0.520 \\
-0.683 \\
0.350 \\
0.618 \\
-0.535 \\
0.033 \\
-0.383 \\
0.464 \\
0.275 \\
0.209 \\
0.518 \\
-1.156 \\
0.462 \\
0.161 \\
-0.340 \\
0.488 \\
-0.640 \\
-0.869 \\
-0.016 \\
0.802 \\
-0.132 \\
-0.624 \\
0.339
\end{bmatrix}
\quad (20)
\qquad
\text{Out } W = \begin{bmatrix}
33421.56 \\
13588.23 \\
-49219.20 \\
-86.16 \\
176.90 \\
-11.10 \\
15.64 \\
-3012.74 \\
350252 \\
427.36 \\
11.95 \\
12645.82 \\
-49.77 \\
-793753 \\
54700.22 \\
726.93 \\
173.72 \\
-19918 \\
399888.50 \\
363657.20 \\
337550.30 \\
54469.64 \\
-2137.21 \\
-17.92 \\
226884.50 \\
-31.81 \\
-47.77 \\
-21614.80 \\
-53.29 \\
61249.13
\end{bmatrix}
\quad (21)
$$

Conclusions

In the study, discharge coefficient of side weirs located on convergent flumes simulated by employing a novel approach entitled "self-adaptive extreme learning machine" (SAELM). Then, the parameters affecting the discharge coefficient of side weirs placed upon convergent channels were introduced. In this study, the value of k for the k-fold cross-validation method was taken into account equal to 6. After that, the number of the hidden layer neurons was evaluated. The optimized number of the hidden layer neurons was obtained to be equal to 6. Subsequently, the results of the activation functions for the SAELM model were assessed. Based on the modeling results, the sigmoid activation function predicted the discharge coefficient values with higher accuracy than the other activation functions. To perform a sensitivity analysis, six SAELM models were introduced through the combination of the input parameters. Then, the effect of each input parameter was eliminated to detect the most influencing input parameter. According to the results obtained from the numerical modeling, the SAELM 6 model was identified as the superior model. This model approximated the discharge coefficient values in terms of F_d, w/h_d, b_d/h_d and L/h. Based on the sensitivity analysis results, the values of R^2, MAPE and RMSE for the superior model were estimated 0.977, 3.997 and 0.027, respectively. Based on the sensitivity analysis, the Froude number at the downstream of the side weir and the ratio of the side weir height to the downstream depth of the side weir (w/h_d) were, respectively, detected as the most influencing input parameters.

Compliance with ethical standards

Conflict of interest All authors declare they have no conflict of interest.

References

Akhbari A, Zaji AH, Azimi H, Vafaeifard M (2017) Predicting the discharge coefficient of triangular plan form weirs using radian basis function and M5'methods. J Appl Res Water Wastewater 4(1):281–289

Azimi H, Shabanlou S (2018) Numerical study of bed slope change effect of circular channel with side weir in supercritical flow conditions. Appl Water Sci 8(6):166

Azimi H, Shabanlou S, Salimi MS (2014) Free surface and velocity field in a circular channel along the side weir in supercritical flow conditions. Flow Meas Instrum 38:108–115

Azimi H, Hadad H, Shokati Z, Salimi MS (2015) Discharge and flow field of the circular channel along the side weir. Can J Civ Eng 42(4):273–280

Azimi H, Bonakdari H, Ebtehaj I (2019) Design of radial basis function-based support vector regression in predicting the discharge coefficient of a side weir in a trapezoidal channel. Appl Water Sci 9(4):78

Bagheri S, Kabiri-Samani AR, Heidarpour M (2014) Discharge coefficient of rectangular sharp-crested side weirs part II: Domínguez's method. Flow Meas Instrum 35:116–121

Bilhan O, Emiroglu ME, Kisi O (2010) Application of two different neural network techniques to lateral outflow over rectangular side weirs located on a straight channel. Adv Eng Softw 41(6):831–837

Cao J, Lin Z, Huang GB (2012) Self-adaptive evolutionary extreme learning machine. Neural Process Lett 36(3):285–305

Dursun OF, Kaya N, Firat M (2012) Estimating discharge coefficient of semi-elliptical side weir using ANFIS. J Hydrol 426–427:55–62

Ebtehaj I, Bonakdari H, Zaji AH, Azimi H, Sharifi A (2015) Gene expression programming to predict the discharge coefficient in rectangular side weirs. Appl Soft Comput 35:618–628

Emiroglu ME, Kisi O, Bilhan O (2010) Predicting discharge capacity of triangular labyrinth side weir located on a straight channel by using an adaptive neuro-fuzzy technique. Adv Eng Softw 41(2):154–160

Ghodsian M (2003) Flow through side sluice gate. J Irrig Drain Eng 129(6):458–462

Gholami A, Bonakdari H, Zaji AH, Akhtari AA (2019) An efficient classified radial basis neural network for prediction of flow variables in sharp open-channel bends. Appl Water Sci 9(6):145

Granata F, de Marinis G, Gargano R, Tricarico C (2013) Novel approach for side weirs in supercritical flow. J Irrig Drain Eng 139(8):672–679

Huang G-B, Zhu Q-Y, Siew C-K (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. Proc Int Joint Conf Neural Netw 2:985–990

Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501

Hussein A, Ahmad Z, Asawa GL (2010) Discharge characteristics of sharp-crested circular side orifices in open channels. Flow Meas Instrum 21(3):418–424

Hussein A, Ahmad Z, Asawa GL (2011) Flow through sharp-crested rectangular side orifices under free flow condition in open channels. Agric Water Manag 98:1536–1544

Karimi M, Attari J, Saneie M, Jalili Ghazizadeh MR (2018) Side weir flow characteristics: comparison of piano key, labyrinth, and linear types. J Hydraul Eng 144(12):04018075

Khoshbin F, Bonakdari H, Ashraf Talesh SH, Ebtehaj I, Zaji AH, Azimi H (2016) Adaptive neuro-fuzzy inference system multi-objective optimization using the genetic algorithm/singular value decomposition method for modelling the discharge coefficient in rectangular sharp-crested side weirs. Eng Optim 48(6):933–948

Kra EY, Merkley GP (2004) Mathematical modelling of open-channel velocity profiles for float method calibration. Agric Water Manag 70(3):229–244

Lewis JW, Wright SJ, Pribak M, Sherill J (2011) Bottom slot discharge outlet for combined sewer diversion structure. J Irrig Drain Eng 137(2):248–253

Maranzoni A, Pilotti M, Tomirotti M (2017) Experimental and numerical analysis of side weir flows in a converging channel. J Hydraul Eng 143(7):04017009

Nezami F, Farsadizadeh D, Nekooie MA (2015) Discharge coefficient for trapezoidal side weir. Alexandria Eng J 54(3):595–605

Pandey P, Govind R (2016) Analysis of randomized performance of bias parameters and activation function of extreme learning machine. Int J Comput Appl 135(1):23–28

Parsaie A, Haghiabi AH (2019) Evaluation of energy dissipation on stepped spillway using evolutionary computing. Appl Water Sci 9(6):144

Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359

Zambrano J, Samuelsson O, Carlsson B (2019) Machine learning techniques for monitoring the sludge profile in a secondary settler tank. Appl Water Sci 9(6):146