Neural Comput & Applic (2018) 29:983–989
https://doi.org/10.1007/s00521-016-2588-x

REVIEW

Predicting discharge coefficient of triangular labyrinth weir using extreme learning machine, artificial neural network and genetic programming

Hojat Karami[1] · Sohrab Karimi[1] · Hossein Bonakdari[2] · Shahabodin Shamshirband[3]

Received: 23 February 2016 / Accepted: 6 September 2016 / Published online: 4 October 2016

Abstract Weirs are a type of hydraulic structure used to direct and transfer water flows in the canals and overflows in the dams. The important index in computing flow discharge over the weir is discharge coefficient (C_d). The aim of this study is accurate determination of the C_d in triangular labyrinth side weirs by applying three intelligence models [i.e., artificial neural network (ANN), genetic programming (GP) and extreme learning machine (ELM)]. The calculated discharge coefficients were then compared with some experimental results. In order to examine the accuracy of C_d predictions by ANN, GP and ELM methods, five statistical indices including coefficient of determination (R^2), root-mean-square error (RMSE), mean absolute percentage error (MAPE), SI and δ have been used. Results showed that R^2 values in the ELM, ANN and GP methods were 0.993, 0.886 and 0.884, respectively, at training stage and 0.971, 0.965 and 0.963, respectively, at test stage. The ELM method, having MAPE, RMSE, SI and δ values of 0.81, 0.0059, 0.0082 and 0.81, respectively, at the training stage and 0.89, 0.0063, 0.0089 and 0.88, respectively, at the test stage, was superior to ANN and GP methods. The ANN model ranked next to the ELM model.

Keywords Weir · Discharge coefficient · Artificial neural network · Extreme learning machine · Genetic programming

1 Introduction

Weirs are among the most important hydraulic structures. They pass the floods from dams' reservoirs, divert water from canals and could be used as discharge-measurement devices in the channels. The safety of channels which transfer water and the security of dams are very closely related to the sufficiency of weir's capacity. Most of the damages made to the water-transporting channels and even dam floodways are due to the weirs which don't have enough capacity [1, 2]. When the flow level increases behind the weirs and reaches a level higher than the weir's crest, the flow passes over it. The velocity profile is curved and nonlinear when the flow passes over the weir and if discharge decreases in the channel, the flow over the weir decreases too [3]. As a result, a strong, safe and highly efficient weir structure must be selected with regard to the sensitivity of its function. This structure has to be ready for exploitation at any time. Selecting the discharge coefficient (C_d) of the weirs is generally one of the most important matters, which has an important role in decreasing structural and financial damages caused by floods. Proper understanding of the function of weirs can significantly decrease their construction expenses.

The most important types of weirs include sharp-crested, broad-crested, ogee, labyrinth, shafts, side weirs, stepped weirs and siphon weirs [4–6]. In the last decade, different researchers have used new methods, called soft computing or intelligence methods, which are desirably efficient and accurate, to solve complicated problems related to

✉ Hojat Karami
hkarami@semnan.ac.ir

[1] Department of Civil Engineering, Semnan University, Semnan, Iran

[2] Department of Civil Engineering, Razi University, Kermashan, Iran

[3] Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

discharge capacity of the weirs, based on the hydraulic parameters of flow and geometry of the weirs. The key hydraulic parameters in this respect are Froude number upstream of the weir, flow depth, crest height and weir height. Emiroglu et al. [7], Subramanya and Awasthy [8], Swamee et al. [9], Bagheri and Heidarpour [10], Kisi [11], Bonakdari et al. [12], Emiroglu et al. [13] and Kisi et al. [14] are among the many researchers who calculated C_d of labyrinth weirs by using soft computing methods.

Huang et al. [15] introduced extreme learning machine (ELM) algorithm for single-layer feed-forward artificial neural network (ANN). This algorithm's problem-solving capability is due to use of an algorithm built on gradient descent like the back propagation, which applies in ANN. ELM is highly trained to reduce the time needed for training a neural network. Researchers have observed that by utilizing ELM, the learning process is significantly faster and produces reliable generalization performance [16]. Several researchers [17–22] have used ELM to solve data problems in different scientific fields.

The aim of the present research is precise determination of C_d in a triangular labyrinth weir through three methods of ANN, ELM and genetic programming (GP). Afterward, the resulting C_ds will be plotted and compared with some experimental results, which were found in the literature. Finally, some well-known statistical criteria are used to select the best estimation method.

2 Materials and methods

In this study, three models (ELM, ANN and GP) were designed to estimate C_d of a triangular labyrinth weir. Some brief explanations of these three models are given here.

2.1 Artificial neural network

As a notion, based on human brain's function, an artificial neural network (ANN) is commonly employed to solve complicated problems in a wide range of sciences. In general, an ANN consists of some linked nodes (so-called neurons) and three kinds of constructed layers including an input layer, some hidden layers and an output layer. Each layer is made up of a number of neurons. The number of hidden layers does not follow a fixed rule. However, when the number of neurons in hidden layers is extremely high, an unacceptable long time is taken by the network to train for each value [23–25]. For this reason, by taking into account various numbers of neurons within the hidden layers, different models have been developed through trial and error. The model with the best results will be chosen as the ultimate ANN model. MATLAB software was used to run the ANN model for this study.

2.2 Genetic programming (GP)

GP can be seen as an evolutionary technique, and it is a very challenging task to devise a theory for it. GP was not commonly used as a search technique in the 1990s. Later on, GP led to the evolution of computer programs and commonly has been demonstrated in the memory. Hence, it is represented as a tree structure (Fig. 1). We are able to easily assess and estimate trees in a recursive form. As we know that every node of a tree has an operator function mode and each terminal node contained with an operand which comforts evolution as well as evaluation of mathematical statements [26, 27]. Therefore, GP traditionally prefers making use of programming languages which naturally represent tree structure. Non-tree representations have also been proposed and prosperously executed, like linear GP, which is consistent with more traditional imperative languages [28, 29]. The majority of non-tree representations possess structurally ineffective codes (interns). Such noncoding genes may appear without use since they do not exert any influence on the performance of any individual. However, research studies have shown faster convergence with program representations (like linear GP and Cartesian GP) which render such noncoding genes possible, compared to tree-based program representations which do not possess any noncoding genes. The two primary operators applied in evolutionary algorithms are crossover and mutation. In a crossover operator, which is applied in an individual, one of its nodes is simply substituted with another node that has been chosen out of another individual in the population. In the case of a tree-based representation, switching a node implies that the whole branch is replaced, and therefore, this causes higher performance and improves capability of the crossover operator [28, 29]. Expressions' appearance, which has been retrieved from crossover, is entirely different from their initial parents. However, mutation, which can affect an individual in the population, will be able to replace a whole node in the chosen individual. Moreover, mutation is exclusively able to replace the node's information just to

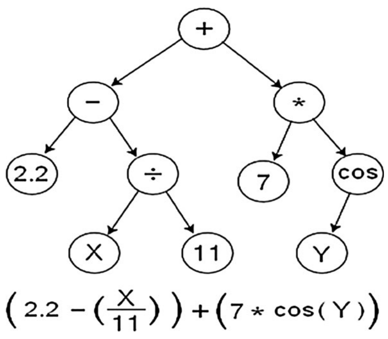

Fig. 1 An example of GP expression tree

keep and retain integrity so that make operations to be fail-safe or furthermore kind of information which is kept by node has to be taken into account. For example, mutation has to know and distinguish binary operation nodes [28, 29]. Otherwise, the operator must be able to make enough efforts to deal with missing values.

2.3 Extreme learning machine (ELM)

ELM is one of the neural network models which has stepped into the spotlight in the recent years. This model has been utilized in a wide range of applications in the preceding decade, owing to its simplicity [30]. Huang et al. [15] proposed an ELM algorithm tool to train single-layer feed-forward neural network (SLFN) architecture. ELM determines random input weight and analytically defines the SLFN's output weight. Since ELM benefits from a faster learning computational analysis and a greater generalized capability, it does not need much intervention during analysis, and therefore, it runs faster than common algorithms. ELM is also able to determine the entire network parameters and hereby minimizes trivial intervention. It can be seen as an effective algorithm with numerous merits such as ease of use, quick learning speed, higher performance and adaptability for varying nonlinear activation and kernel functions. ELM is designed in a way that L hidden neurons constitute SLFN [15]. It is composed of L distinct samples of zero error. Hidden nodes are assigned random values. On the other hand, output weights are computed through pseudo-inverse of H, with minimum error, even in cases which the number of exclusive samples (N) is larger than the number of neurons (L). Random values can be conveyed to the hidden node parameters of ELM a_i and b_i. These two above-mentioned parameters should not be tuned within the training stage.

Theorem 1 According to Liang et al. [22], if we assign a certain SLFN which represent infinitely differentiable in any interval of R, so it has an activation function $g(x)$ as well as RBF hidden nodes or L additive nodes; as a result, two exclusive input vectors are produced for any continuous probability distribution. Henceforth, it shows that $\{x_i | x_i \in R^n, i = 1, \ldots, L\}and\{(a_i, b_i)\}_{i=1}^{L}$ are the two input vectors, respectively, so that is the hidden layer's output matrix which is invertible for probability one; its output matrix H is invertible; then, we have $\|H\beta - T\| = 0$ [15, 19, 22].

Theorem 2 Pursuant to Liang et al. [22], the condition that "any small positive value $\varepsilon > 0$" is an assumption we make and an activation function $g(x)$: $R \rightarrow R$ which serves infinitely differentiable in any interval, then this condition is true $L \le N$ such that for N arbitrary distinct input vectors $\{x_i | x_i \in R^n, i = 1, \ldots, L\}$ for any $\{(a_i, b_i)\}_{i=1}^{L}$ randomly generated based on any continuous probability distribution $\|H_{N \times L}\beta_{L \times m} - T_{N \times m}\| < \varepsilon$ with probability one [19, 22].

Equation (1) is a kind of a linear system; based on aforementioned items, the ELM's hidden node parameters should not be tuned during the training and these parameters can easily be assigned random values. Output weights could be assessed as follows:

$$\beta = H^{+}T \tag{1}$$

where H^{+} represents the Moore–Penrose generalized inverse of the hidden layer output matrix H. Various approaches such as orthogonal projection, iterative and singular value decomposition (SVD) can be utilized for its computation [19]. Only when H^{+} is non-singular and $H^{+} = (H^T T)^{-1} H^T$, then we can employ the orthogonal projection method. Since searching and iterations are used, orthogonalization and iterative methods have restrictions. ELM implementation utilizes SVD to calculate Moore–Penrose generalized inverse of H for it can be employed under any conditions. Therefore, we can conclude that ELM serves as a batch learning method [19].

2.4 Experimental model

The experimental model of Kumar et al. [4] has been used in this research to predict the discharge coefficient of the weirs. The experimental model was a 12-m-long rectangular channel with a width of 0.28 m and a height of 0.41 m. A triangular weir has been used in this experiment (Fig. 2). The weir is placed at 11 m away from the channel entrance. Point gages, with ±0.1 m measurement precision, are used above the weir to measure the water height. A number of pores are installed in the channel wall and in the weir in order to create a nape flow. Grid walls and preventive flows were installed in the upstream of the channel in order to prevent and reduce the formation of vortexes and water-surface disturbance.

The hydraulic parameters of Kumar et al. [4] experiment are listed in Table 1. Table 2 shows the range of the parameters which were used in this study. Number of input data in training mode of the three models was 86 and in the test mode was 37.

2.5 Statistical indices

In order to verify the accuracy of the estimated C_ds by ANN, ELM and GP models, different statistical criteria including coefficient of determination (R^2), root-mean-square error (RMSE), mean absolute percentage error (MAPE), SI and δ are used, as defined in the following equations:

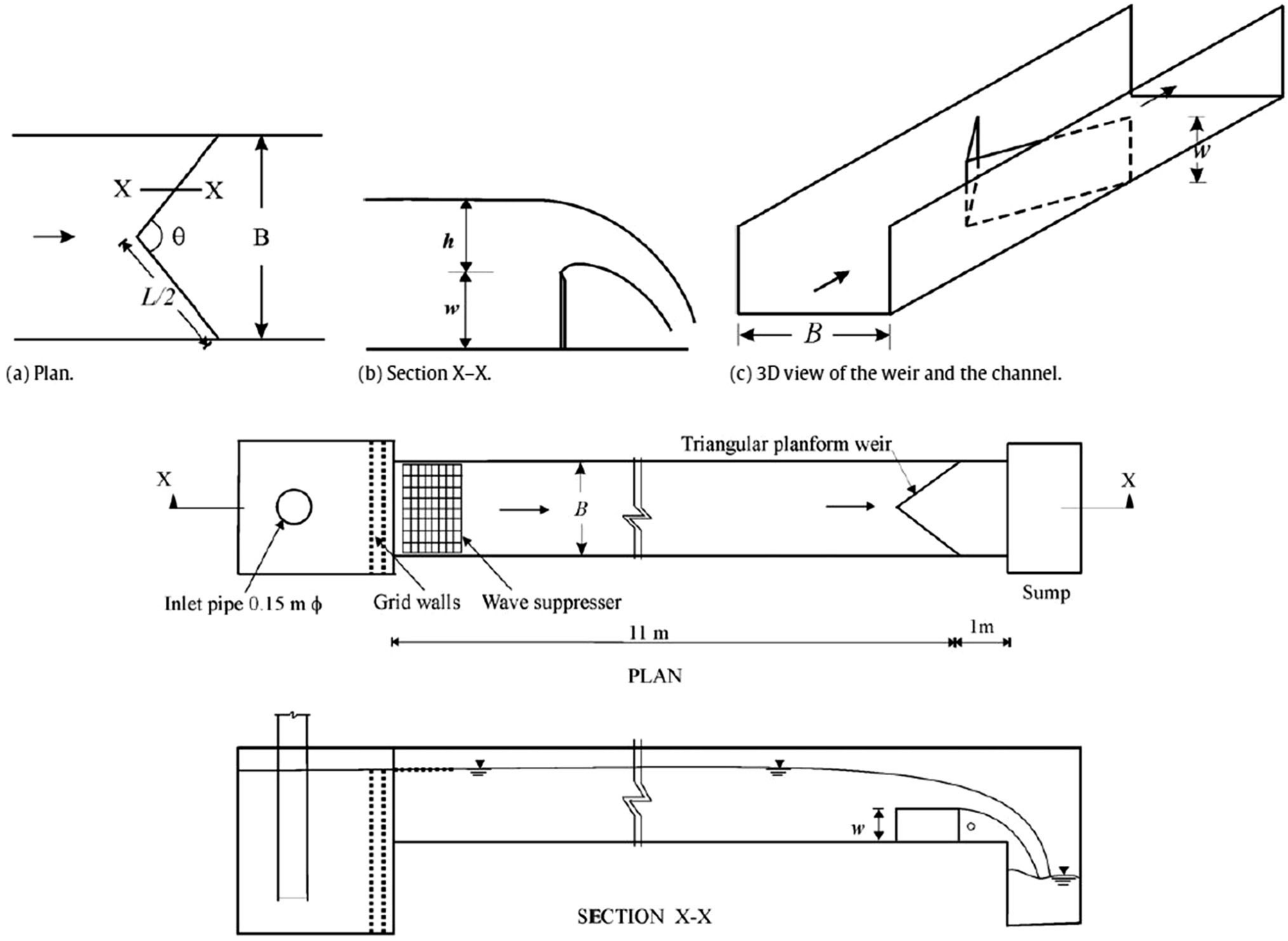

Fig. 2 Plan of the experimental channel used in Kumar et al. [4]

Table 1 Hydraulic parameters used to estimate C_{d} in this study

L	Weir crest length (m)
F	Froude number
B	Channel width (m)
θ	Vortex angle (rad)
w	Crest height (m)
H	Hydraulic head over the weir crest (m)
$y = (h + w)$	Hydraulic head behind the weir (m)

$$R^2 = \left[\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \Big/ \sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}\right]^2 \quad (1)$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2} \quad (2)$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|x_i - y_i|}{x_i} \quad (3)$$

$$\mathrm{SI} = \frac{\mathrm{RMSE}}{\bar{x}} \quad (4)$$

$$\delta\,\% = \frac{\sum_{i=1}^{N}|(y_i - x_i)|}{\sum_{i=1}^{N} y_i} \times 100 \quad (5)$$

where y_i and x_i are predicted (by models) and observed (experimental) C_{d} values, respectively, and $\bar{y}$ and $\bar{x}$ are average predicted and observed C_{d} values, respectively.

Table 2 Parameters used to estimate average discharge coefficient (Kumar et al. [4])

	F	L/h	L/w	h/b	θ (°)	(sin θ × w)/l	y/(sin θ × w)	C_{d}
Min	0.608	3.88	2.68	0.028	30	4.37E−17	1.13	0.54
Max	3.261	135.25	11.76	0.260	180	0.28	1.40E+16	0.906

3 Results and discussions

Figure 3 presents plots of the estimated values of discharge coefficient (C_d) by ELM, GP and ANN models versus calculated experimental values. As shown in this figure, the estimated results confirm fairly well with the experimental values almost in all the three models. It seems from this figure that the ELM model's estimated C_d values are much closer to the experimental C_d values than the ANN and GP models.

Figure 4 indicates that in the training mode, more than 90, 70 and 65 % of the C_d data are estimated with a relative error smaller than 1.5 % in the ELM, ANN and GP models, respectively. In the ELM model, almost 100 % of the C_d data are estimated with an error of <2.5 %. This situation happens in 5.5 % error for ANN and GP models.

In the test mode, the ELM, ANN and GP models' predictions are much closer to each other and are much better than their predictions in the training mode. But again, it could be said that in general, the ELM model acts better than the ANN and GP models (Fig. 5).

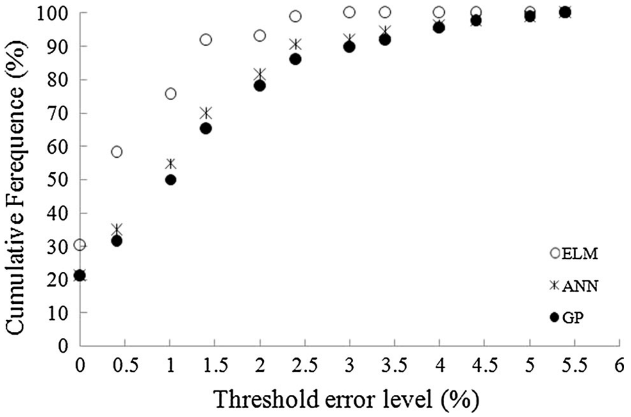

Fig. 4 Error distribution for the three models (training mode)

Tables 3 and 4 are presented to investigate the accuracy of estimating the C_d values through using different statistical indices using the training data and test data, respectively. It can be observed that the values of R^2, RMSE, MAPE, SI and δ are accurate for the three ELM, ANN and GP models. In addition, considering these tables, it could be observed that average relative error is almost 1 % for all three models. For ELM model, R^2 is 0.993 and 0.971 in the training and test modes, respectively, and therefore, this model is the best for prediction of C_d values. The ANN and GP models are ranked next to ELM model. Taking a look at Tables 3 and 4 reveals that the MAPE,

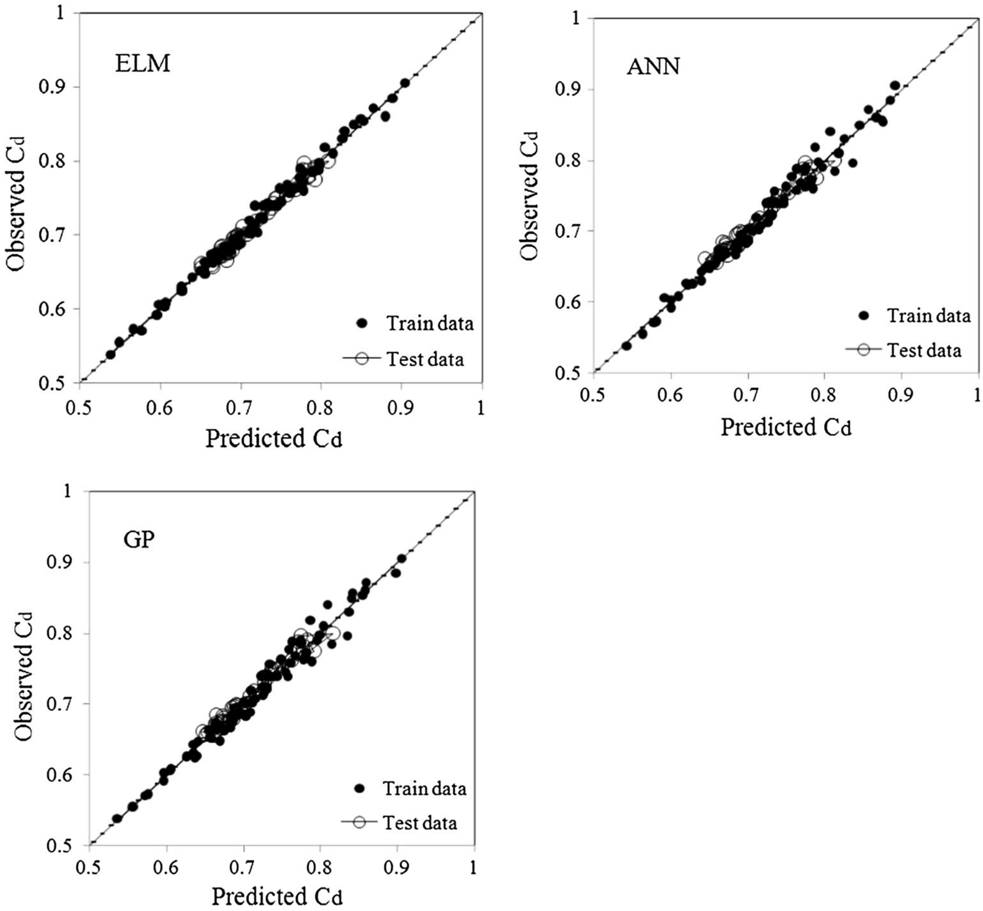

Fig. 3 Comparison of estimated C_ds with experimental results in training and test modes

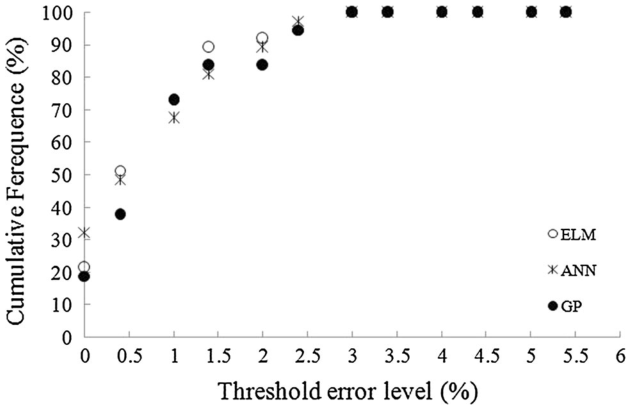

Fig. 5 Error distribution for the three models (test mode)

Table 3 Statistical indices for the three models (training mode)

Index	ELM	ANN	GP
R^2	0.993	0.886	0.884
MAPE	0.81	1.30	1.34
RMSE	0.0059	0.0096	0.0098
SI	0.0082	0.0133	0.0137
δ	0.81	1.30	1.33

Table 4 Statistical indices for the three models (test mode)

Index	ELM	ANN	GP
R^2	0.971	0.965	0.963
MAPE	0.89	0.93	1.01
RMSE	0.0063	0.0067	0.0073
SI	0.0089	0.0094	0.0102
δ	0.88	0.94	1.02

RMSE, SI and δ indices for ELM model are also much smaller than the values for ANN and GP models.

Table 5 shows the predicted C_d values by ELM model and their comparison with the experimental ones. It could be seen in the table that the predicted values do not follow a specific trend; sometimes, it overpredicts and sometimes it under-predicts the C_d values. The point to be noted, however, is that this model predicts relatively well under different hydraulic conditions in such a manner that maximum relative error by this model is approximately 2.27 %.

Table 5 Comparison of the C_d values as predicted by using the ELM model and calculated in the experiment

θ (°)	L (m)	w (m)	y (m)	Q (m^3/s)	C_d (Exp.)	C_d (ELM)
30	1.082	0.092	0.011	0.003	0.860	0.878
30	1.082	0.092	0.017	0.006	0.760	0.777
30	1.082	0.092	0.026	0.009	0.684	0.686
30	1.082	0.092	0.032	0.012	0.625	0.627
60	0.560	0.101	0.013	0.002	0.872	0.863
60	0.560	0.101	0.031	0.006	0.705	0.720
60	0.560	0.101	0.051	0.011	0.573	0.565
60	0.560	0.101	0.029	0.006	0.713	0.715
90	0.396	0.103	0.014	0.002	0.789	0.789
90	0.396	0.103	0.047	0.008	0.702	0.697
90	0.396	0.103	0.069	0.012	0.572	0.577
90	0.396	0.103	0.058	0.010	0.626	0.625
120	0.323	0.106	0.027	0.003	0.791	0.773
120	0.323	0.106	0.044	0.007	0.740	0.738
120	0.323	0.106	0.073	0.012	0.665	0.661
120	0.323	0.106	0.060	0.010	0.697	0.691
150	0.290	0.108	0.014	0.001	0.797	0.798
150	0.290	0.108	0.071	0.011	0.698	0.697
150	0.290	0.108	0.034	0.004	0.796	0.779
150	0.290	0.108	0.052	0.008	0.736	0.738
180	0.280	0.100	0.055	0.007	0.656	0.665
180	0.280	0.100	0.072	0.011	0.675	0.680
180	0.280	0.100	0.045	0.005	0.660	0.665
180	0.280	0.100	0.061	0.008	0.680	0.680

4 Conclusions

Weirs are one of the methods for controlling floods in dam reservoir and diverting and measuring the flow in channels. Discharge capacity over the weir crest is an important hydraulic parameter in this respect. In this research, discharge coefficient of weir has been predicted using three intelligent models of extreme learning machine (ELM), genetic programming (GP) and artificial neural network (ANN). To that end (L/h) (L/w) (h/b) (sin θ), (sin θ) × w/L, and (y/(sin θ) × w), dimensionless parameters were used to train and test the designed models. Results of the ELM, ANN and GP models were compared with some experimental results. Five statistical indices of R^2, RMSE, MAPE, SI and δ were used to compare the predicted and experimental C_d values. The examinations indicated that with an R^2 of 0.993 in the training mode, an R^2 of 0.971 in the test model, and minimum MAPE value of 0.81 % in training mode and 0.89 % in the test mode, the ELM model presents the best results in comparison with the rest of the models. The ANN model also presented relatively good results, similar to those of the ELM model.

References

1. El-Khashab A, Smith KVH (1976) Experimental investigation of flow over side weirs. ASCE J Hydraul Div 102:1255–1268
2. De Marchi G (1934) Essay on the performance of lateral weirs. L’Energia Electtrica 11(11):849–860 **(In Italian)**

3. Borghei SM, Parvaneh A (2011) Discharge characteristics of a modified oblique side weir in subcritical flow. J Flow Meas Instrum 22(5):370–376
4. Kumar S, Ahmad Z, Mansoor T (2011) A new approach to improve the discharging capacity of sharp-crested triangular plan form weirs. J Flow Meas Instrum 22(3):175–180
5. Wormleaton PR, Tsang CC (2000) Aeration performance of rectangular plan form labyrinth weirs. ASCE J Environ Eng 126(5):456–465
6. Emiroglu ME, Baylar A (2005) Influence of included angle and sill slope on air entrainment of triangular plan form labyrinth weirs. ASCE J Hydraul Eng 131(3):184–189
7. Emiroglu ME, Kaya N, Agaccioglu H (2010) Discharge capacity of labyrinth weir located on a straight channel. J Irrig Drain Eng 136(1):37–46
8. Subramanya K, Awasthy SC (1972) Spatially varied flow over side weirs. ASCE J Hydraul Div 98(1):1–10
9. Swamee PK, Santosh KP, Masoud SA (1994) Side weir analysis using elementary discharge coefficient. J Irrig Drain Eng 120(4):742–755
10. Bagheri S, Heidarpour M (2010) Application of free vortex theory to estimate discharge coefficient for sharp-crested weirs. Biosyst Eng 105(3):423–427
11. 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
12. Bonakdari H, Baghalian S, Nazari F, Fazli M (2011) Numerical analysis and prediction of the velocity field in curved open channel using artificial neural network and genetic algorithm. Eng Appl Comput Fluid Mech 5(3):384–396
13. Emiroglu ME, Kaya N, Agaccioglu H (2010) Discharge capacity of labyrinth weir located on a straight channel. J Irrig Drain Eng 136(1):37–46
14. 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
15. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feed forward neural networks. Int Jt Conf Neural Netw 2:985–990
16. Hornik K (1991) Approximation capabilities of multilayer feedforward networks. Neural Netw 4:251–257
17. Leshno M, Lin VY, Pinkus A, Schocken S (1993) Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Netw 6:861–867
18. Huang GB, Babri HA (1998) Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. IEEE Trans Neural Netw 9(1):224–229
19. Tamura S, Tateishi M (1997) Capabilities of a four-layered feedforward neural network: four layers versus three. IEEE Trans Neural Networks 8(2):251–255
20. Huang GB (2003) Learning capability and storage capacity of two hidden-layer feedforward networks. IEEE Trans Neural Netw 14(2):274–281
21. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
22. Liang NY, Huang GB, Rong HJ, Saratchandran P, Sundararajan N (2006) A fast and accurate on-line sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw 17:1411–1423
23. Smith M (1993) Neural Netw for statistical modeling. Van Nostrand Reinhold, New York
24. Dawson CW, Wilby R (1998) An artificial neural network approach to rainfall-runoff modelling. Hydrol Sci J 43(1):47–66
25. Kisi O, Emiroglu ME, Bilhan O, Guven A (2012) Prediction of lateral outflow over triangular labyrinth side weirs under subcritical conditions using soft computing approaches. Expert Syst Appl 39:3454–3460
26. Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. Bradford Book, MIT Press, Cambridge
27. Khan M, Azamathulla HMd, Tufail M, AbGhani A (2012) Bridge pier scour prediction by gene expression programming. Proc Inst Civ Eng Water Manag 165(9):481–493
28. Ferreira C (2001a) Gene expression programming in problem solving. In: 6th online world conference on soft computing in industrial applications
29. Ferreira C (2001) Gene expression programming: a new adaptive algorithm for solving problems. Complex Syst 13(2):87–129
30. Dianhui W, Huang GB (2005) Protein sequence classification using extreme learning machine. Proc Int Jt Conf Neural Netw 3:1406–1411