Hindawi
Advances in Civil Engineering
Volume 2022, Article ID 2495631, 14 pages
https://doi.org/10.1155/2022/2495631

Research Article

Prediction of the Coefficient of Pressure Fluctuations during the Hydraulic Jump Using ELM, GMDH, and M5MT

Tzu-Chia Chen,[1] **Biju Theruvil Sayed**,[2] **Maria Jade Catalan Opulencia**,[3] **Raed H. C. Alfilh**,[4] **Maki Mahdi Abdulhasan**,[5] **and Sayed Hashmat Sadat**[6]

[1]*Department of Industrial Engineering and Management, Ming Chi University of Technology, New Taipei City, Taiwan*
[2]*Department of Computer Science Dhofar University, Salalah, Oman*
[3]*College of Business Administration Ajman University, Ajman, UAE*
[4]*Refrigeration and Air-conditioning Technical Engineering Department, College of Technical Engineering Islamic University, Najaf, Iraq*
[5]*Al-Nisour University College, Baghdad, Iraq*
[6]*Kabul University, Kabul, Afghanistan*

Correspondence should be addressed to Sayed Hashmat Sadat; s.h.sadat@ku.edu.af

Received 23 June 2022; Accepted 28 August 2022; Published 23 September 2022

Academic Editor: Mohammad Najafzadeh

Pressure fluctuations are a critical phenomenon that can endanger the safety and stability of hydraulic structures, especially stilling basins. Hence, the accurate estimation of the dimensionless coefficient of pressure fluctuations (C_P') is critical for hydraulic engineers. This study proposed predictive soft computing models to estimate C_P'on sloping channels. Therefore, three robust soft computing methods, including extreme learning machine (ELM), group method data of handling (GMDH), and M5 model tree (M5MT), were used to estimate C_P' The results revealed that ELM was more accurate than GMDH and M5MT methods when comparing statistical indices, including correlation coefficient (CC), root mean square error (RMSE), mean absolute error (MAE), scatter index (SI), index agreement (I_a), and BIAS values. The performance of ELM was found to be more accurate (CC = 0.9183, RMSE = 0.0067, MAE = 0.0051, SI = 11.88%, Ia = 0.9569) when compared with the results of GMDH (CC = 0.8818, RMSE = 0.0078, MAE = 0.0058, SI = 13.89%, Ia = 0.9361) and M5MT (CC = 0.6883, RMSE = 0.0120, MAE = 0.0090, SI = 21.28%, Ia = 0.7905) in the testing stage. In addition, the BIAS values revealed that ELM slightly overestimated the values of C_P', especially at the peak point compared with GMDH and M5MT results. Overall, the suggested soft computing techniques worked well for predicting pressure fluctuation changes in the hydraulic jump.

1. Introduction

Stilling basins are the most widely used dissipation hydraulic structures of large dams. Energy dissipation in stilling basins is related to hydraulic jumps with high turbulent flow. Hydraulic jump is the common phenomenon for flow energy dissipation in the stilling basins. This phenomenon transforms the supercritical flow into a subcritical flow at a short distance. This is also accompanied by large-scale turbulence, surface waves, air entrainment into the flow, an increase in flow depth, and considerable energy dissipation in the water flow [1, 2]. The turbulent flow within stilling basins is related to the movement of large-scale vortices and severe pressure fluctuations, which may cause significant damage in stilling basins through cavitation, erosion, lifting force, and material fatigue [3]. The importance of pressure fluctuations in hydraulic jumps was revealed after the destruction of the Karnafoli and Malpaso dams in Bangladesh and Mexico, respectively [4]. Therefore, pressure fluctuations in hydraulic structures have included a considerable volume of hydraulic engineers' investigations. Hydraulic models are standard tools for measurements of pressure fluctuations. Experimental studies were carried out for the analysis of characteristics of pressure fluctuations at the flip buckets and stilling basins.

Soft computing methods are alternative tools compared with traditional regression approaches and analytical solutions for modeling complicated problems and systems [5]. Moreover, soft computing techniques are much more cost-effective and have less run time than experimental studies. It is noteworthy that soft computing models are newer tools than experimental studies. The capability of different soft computing approaches has been proven for modeling C_P' in the hydraulic jump [6–8]. These models were developed and assessed using experimental data. The suggested soft computing methods were more accurate than conventional regression schemes for predicting C_P' during the hydraulic jump. Moreover, Samadi et al. [9] modeled dynamic pressure distribution in flip bucket spillways using a variety of soft computing approaches and provided simple, high-precision mathematical expressions for estimating it.

According to the literature review's assessment, there were few investigations into the performance of soft computing techniques for estimating pressure fluctuations in hydraulic structures. In addition, the applications of ELM, GMDH, and M5MT methods have not yet been reported in estimating pressure fluctuations. However, the results mentioned above show that soft computing methods are a desirable alternative to model pressure fluctuations in hydraulic structures.

The ELM algorithm is an advanced method that is constructed on a single hidden layer feed-forward neural network (SFLN) [10]. The ELM improves training time and accuracy by transforming training data into fixed-length batches and only updating the weight without retraining the trained samples. The ELM approach has been successfully reported for longitudinal dispersion coefficients in water pipelines [11], compressive strength concrete estimation [12], and predicting total dissolved gas [13].

The group method of data handling is a set of induction techniques that can be used to make mathematical modeling of multi-parametric datasets. GMDH algorithm uses an inductive approach to sort and order more complex polynomial models, with an external criterion selecting the best result. The application of GMDH has been widely and successfully carried out in scour depth estimation and water-related engineering problems [14–16].

M5MT is a common decision tree method that uses piecewise multiple linear regression equations to approximate nonlinear functions. The M5MT was applied to predict drought events [17], scour depth [15, 18], alga growth in reservoirs [19], the capacity of shallow foundations [20], and groundwater modeling [21].

The major goal of this study is to investigate the usefulness and capability of ELM, GMDH, and M5MT for predicting C_P' is taking place during the hydraulic jump on sloping channels. The author's knowledge indicated that ELM, GMDH, and M5MT applications had not been investigated yet for prediction C_P'. Therefore, the current research in the first study evaluates the ability of ELM, GMDH, and M5MT to predict C_P' occurring in hydraulic jumps on sloping channels.

2. Materials and Methods

The data description and soft computing methods, including ELM, GMDH, and M5MT, are briefly explained in the following subsections.

2.1. Data Description. Data were collected from reliable and published experimental results of Gunal [22]. During the hydraulic jump phenomenon in the laboratory, pressure fluctuations occurred on the sloping channel. Gunal's [22] experiments were conducted in a 91 cm wide and 320 cm long rectangular flume. The inclined angles of the sloping channel were set to 10, 20, and 30 degrees. The cross-sectional representation of the hydraulic jump downstream of the sloping channel is displayed in Figure 1.

Guven [7] determined the relationship between $\sqrt{\overline{P'^2}}$ as root mean square value of the pressure fluctuations with characteristics of the flow and geometry of channels as follows:

$$\sqrt{\overline{P'^2}} = f\left(y_t, y_1, V_1, x, g, \rho_w, \theta\right), \tag{1}$$

where $\sqrt{\overline{P'^2}}$ is calculated from the following equation:

$$\sqrt{\overline{P'^2}} = \frac{1}{\sqrt{N}} \sqrt{\sum_{i=1}^{N} \left[P\left(x, y, n\Delta t\right) - \overline{P}\left(x, y\right)\right]^2}, \tag{2}$$

where $\overline{P}(x, y)$ is the mean pressure, $P(x, y, n\Delta t)$ is the value instantaneous of pressure, N is the number of data collected in a discrete-time series, and Δt is the sampling time step.

In (1), y_t is the depth of tail-water downstream of the stilling basin; y_1 is the gate opening of sluice gate; V_1 is the velocity issuing from the gate; x is the distance between the horizontal and inclined parts' intersection; g is the acceleration of gravity; ρ_w is water's mass density; and θ is the angle of the sloping section of the flume. Finally, the dimensional analysis provided the dimensionless functional form of (1) as follows [7]:

$$C_P' = f\left(Fr_1, \frac{x}{y_1}, \frac{y_t}{y_1}, \theta\right), \tag{3}$$

where $C_P' = \sqrt{\overline{P'^2}}/V_1^2/2g = RMS/V_1^2/2g$ and $Fr_1 = V_1/\sqrt{gy_1}$ are the dimensionless coefficient of pressure fluctuations and inflow Froude number, respectively. C_P' is an important factor used to describe pressure fluctuations.

2.2. Data Preparation for the Development of the Soft Computing Approaches. Overall, 112 values were obtained for C_P' under different flow conditions by recording pressure data. The statistical indices in the testing and training data sets should be similar; otherwise, we will be unable to test our models under certain hydrodynamic conditions that are

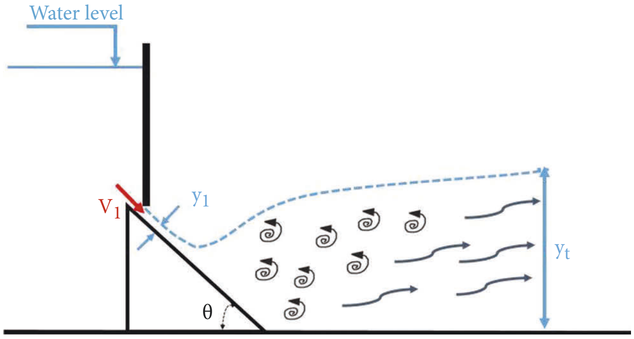


Figure 1: The cross-sectional representation of a hydraulic jump downstream of a sloping channel.

designated for the train set [23]. Therefore 80% of all data is randomly chosen for training, and the rest is used to assess the created models. The major statistical parameters of the testing and training data set are shown in Table 1. As seen in Table 1, the statistical parameters for testing and training subsets were close.

2.3. Extreme Learning Machine (ELM) and Model Development. Although the standard backpropagation neural network has several advantages, it also has some limitations, including tuning the parameter settings by selecting the number of hidden layers, momentum coefficient, and learning rate values. It suffers from the iterative learning process used to determine the weights, which takes an extended period. In addition, there is a possibility of a local minimum. By contrast, ELM provides it with the advantages of rapid convergence, fewer parameters to tweak, and a high degree of generalization [24].

The ELM comprises a single hidden layer that contains L nodes. The following equation is valid for N arbitrarily separate samples (x_i, t_i), with an activation function $g(x)$, randomly distributed weights (w_i), randomly distributed biases (b_i), and output weights (β_i).

$$\sum_{i=1}^{L} \beta_i g\left(w_i x_j + b_i\right) = O_j, j = 1, \ldots.N, \tag{4}$$

where $\beta = [\beta_1, \ldots\ldots, \beta_I]^T$ is a vector containing the output weights of the hidden layer of L nodes and the output node. It is possible to approximate N samples with zero error using typical SLFN with L hidden nodes and activation function $g(x)$. This means that

$$\sum_{j=1}^{N} \|Oj - tj\| = 0. \tag{5}$$

In other words, there are β_i, w_i, and b_i such that

$$\sum_{i=1}^{L} \beta_i g\left(w_i x_j + b_i\right) = tj, j = 1, \ldots.N. \tag{6}$$

Let

$$H = \begin{bmatrix} g(w_i x_i + b_1) & \cdots & g(w_L x_i + b_L) \\ \vdots & \ddots & \vdots \\ g(w_i x_N + b_1) & \cdots & g(w_L x_N + b_L) \end{bmatrix}, \quad \beta = \left[\beta_1^T \ldots \beta_L^T\right]^T. \tag{7}$$

And

$$T = \left[t_1^T \ldots t_L^T\right]^T. \tag{8}$$

Then,

$$\mathbf{H}\beta = T. \tag{9}$$

H is referred to as the neural network's hidden layer output matrix. Huang et al. [24] prove $L \leq N$ hidden neurons as necessary for infinitely differentiable activation functions. H remains unchanged once the biases and weights of hidden nodes are established, and only β needs to be estimated. The least-squares solution with a minimum norm for (9) is as follows:

$$\min\left\|H\left(w_1 w_L, b_1 b_L\right)\beta - T\right\|. \tag{10}$$

H is square if the neurons in the hidden layer equal the training set (N), and β may be calculated by inverting H. Nevertheless, to obtain higher generalization, the number of hidden nodes is modified, and it may be less than N. Then, the Moore–Penrose generalized inverse of H must be utilized in this case.

$$\widehat{\beta} = H^{\dagger} T, \tag{11}$$

where $H^{\dagger}$ can be used to validate the algorithm because it is the Moore–Penrose generalized inverse of H. The ELM model that has been constructed has the following simple general form for the prediction of C_P':

$$C_P' = \left[\frac{1}{\left(1 + \exp\ \left(In\ W \times In\ V + BHI\right)\right)}\right]^T \times \text{Out}\, W, \tag{12}$$

where InV is the input variables, InW is the input weight matrix, BHI is the hidden neuron vector's bias, and OutW is the output weight vector. The following relations were obtained for the ELM model to calculate C_P'.

Table 1: The main statistical parameters of training and testing data set for developing the proposed soft computing methods.

Dataset	Variables	Parameter	Min	Max	Avg	St. dev.
Training	Input	Fr_1	4.944	8.662	6.712	1.473
		x/y_1	0.750	31.875	11.993	7.761
		y_t/y_1	6.750	11.111	8.992	1.721
		θ	10	20	20.333	8.670
	Output	C_P'	0.027	0.096	0.056	0.017
Testing	Input	Fr_1	4.944	8.662	6.325	1.357
		x/y_1	0.750	28.333	12.550	8.731
		y_t/y_1	6.750	11.111	8.509	1.613
		θ	10	30	18.636	8.888
	Output	C_P'	0.028	0.090	0.057	0.018

The variables of Fr_1, x/y_1, y_t/y_1, and θ are considered as input variables of soft computing methods for the prediction of C_P'.

$$\mathrm{In}V = \begin{bmatrix} Fr_1 \\ \dfrac{y_t}{y_1} \\ \dfrac{x}{y_1} \\ \theta \end{bmatrix},$$

$$\mathrm{In}\,W = \begin{bmatrix} -0.41 & -0.17 & 0.88 & -0.61 \\ -0.56 & -0.98 & -0.92 & 0.02 \\ 0.96 & -0.24 & -0.65 & -0.94 \\ -0.81 & -0.91 & -0.66 & -0.14 \\ 0.44 & 0.68 & 0.55 & 0.16 \\ 0.61 & -0.88 & -0.93 & -0.86 \\ -0.31 & -0.65 & -0.02 & 0.36 \\ 0.92 & 0.98 & 0.49 & 0.74 \\ 0.57 & -0.32 & 0.53 & -0.90 \\ -0.07 & 0.06 & -0.39 & -0.86 \end{bmatrix},$$

$$\mathrm{BHI} = \begin{bmatrix} -0.68 \\ 0.29 \\ -0.99 \\ 0.55 \\ 0.30 \\ -0.84 \\ 0.72 \\ -0.24 \\ -0.97 \\ -0.96 \end{bmatrix},$$

$$\mathrm{Out}\,W = \begin{bmatrix} -0.05 \\ 0.03 \\ -0.01 \\ -0.09 \\ 0.02 \\ -0.02 \\ -0.06 \\ -0.01 \\ 0.04 \\ 0.01 \end{bmatrix}. \tag{13}$$

2.4. Group Method of Data Handling (GMDH) and Model Development.

The GMDH polynomial neural network introduced by Ivakhnenko [25] is a feed-forward neural network. This algorithm is a self-organizing system that is used to search progressively for the optimal solution to complicated nonlinear problems. This algorithm approximates the relationship between input and output based on quadratic polynomials. Therefore, GMDH creates new neurons in each layer by connecting pairs of neurons with quadratic polynomials. The goal of a mathematical model problem of GMDH is to discover a function $(\widehat{f})$ that can be used to approximate an original function (f) in order to estimate the model's output $(\widehat{y})$ for an assumed input vector X including n input variables [26]. For this, given n data instances of multi-input single-output data pairs, the following results are obtained:

$$y_i = f(X) = f(x_{i1}, x_{i2}, x_{i3}, \ldots, x_{in}); \ (i = 1, 2, \ldots, M). \tag{14}$$

A mathematical formulation describes the general equation between input-output variables. The goal now is to design a GMDH network in such a way that the square of the deviation output and the estimated output is as little as possible, which means:

$$\sum_{i=1}^{M}{}'\left[\hat{f}\left(x_{i1}, x_{i2}, x_{i3}, \ldots, x_{in}\right) - y_i\right]^2 \longrightarrow \min. \tag{15}$$

The mathematical expression between input and output variables can be described by a sophisticated discrete variant of the Volterra function called the Kolmogorov–Gabor polynomial. This series is presented in the following forms.

$$yw_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} w_{ijk} x_i x_j x_k + \cdots. \tag{16}$$

It is employed in this study to calculate the GMDH network's quadratic polynomial, which can be represented as [27]:

$$\hat{y} = G\left(x_i, x_j\right) = w_0 + w_1 x_i + w_2 x_j + w_3 x_i x_j + w_4 x_i^2 + w_5 x_j^2. \tag{17}$$

The mathematical expression (18) shows how neurons in a GMDH network are linked together to create the equation between input-output variables. Using the least-squares regression method, the weighting coefficient values of (17) are determined. This means that the deviation output, y, and the one that is calculated, $\hat{y}$, for each pair of x_i and x_j input variables are as minimal as possible. Thus, the weighting coefficients values of the quadratic function G_i are determined in order to optimize the fit of the output throughout the entire number of sample data pairs, that is [27].

$$E = \frac{\sum_{i=1}^{M}{}'\left(y_i - G_i\right)^2}{M} \longrightarrow \min. \tag{18}$$

In its ordinary form, the GMDH method considers all possible combinations of two input variables from a total of n input variables in order to generate the regression polynomial in the form of (17) that fits the dependent data ($\mathrm{y_i}$, $\mathrm{i} = 1, 2, \ldots, \mathrm{M}$) the best in a least-squares scheme. As a consequence, the initial layer of the GMDH network's architecture will be selected with $C_n^2 = n(n-1)/2$ input neurons for the creation of the quadratic polynomial based on observations $\left\{\left(y_i, x_{ip}, x_{iq}\right); \left(i = 1, 2, \ldots, M\right)\right\}$ for varied $p, q \in \{1, 2, \ldots, n\}$. To put it another way, it is currently possible to generate M data triples $\left\{\left(y_i, x_{ip}, x_{iq}\right); \left(i = 1, 2, \ldots, M\right)\right\}$ from observations by utilizing such $p, q \in \{1, 2, \ldots, n\}$ in the following way:

$$\begin{bmatrix} x_{1p} & x_{1q} & y_1 \\ x_{2p} & x_{2q} & y_2 \\ x_{mp} & x_{mq} & y_m \end{bmatrix}. \tag{19}$$

By replacing the quadratic sub-expression in the shape of (17) for each row of M data triples, the matrix equation shown below is simple to construct.

$$AW = Y, \tag{20}$$

where W and Y are the vectors of unknown quadratic polynomial weighting coefficients in (17) and a vector containing the observed values of outputs, respectively.

$$\begin{aligned} W &= \left\{w_0, w_1, w_2, w_3, w_4, w_5\right\}^T, \\ Y &= \left\{y_1, y_2, y_3, \ldots, y_M\right\}^T. \end{aligned} \tag{21}$$

The superscript T denotes the matrix's transposition. It is self-evident that:

$$\mathrm{A} = \begin{bmatrix} 1 & x_{1p} & x_{1q} & x_{1p}x_{1q} & x_{1p}^2 & x_{1q}^2 \\ 1 & x_{2p} & x_{2q} & x_{2p}x_{2q} & x_{2p}^2 & x_{2q}^2 \\ 1 & x_{mp} & x_{mq} & x_{mp}x_{mq} & x_{mp}^2 & x_{mq}^2 \end{bmatrix}. \tag{22}$$

To solve regression analysis problems, we employ the least-squares approach.

$$W = \left(A^T A\right)^{-1} A^T Y. \tag{23}$$

This returns the vector with the best quadratic (17) weighting coefficients for the entire set of M data triples. It is worth noting that this technique is replicated for each neuron in the subsequent hidden layer, depending on the network's connectivity structure. Using the GMDH algorithm, the following GMDH network was obtained for the prediction C_P' (see Figure 2).

With respect to Figure 2, the following equations were obtained for the prediction C_P' using the GMDH approach.

The first layer output:

$$\begin{aligned} \left(C_P'^{*}\right)_1^1 &= 0.1101 - 0.0110\left(\frac{y_t}{y_1}\right) + 3.6904(\theta) + 6.2904\times 10^{-4}\left(\frac{y_t}{y_1}\right)^2 + 8.3932\times 10^{-5}(\theta)^2 - 3.0321\times 10^{-4}\left(\frac{y_t}{y_1}\right)(\theta), \\ \left(C_P'^{*}\right)_3^1 &= 0.1253 - 0.0180\left(Fr_1\right) - 5.5504\times 10^{-4}(\theta) + 0.0013\left(Fr_1\right)^2 + 9.1284\times 10^{-5}(\theta)^2 - 3.02542\times 10^{-4}\left(Fr_1\right)(\theta), \\ \left(C_P'^{*}\right)_4^1 &= 0.0979 - 0.0079\left(Fr_1\right) + 0.0014\left(\frac{x}{y_1}\right) + 8.1446\times 10^{-5}\left(Fr_1\right)^2 - 6.6223\times 10^{-5}\left(\frac{x}{y_1}\right)^2 + 5.4575\times 10^{-5}\left(Fr_1\right)\left(\frac{x}{y_1}\right), \\ \left(C_P'^{*}\right)_6^1 &= 0.0109 + 0.01502\left(\frac{y_t}{y_1}\right) + 0.0011\left(\frac{x}{y_1}\right) - 0.0011\left(\frac{y_t}{y_1}\right)^2 - 6.2790\times 10^{-5}\left(\frac{x}{y_1}\right)^2 + 5.8831\times 10^{-5}\left(\frac{y_t}{y_1}\right)\left(\frac{x}{y_1}\right). \end{aligned} \tag{24}$$

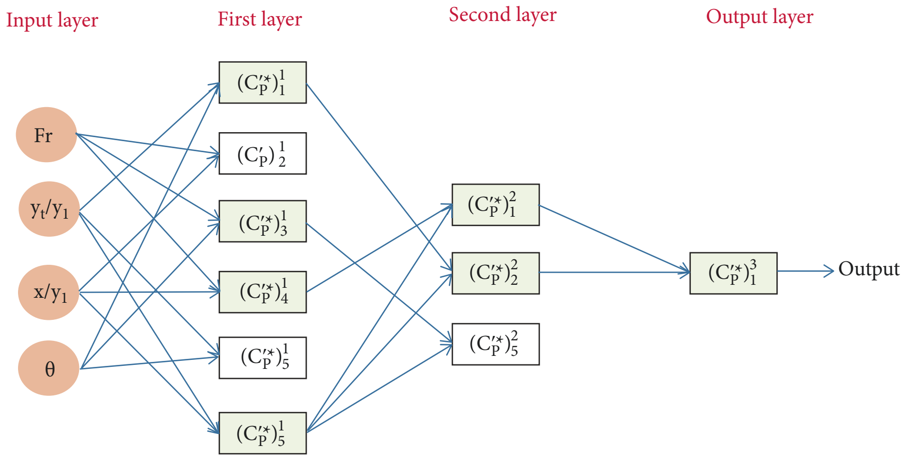


FIGURE 2: The GMDH network for prediction C_P'.

The second layer output:

$$\begin{aligned}(C_P^{'*})_1^2 &= 0.1382 - 2.6925(C_P^{'*})_4^1 - 1.7543(C_P^{'*})_6^1 - 1.3848\times 10^3\left((C_P^{'*})_4^1\right)^2 - 1.3189\times 10^3\left((C_P^{'*})_6^1\right)^2 + 2.7568\times 10^3\left((C_P^{'*})_4^1\right)\left((C_P^{'*})_6^1\right),\\ &(C_P^{'*})_2^2 0.1671 - 0.7202(C_P^{'*})_1^1 - 4.8455(C_P^{'*})_6^1 + 31.2869\left((C_P^{'*})_1^1\right)^2 + 70.9947\left((C_P^{'*})_6^1\right)^2 - 41.3749\left((C_P^{'*})_1^1\right)\left((C_P^{'*})_6^1\right),\\ &(C_P^{'*})_3^2 0.1845 - 1.1091(C_P^{'*})_3^1 - 5.1238(C_P^{'*})_6^1 + 36.6985\left((C_P^{'*})_3^1\right)^2 + 75.5852\left((C_P^{'*})_6^1\right)^2 - 45.3275\left((C_P^{'*})_3^1\right)\left((C_P^{'*})_6^1\right).\end{aligned}\tag{25}$$

The output layer:

$$(C_P^{'*})_1^3 0.0337 - 0.4779(C_P^{'*})_1^2 + 0.1849(C_P^{'*})_2^2 + 6.7083\left((C_P^{'*})_1^2\right)^2 + 5.1880\left((C_P^{'*})_2^2\right)^2 - 0.2282\left((C_P^{'*})_1^2\right)\left((C_P^{'*})_2^2\right).\tag{26}$$

Finally, the value of C_P' is calculated from the above formulations of GMDH. As seen with replacing the input variables with the GMDH formulation, the $(C_P^{'*})_1^3$ approximate the outcome GMDH network for prediction of C_P'.

2.5. M5 Model Tree (M5MT) and Model Development. M5MT is one of the most widely used decision tree techniques in data-driven modeling. In the M5MT, the entire input domain is recursively partitioned into subdomains, with each subdomain being predicted using a multiple linear regression model. The graphical M5MT is constructed from a root node, the number of binary branches, a group of inner nodes (splits), and a number of terminal nodes (leaves) [28]. For this reason, the resulting tree model has a clear decision structure and is understandable for everyone.

The constructing, pruning, and smoothing of the tree are the three main components of the M5MT algorithm. The splitting criterion is used to construct the primary tree. The expected reduction in error resulting from evaluating each attribute at the node is calculated using this splitting criterion. A measure of the error at a node is defined as the standard deviation of the class values that reach the node. After that, the attribute with the highest anticipated error reduction is chosen. The following formula is used to get the standard deviation reduction (SDR) for M5MT:

$$\mathrm{SDR} = s\,d(T) - \sum_i \frac{|T_i|}{|T|} \times s\,d(T_i),\tag{27}$$

where T denotes the set of instances that reach the node, T_i is the result of separating the node according to the attribute chosen, and s d denotes the standard deviation [29]. To stop splitting, either only a few examples remain, or their class values are less than 5% of the initial instance set's standard deviation. It is possible to encounter an over-fitting problem based on training data during the creation tree process. Pruning is a technique that has been employed in trying to alleviate this difficulty in the past. It decreases the size of the model tree by deleting splits that don't meaningfully enhance prediction.

For the test data, the pruning algorithm uses an estimate of the predicted error at each node. To begin, the absolute deviation between the observed and estimated output values for each of the training cases entering the node is averaged. Due to the fact that the trees were constructed specifically for this dataset, the average will underrate the predicted error for new instances. This is compensated for by multiplying the output value by the factor $(n + v)/(n - v)$. where n denotes the size of the training instances received at the node [30]. In addition, v is the number of model attributes that signify the output value at that node. The leaf node can be omitted if the estimated error is lower at the parent. As a result, this multiplication is done to make sure that new data, rather than training data, don't get underestimated by the multiplication.

Quinlan [31] describes a smoothing approach that employs the leaf model to determine the estimated value during the smoothing phase. The value is then smoothed by merging it with the linear model's estimated value for each node along the path back to the root. This requires the following calculation:

$$P' = \frac{np + kq}{n + k}, \tag{28}$$

P' is a prediction that exceeds the upper node; p is a prediction from below that is passed to the current node; q is the model's predicted value at the node; n represents the number of training instances that have made it to the preceding node, and Wang and Witten [28] constant is denoted by k. Using the M5MT created, the graphical tree, is shown in Figure 3.

The decision rules concerned with Figure 3 are as follows:

$$LM\,(4) = C_P' = -0.0041 \times Fr_1 + 0.0792. \tag{29}$$

As seen, four linear rules were obtained for the prediction C_P'. Concerning the values of Fr_1, the appropriate rule was selected then C_P' was calculated.

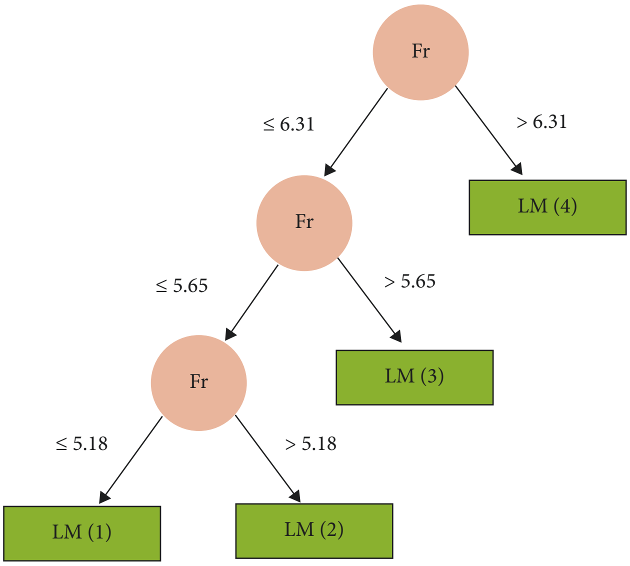


FIGURE 3: M5MT created the regression tree to predict C_P'.

3. Results and Discussion

Statistical indices such as correlation coefficient (CC), root mean square error (RMSE), mean absolute error (MAE), index of agreement (Ia), scatter index (SI), and BIAS index was employed to assess the qualitative evaluation of the developed suggested models.

$$\begin{aligned} \mathrm{CC} &= \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}{}'(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}{}'(y_i - \overline{y})^2}}, \\ \mathrm{RMSE} &= \sqrt{\frac{\sum_{i=1}^{N}{}'(x_i - y_i)^2}{N}}, \\ \mathrm{MAE} &= \frac{1}{N}\sum_{i=1}^{N} |x_i - y_i|, \\ Ia &= 1 - \frac{\sum_{i=1}^{n}{}'(x_i - y_i)^2}{\sum_{i=1}^{n}{}'(|x_i - \overline{x}| + |y_i - \overline{y}|)^2}, \\ SI &= \frac{RMSE}{\overline{x}} \times 100, BIAS = \overline{y} - \overline{x}, \end{aligned} \tag{30}$$

where x_i and y_i denote measured and predicted values, $\overline{x}$ and $\overline{y}$ represent the average of measured and predicted values, and N denotes the number of the dataset. In Table 2, the statistical index values for the proposed soft computing methods in the training and testing stages for the prediction of C_P'.

Table 2 shows that the ELM model performed C_P' prediction with the lowest errors (RMSE, MAE, and SI) and higher coefficient correlation (CC) and index agreement (Ia) than the GMDH and M5MT models in both the training and testing stages. As observed in Table 2, the ELM method with CC = 0.9183, RMSE = 0.0067, and MAE = 0.0051 has the best prediction accuracy compared with GMDH and M5MT. On the other hand, it can be deduced from Table 2 that GMDH (CC = 0.8818, RMSE = 0.0078, and MAE = 0.0058) and M5MT (CC = 0.6883, RMSE = 0.0120, and MAE = 0.0090) have the second and third level of accuracy for the prediction C_P' in the testing stage. Furthermore, the scatter index (SI = 11.88%) for the ELM method is smaller than the SI for GMDH (SI = 13.89%) and M5MT (SI = 21.28%). It is noteworthy that the BIAS values of ELM for training and testing data are 0.0008 and 0.0020, respectively, indicating a slightly overestimated C_P'. However, the BIAS values of GMDH in training and testing were −0.0005 and 0.0016, and for M5MT, they were −0.0003 and 0.0014 for the training and testing stages, respectively. Therefore, it can be concluded that, overall, ELM was more conservative for prediction C_P' than GMDH and M5MT.

By comparing two soft computing approaches, including the ELM (as the best model) and the M5MT (as the worst model) in the testing stage, it can be shown that the ELM produced significantly lower errors in prediction C_P' than the M5MT. The values of RMSE and MAE obtained via ELM showed that they decreased by about 79.10% and 76.47% compared with the M5MT in the testing stage, respectively.

Besides, the CC and Ia values for ELM increased by 33.42% and 21.05% compared with M5MT. The error values

Table 2: The statistical indices for ELM, GMDH, and M5MT for the prediction of C_P'.

Approach	Data set	CC	RMSE	MAE	Ia	SI (%)	BIAS
ELM	Training	0.9770	0.0046	0.0040	0.9841	8.00	0.0008
ELM	Testing	0.9183	0.0067	0.0051	0.9569	11.88	0.0020
GMDH	Training	0.9135	0.0074	0.0066	0.9512	12.85	−0.0005
GMDH	Testing	0.8818	0.0078	0.0058	0.9361	13.89	0.0016
M5MT	Training	0.8309	0.0108	0.0090	0.8467	18.81	−0.0003
M5MT	Testing	0.6883	0.0120	0.0090	0.7905	21.28	0.0014

of RMSE and MAE from GMDH in the testing stage showed that these values improved by about 53.85% and 55.17% compared with those obtained from M5MT, respectively. Moreover, the CC and Ia values of GMDH increased by about 28.11% and 18.42% compared with M5MT. Based on error measures, it is clear that the ELM model was better than the GMDH and M5MT in terms of accuracy.

The mathematical expressions for the prediction C_P' were formulated from the proposed soft computing models. It seems the M5MT result was more transparent and easier to use than ELM and GMDH. M5MT provided four multivariate linear models for the prediction of C_P'. Although the derived rules from M5 had less accuracy than ELM and GMDH, these rules are more straightforward for predicting C_P'. On the other hand, GMDH presented complex mathematical formulations for the prediction of C_P'. The GMDH used two hidden layers of neurons for the prediction of C_P'. The GMDH utilized the quadratic polynomial of input variables and the combination of the best neurons in each layer to predict C_P'. ELM generated the coefficients matrixes for the prediction of C_P'. It seems the ELM mathematical shapes were simple compared with the GMDH equations.

M5MT method provided four simple equations for the prediction of C_P' based on dividing the domain of the problem. It is worth mentioning that M5MT selected the appropriate rule concerning only one input variable (i.e., Froude number) and used two variables, including Fr_1 and $\mathrm{x/y}_1$ in their rules. However, the ELM and GMDH used all the independent variables to generate the predictive expressions for estimating C_P' values. Although M5MT has led to the creation of simple rules for predicting C_P', these simple formulas cause M5MT to be less accurate than ELM and GMDH expressions. M5MT divided the domain of input variables into four subdomains and represented four regression equations to estimate C_P'. In fact, separating the domain of the problem into the local subdomains and combining their results caused an improvement in the accuracy compared with the single equation.

Figures 4–9 compare observed values of C_P' versus estimated values obtained from ELM, GMDH, and M5MT during the training and testing stages. As observed in scatter plots, the predicted C_P' values from ELM were more concentrated around the ideal line (the 45-degree line).

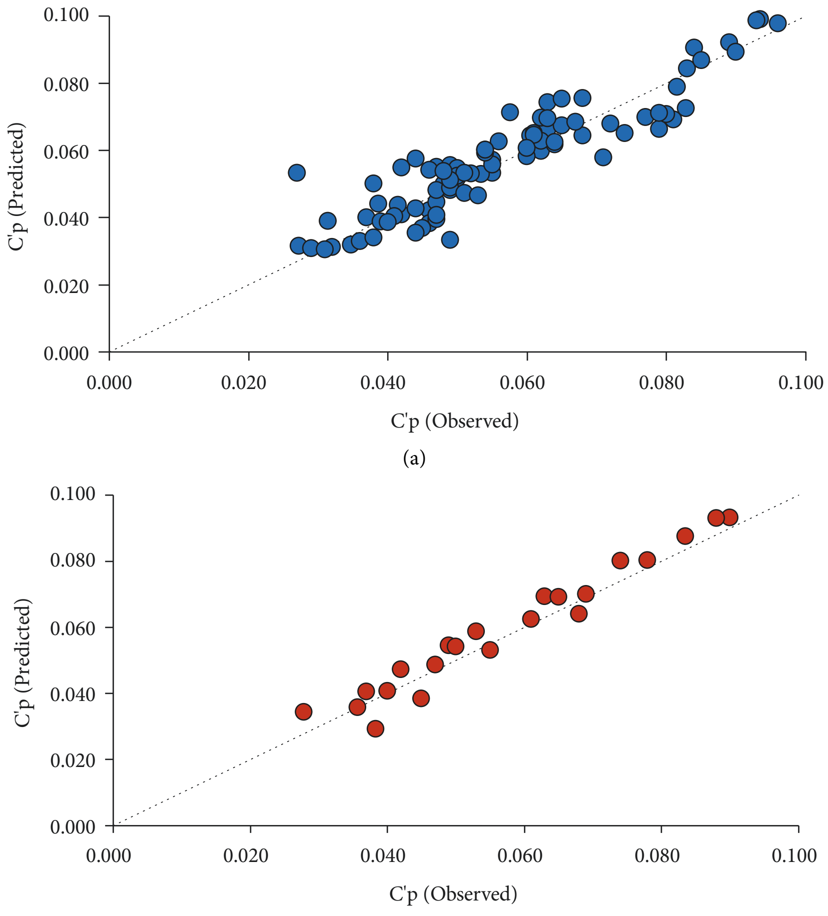
In addition, the variation of C_P' values obtained by soft computing methods versus observed C_P' demonstrated the capability of estimating C_P' by the proposed approaches. These figures graphically confirmed the higher ELM accuracy than GMDH and M5MT.

These figures indicated that ELM performed better in the training and testing stages than GMDH and M5MT. As shown, the ELM reasonably estimated C_P' in the training and testing stages (bias = 0.0008 and 0.0020). The remarkable point is that in Figures 4 and 5, the peak values of C_P' by ELM were estimated well. In contrast, GMDH and M5MT slightly underestimated the values of C_P', especially at the peak C_P' values. From the comparison between Figures 4–9, it can be deduced that ELM is more skillful and accurate than GMDH and M5MT in the prediction of C_P'.

The present study results were compared with earlier research conducted by Samadi et al. [8]. They used three data-driven algorithms, such as MARS, GEP, and CART, for the prediction of C_P'. Samadi et al. [8] indicated that CART results for the prediction of C_P' have three and four nonterminal and terminal nodes, respectively. The CART tree structure used only the Fr_1 variable with threshold values of 6.31, 5.18, and 5.65 for the CART tree structure. The present study used another common decision tree technique, namely M5MT, to predict C_P'.

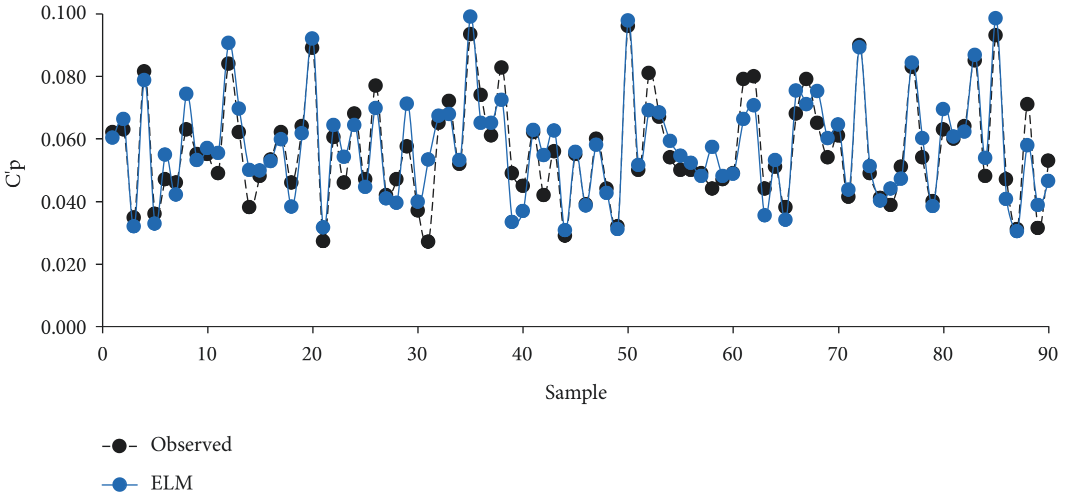
It is worth noting that M5MT and CART are two common decision tree algorithms used for regression problems. In the present study, M5MT, among the four input variables included, Fr_1, x/y_1, y_t/y_1, and θ, only selected the Fr_1 variable for prediction of C_P'. As illustrated in Figure 3, M5MT's tree structure uses three nonterminal and four terminal nodes. In addition, the splitting values for M5MT were 6.31, 5.18, and 5.65. A comparison of the tree structures of the decision tree (i.e., M5MT) and the proposed CART presented by Samadi et al. [8] revealed that the two models have similar structures. They employed a splitting variable (Fr_1) and the same splitting values for the generation of regression trees. As a result, the graphical structures of these two decision tree algorithms for selecting the independent variable (Fr_1) and creating four if-then rules are similar. However, M5MT provided linear regression functions while CART presented constant values in their terminal nodes. This was the main distinctive characteristic of the differences between M5MT and CART concerning the nature of CART and M5MT. It should be noted that due to the nature of the M5MT algorithm, the Fr_1 parameter was an attribute that caused error reduction for the prediction of C_P' in the regression tree obtained by M5MT. In addition, the present study's findings about selecting the Fr_1 parameter is completely consistent with the results of the CART tree provided by Samadi et al. [8].

Also, the statistical measures indicated the results of M5MT and CART were more or less the same. M5MT with RMSE = 0.0120 and MAE = 0.0090, compared with CART



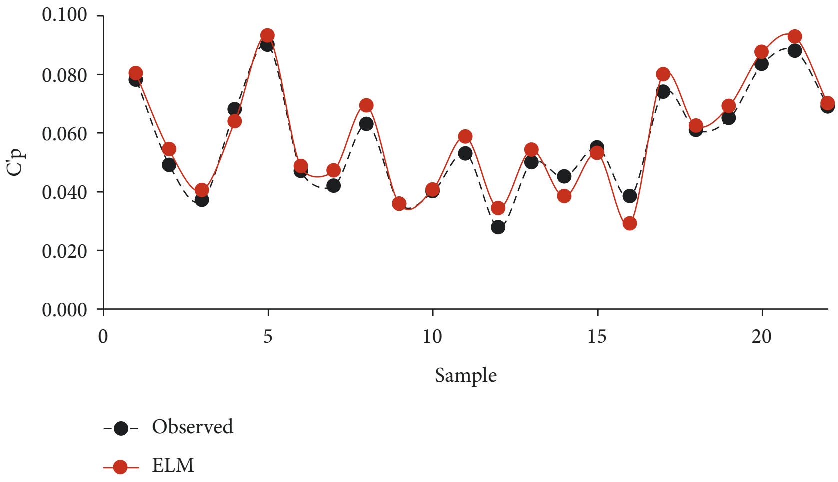
(b)

FIGURE 4: Scatter plots of measured and estimated values of C_P' for ELM during (a) training stage and (b) testing stage.



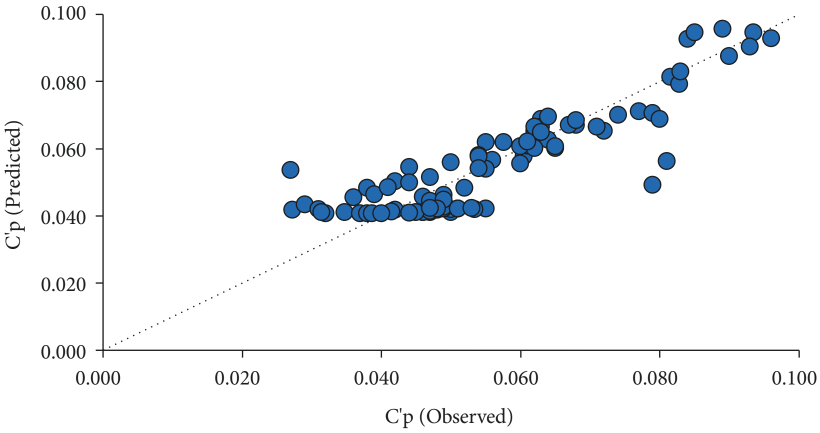
(a)

FIGURE 5: Continued.

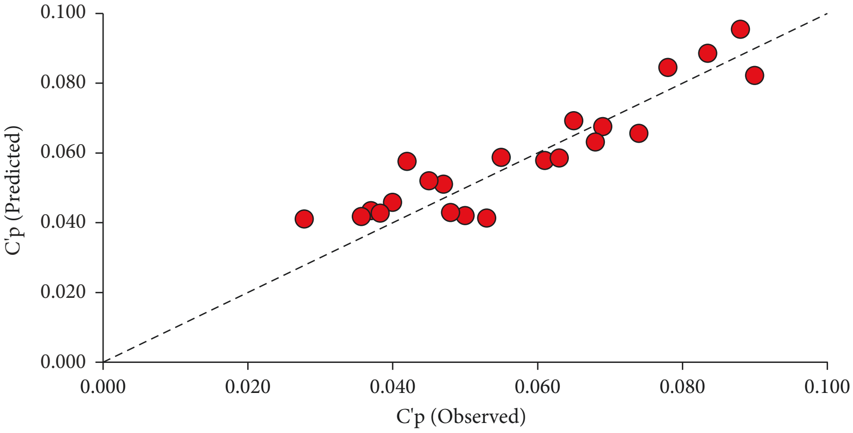


(b)

Figure 5: Variation of C_P' with ELM during the training stage (a) and testing stage (b).

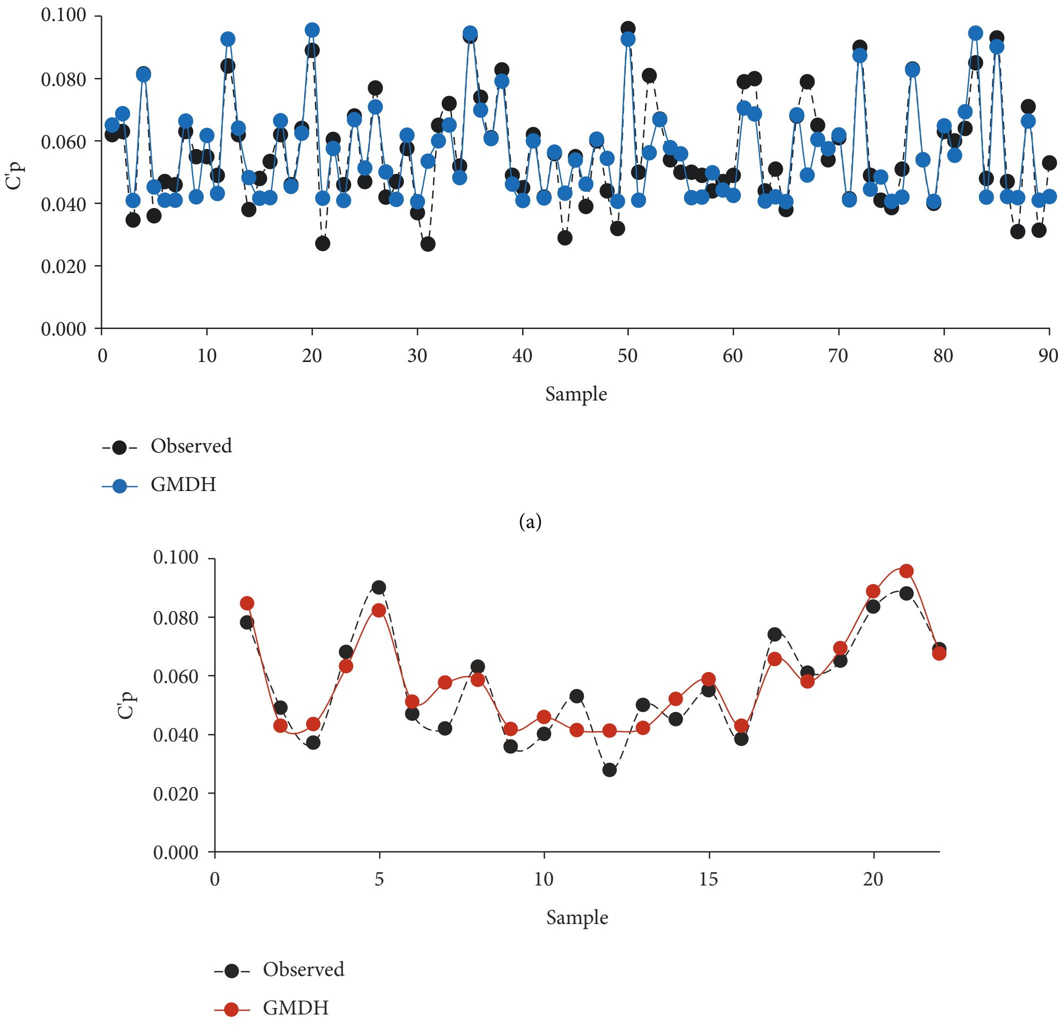


(a)



(b)

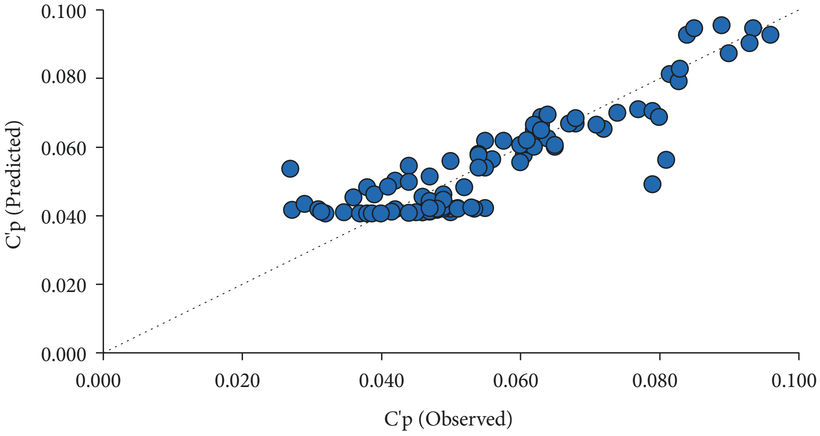
Figure 6: Scatter plots of measured and estimated values of C_P' for GMDH during (a) training stage and (b) testing stage.



(a)

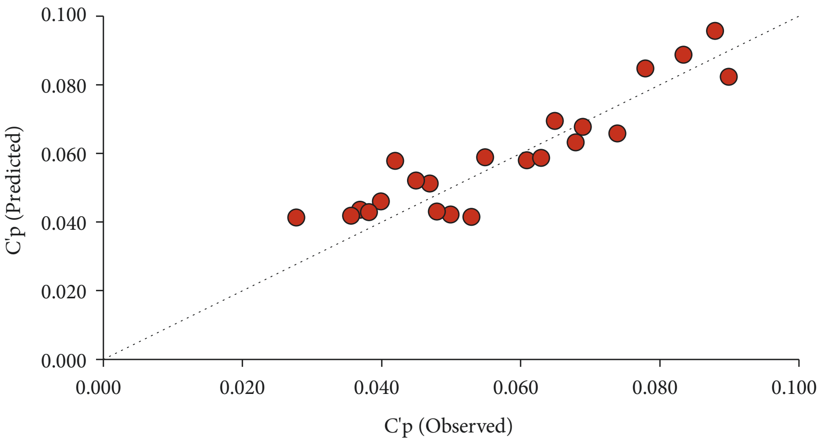
(b)

Figure 7: Variation of C_P' with GMDH during the training stage (a) and testing stage (b).



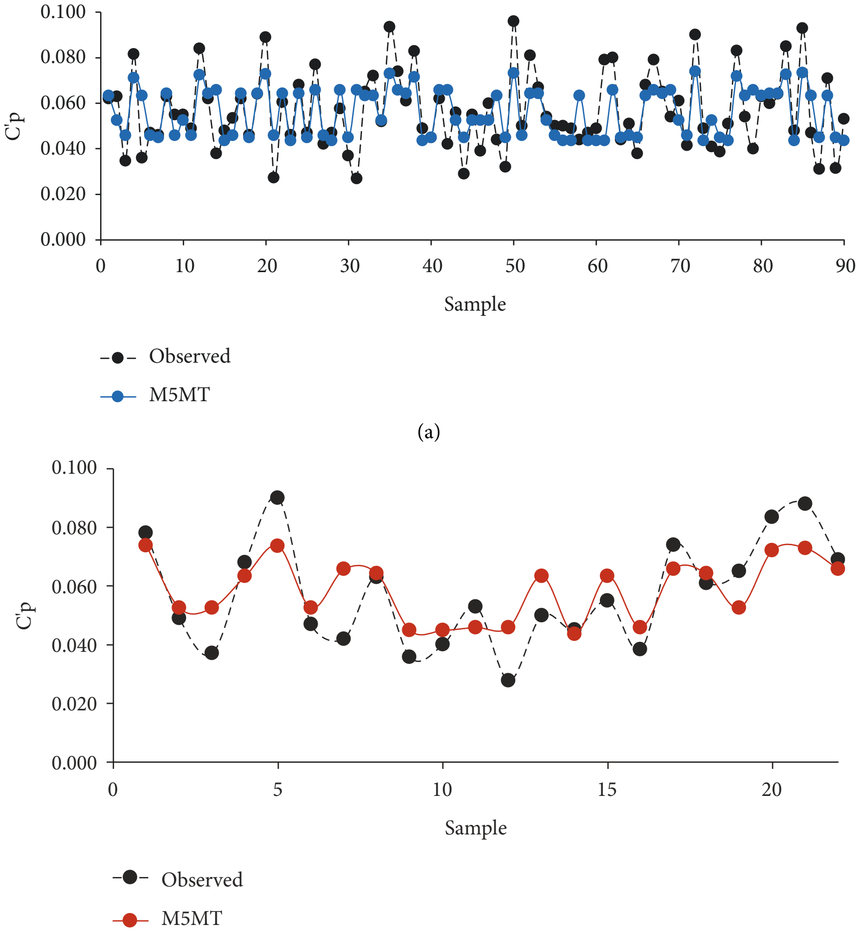
(a)

Figure 8: Continued.



(b)

Figure 8: Scatter plots of measured and estimated values of C_P' for M5MT during (a) training stage and (b) testing stage.



(a)

(b)

Figure 9: Variation of C_P' with M5MT during the training stage (a) and testing stage (b).

[12] with RMSE = 0.012, MAE = 0.009, was similar results. However, M5 generated multivariate linear regression equations while CART generated the constant values for the prediction of C_P'. The linear equations by M5MT caused the flexibility and generalizability of M5 to improve the prediction of C_P' compared with the constant values of C_P' that was yielded by CART. Compared with GMDH in the present study with GEP [12], both algorithms provided nonlinear mathematical expressions for the prediction of C_P'. The accuracy of GMDH with RMSE = 0.0078 and MAE = 0.0058 is slightly better than GEP results with RMSE = 0.008 and MAE = 0.006. Moreover, compared with the formulas provided by GEP and GMDH, GEP requires more difficult calculations related to trigonometric and algebraic functional sets appearing in the computing process, including exp, ln, sin, and atan functions. In contrast, GMDH used polynomial quadratic equations. From this perspective as well, it seems the computational effort was less than GEP. Finally, the comparison results of ELM with MARS indicated that both algorithms had almost identical results.

4. Summary and Conclusions

Stilling basins are used widely as hydraulic dissipation structures in large dams. An accurate estimation of C_P' within stilling basins is a critical issue for hydraulic engineering for the design of stilling basins. This study employed three soft computing methods, including ELM, GMDH, and M5MT, to estimate the C_P' that occurred during the hydraulic jump in the sloping channels.

Different soft computing models were developed to estimate C_P' according to the dimensionless parameters and experimental data. ELM showed the lowest error values of RMSE, MAE, and SI in the training and testing stages. In addition, ELM has the highest correlation coefficient and Ia values than those obtained from GMDH and ELM. Therefore, the proposed soft computing models provided sufficiently accurate results due to this problem's complexity.

ELM formulated the matrices of coefficients for the prediction of C_P'. In addition, GMDH provided mathematical quadratic polynomials and combined input variables for the prediction of C_P'. M5MT generated four simple rules for the estimation of C_P'. It seems the application of M5MT was the most straightforward method for the prediction of C_P'. It is noteworthy that complexity degree equations were derived from ELM and GMDH for the prediction C_P' have been more than M5MT rules.

In summary, the ELM method provided a weight matrix for predicting C_P'. GMDH generated second-order polynomial equations for the prediction of C_P'. M5MT developed piecewise multiple linear regression equations for the calculation of C_P'. GMDH and M5MT methods generated explicit and clear mathematical expressions to estimate C_P'. In addition, M5MT divided the problem domain into subdomains and fitted local linear models to compute C_P'. It seems that in terms of the degree of complexity of the developed models in estimating C_P', the ELM model has the highest complexity, followed by GMDH and M5MT models. However, in contrast to the complexity degree of the proposed models, their computational accuracy has increased for the estimation of C_P' so that the ELM model has the highest accuracy, followed by GMDH and M5MT.

Finally, the comparisons of the results with previous research revealed that the proposed applications of soft computing methods have good performance for prediction C_P'.

Further works can be considered to use pressure field data for modeling with soft computing methods. It is also recommended that hybrid data-driven models with evolutionary algorithms be used instead of stand-alone data-driven models to figure out the coefficient of pressure fluctuations.

Data Availability

The data references are described in the text of the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] W. H. Hager, "Energy Dissipators and Hydraulic Jump," *Kluwer Academic Publ., Water Science and Technology Library*, vol. 8, 1992.

[2] H. Chanson (ED.), *Energy Dissipation in Hydraulic Structures*, CRC Press, FL, USA pp. 178, 2015, ISBN: 9780367575731.

[3] V. Fiorotto and A. Rinaldo, "Turbulent pressure fluctuations under hydraulic jumps," *Journal of Hydraulic Research*, vol. 30, no. 4, pp. 499–520, 1992a.

[4] V. Fiorotto and A. Rinaldo, "Fluctuating uplift and lining design in spillway stilling basins," *Journal of Hydraulic Engineering*, vol. 118, no. 4, pp. 578–596, 1992b.

[5] A. H. Zaji, H. Bonakdari, S. R. Khodashenas, and S. Shamshirband, "Firefly optimization algorithm effect on support vector regression prediction improvement of a modified labyrinth side weir's discharge coefficient," *Applied Mathematics and Computation*, vol. 274, pp. 14–19, 2016.

[6] A. Guven, M. Günal, and A. Çevik, "Prediction of pressure fluctuations on sloping stilling basins," *Canadian Journal of Civil Engineering*, vol. 33, no. 11, pp. 1379–1388, 2006.

[7] A. Guven, "A predictive model for pressure fluctuations on sloping channels using support vector machine," *International Journal for Numerical Methods in Fluids*, vol. 66, no. 11, pp. 1371–1382, 2011.

[8] M. Samadi, H. Sarkardeh, and E. Jabbari, "Explicit data-driven models for prediction of pressure fluctuations occur during turbulent flows on sloping channels," *Stochastic Environmental Research and Risk Assessment*, vol. 34, no. 5, pp. 691–707, 2020.

[9] M. Samadi, H. Sarkardeh, and E. Jabbari, "Prediction of the dynamic pressure distribution in hydraulic structures using soft computing methods," *Soft Computing*, vol. 25, no. 5, pp. 3873–3888, 2021.

[10] Q. Chen, C. Wang, and L. Song, "Prediction of low-temperature rheological properties of SBS modified asphalt," *Advances in Civil Engineering*, vol. 2020, Article ID 8864766, 8 pages, 2020.

[11] F. Saberi-Movahed, M. Najafzadeh, and A. Mehrpooya, "Receiving more accurate predictions for longitudinal dispersion coefficients in water pipelines: training group method

of data handling using extreme learning machine conceptions," *Water Resources Management*, vol. 34, no. 2, pp. 529–561, 2020.

[12] M. M. Hameed, M. A. Abed, N. Al-Ansari, and M. K. Alomar, "Predicting compressive strength of concrete containing industrial waste materials: novel and hybrid machine learning model," *Advances in Civil Engineering*, vol. 2022, Article ID 5586737, 19 pages, 2022.

[13] M. K. AlOmar, M. M. Hameed, N. Al-Ansari, and M. A. AlSaadi, "Data-driven model for the prediction of total dissolved gas: robust artificial intelligence approach," *Advances in Civil Engineering*, vol. 2020, Article ID 6618842, 20 pages, 2020.

[14] M. Najafzadeh, G. A. Barani, and H. M. Azamathulla, "GMDH to predict scour depth around a pier in cohesive soils," *Applied Ocean Research*, vol. 40, pp. 35–41, 2013.

[15] M. Najafzadeh and G. Oliveto, "More reliable predictions of clear-water scour depth at pile groups by robust artificial intelligence techniques while preserving physical consistency," *Soft Computing*, vol. 25, no. 7, pp. 5723–5746, 2021.

[16] M. Najafzadeh, M. Rezaie-Balf, and A. Tafarojnoruz, "Prediction of riprap stone size under overtopping flow using data-driven models," *International Journal of River Basin Management*, vol. 16, no. 4, pp. 505–512, 2018.

[17] A. Barzkar, M. Najafzadeh, and F. Homaei, "Evaluation of drought events in various climatic conditions using data-driven models and a reliability-based probabilistic model," *Natural Hazards*, vol. 110, no. 3, pp. 1931–1952, 2022.

[18] M. Samadi, E. Jabbari, and H. M. Azamathulla, "Assessment of M5$'$ model tree and classification and regression trees for prediction of scour depth below free overfall spillways," *Neural Computing & Applications*, vol. 24, no. 2, pp. 357–366, 2014.

[19] N. C. Jung, I. Popescu, P. Kelderman, D. P. Solomatine, and R. K. Price, "Application of model trees and other machine learning techniques for algal growth prediction in Yongdam reservoir, Republic of Korea," *Journal of Hydroinformatics*, vol. 12, no. 3, pp. 262–274, 2010.

[20] R. Khorrami, A. Derakhshani, and H. Moayedi, "New explicit formulation for ultimate bearing capacity of shallow foundations on granular soil using M5'model tree," *Measurement*, vol. 163, Article ID 108032, 2020.

[21] M. Najafzadeh, F. Homaei, and S. Mohamadi, "Reliability evaluation of groundwater quality index using data-driven models," *Environmental Science and Pollution Research*, vol. 29, no. 6, pp. 8174–8190, 2022.

[22] M. Gunal, *Numerical and Experimental Investigation of Hydraulic Jumps*, Ph.D. Thesis, University of Manchester Institute of Science and Technology, Manchester, 1996.

[23] B. Bhattacharya, R. K. Price, and D. P. Solomatine, "Machine learning approach to modeling sediment transport," *Journal of Hydraulic Engineering*, vol. 133, no. 4, pp. 440–450, 2007.

[24] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[25] A. Ivakhnenko, "The group method of data handling in long-range forecasting," *Technological Forecasting and Social Change*, vol. 12, no. 2-3, pp. 213–227, 1978.

[26] M. Najafzadeh and F. Saberi-Movahed, "GMDH-GEP to predict free span expansion rates below pipelines under waves," *Marine Georesources & Geotechnology*, vol. 37, no. 3, pp. 375–392, 2019.

[27] M. Najafzadeh, G. A. Barani, and M. R. Hessami-Kermani, "Evaluation of GMDH networks for prediction of local scour depth at bridge abutments in coarse sediments with thinly armored beds," *Ocean Engineering*, vol. 104, pp. 387–396, 2015.

[28] Y. Wang and I. H. Witten, "Induction of model trees for predicting continuous lasses," University of Economics, Faculty of Informatics and Statistics, Prague, 1997.

[29] N. C. Jung, "Eco-hydraulic modelling of eutrophication for reservoir management," CRC Press, FL, USA, 2010, ISBN: 9780415573825.

[30] I. H. Witten and E. Frank, *Data Mining-Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publisher, San Francisco, CA, 2005.

[31] J. R. Quinlan, *Learning with continuous classes*, A. Adams and L. Sterling, Eds., pp. 343–348, World Scientific, Singapore, 1992.