Pipeline Scour Rates Prediction-Based Model Utilizing a Multilayer Perceptron-Colliding Body Algorithm

Mohammad Ehteram¹, Ali Najah Ahmed², Lloyd Ling³,⁴, Chow Ming Fai⁵, Sarmad Dashti Latif⁶,⁷, Haitham Abdulmohsin Afan⁶,⁷, Fatemeh Barzegari Banadkooki⁷ and Ahmed El-Shafie⁸,⁹

¹Department of Water Engineering and Hydraulic Structures, Faculty of Civil Engineering, Semnan University, Semnan, Iran; mohammdehteram@semnan.ac.ir
²Institute of Energy Infrastructure (IEI), Universiti Tenaga Nasional (UNITEN), Selangor 43000, Malaysia; Mahfoodh@uniten.edu.my
³Department of Civil Engineering, Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Kajang 43000, Selangor, Malaysia
⁴Institute of Sustainable Energy (ISE), Universiti Tenaga Nasional (UNITEN), Selangor 43000, Malaysia; Chowmf@uniten.edu.my
⁵Department of Civil Engineering, College of Engineering, Universiti Tenaga Nasional (UNITEN), Selangor 43000, Malaysia; Sarmad.latif@uniten.edu.my
⁶Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam
⁷Agricultural Department, Payam Noor University, Tehran, Iran; barzegari@pnu.ac.ir
⁸Department of Civil Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia; elshafie@um.edu.my
⁹National Water Center, United Arab Emirates University, Al Ain, UAE
*Correspondence: linglloyd@utar.edu.my (L.L.); haithamabdulmohsinafan@duytan.edu.vn (H.A.A.)

Received: 27 November 2019; Accepted: 4 March 2020; Published: 23 March 2020

Abstract: In this research, the advanced multilayer perceptron (MLP) models are utilized to predict the free rate of expansion that usually occurs around the pipeline (PL) because of waves. The MLP model was structured by integrating it with three optimization algorithms: particle swarm optimization (PSO), whale algorithm (WA), and colliding bodies’ optimization (CBO). The sediment size, wave characteristics, and PL geometry were used as the inputs for the applied models. Moreover, the scour rate, vertical scour rate along the pipeline, and scour rate at both right and left sides of the pipeline were predicted as the model outputs. Results of the three suggested models, MLP-PSO, MLP-WA, and MLP-PBO, for both testing and training sessions were assessed based on different statistical indices. The results indicated that the MLP-CBO model performed better in comparison to the MLP-PSO, MLP-WA, regression, and empirical models. The MLP-CBO can be used as a powerful soft-computing model for predictions.

Keywords: scour rate; prediction model; optimization model; MLP model; colliding bodies’ optimization

1. Introduction

Background

Protecting hydraulic structures against scouring is a vital issue for water engineering. More specifically, the depth of scouring is the most interesting hydraulic parameter in the scouring phenomena [1]. The maximum depth of scouring is achieved due to the interaction of the flow,
bed materials, and the pipeline (PL). The start of scouring underneath a PL leads to free spans (FS) [2]. PLs are one of the most essential structures utilized to convey oil, gas, and drinking water. Landside and bed scouring below the PLs threaten the PLs [2]. Numerous experimental studies have been performed to determine the critical conditions for the onset of free span development [3]. Predicting three-dimensional free spans is required to provide PL stability. Figure 1 illustrates scouring behavior that can lead to free spans; the diagram shows the essential parameters to keep in mind when designing a pipeline that must survive storms, especially the FS length and growth rate. When the waves travel in opposite directions, the scour rate is propagated (Figure 1). For submarine pipelines, several experiments have been carried out to determine the depth and the time scale of the local scour. Notwithstanding the results of these studies, a precise prediction for the scour rate at all dimensions below the pipelines is required [2]. The empirical methods are restricted to the limits of the dataset range. In this context, it could be concluded that the empirical models do not experience proper generalization ability for estimating the maximum scour below the PLs. In recent decades, soft computing models such as genetic programming (GP) [4–7], artificial neural network (ANN) [8–11], adaptive neuro fuzzy interface system (ANFIS) [12–15], and regression tree [16,17] were successfully utilized to forecast the scour depth below PLs. It should be noted that the scour rate is dependent on the flow direction, flow characteristics, PL diameter, and sediment properties. Additionally, the structural damages related to human activities occur in the free spans [1]. The natural self-burial below the PLs is caused by the free spans. One of the most significant issues in the design of PLs is assessing the length of the free spans during storms.

![Three-dimensional scour process below the pipeline (PL)](image)

**Figure 1.** Three-dimensional scour process below the pipeline (PL) [18].

In the case of empirical models, no comprehensive study on the effects of different parameters on scouring have been reported [2].

An important motivation for hydraulic engineers is whether a hybrid soft computing model with an evolutionary algorithm can improve the accuracy of the performance in a field where they have yet to be examined. Different optimization algorithms such as genetic algorithm (GA), firefly algorithm (FFA), shark algorithm (SA), and bat algorithm (BA) have been utilized to train the soft computing models [19–23]. In recent years, multilayer perceptron models have been applied to forecast the scour depth around hydraulic structures. Applying the multilayer perceptron (MLP) models and optimization algorithms achieved successful performance in various fields of science such as scour forecast below the PLs, modeling of hydraulic jumps, spillway design, and stable open channel design [24–26]. In this study, the structure of an MLP model based on a new optimization algorithm (colliding bodies optimization (CBO)) was used for estimating three-dimensional scour along PLs. The vertical scour rate, scour rate along the PL, and scour rate to the left and right of the PL were computed as the outputs.

The physical conceptions of energy and momentum were utilized to define the CBO [27]. The CBO has been widely utilized in the field of mathematical sciences, structural optimization problems, benchmark functions, power engineering, and image processing [28–30]. The algorithm is simple and
does not rely on any internal parameters [30]. The outstanding enhancement presented in this research is the application of a CBO model at the neuron level of an MLP. As a result, there is no need for the MLP to update the weight values throughout the training level by a new optimization algorithm.

2. Material and Methods

2.1. Multilayer Perceptron (MLP)

MLP is known as a well-considered group of ANN-based neural models, which are capable of approximating nonlinear problems based on a layered structure [31,32]. In ANN models, the evolution of information occurs based on three kinds of layers: input, middle (hidden), and output. Figure 2 shows an MLP structure that has a single middle layer. Two functions can be performed on every node of the MLP: activation and summation. The output of input variables, bias values, and input values can be calculated based on the summation function explained in Equation (1):

$$S_j = \sum_{i=1}^{n} \omega_{ij}I_i + \beta_j$$

(1)

where \(n\) is the total number of inputs, \(I_i\) is the input variable \(i\), \(\beta_j\) is a bias value, \(\omega_{ij}\) is connection weight, \(I\) is the input layer, and \(j\) is in the next level. The activation function is utilized in the MLP model based on the results of Equation (2). Different activation functions can be used in the MLP model; according to previous research, the sigmoid function is used most often. This function can be computed based on Equation (2)

$$f_j = \frac{1}{1 + e^{-S_j}}$$

(2)

Figure 2. The MLP structure for the current study (\(\beta_H\): bias for the hidden neuron and \(\beta_H\): bias for the output neuron).

Therefore, the final output neuron \(j\) is computed using Equation (3):

$$y_j = f_j \left( \sum_{i=1}^{n} \omega_{ij}I_j + \beta_j \right)$$

(3)

After constructing the final structure of the MLP, the training process is used to tune the weighting vectors of the MLP model. These weighting coefficients should be updated to estimate the results and optimize the total error of the MLP model. The training level of the MLP model is a challenging level, which has an important effect on the ability of the MLP to solve complex problems (Figure 2).

2.2. Colliding Body Optimization

The colliding bodies’ optimization (CBO) algorithm uses the physical laws of energy and momentum that govern the collisions that occur between solid bodies to solve nonlinear problems.
A set of random initial populations is used to define the CBO. In each level of the algorithm, all populations are arranged according to their fitness functions. In this algorithm, one object collides with another object and they attempt to reach a minimum energy level. When a collision happens in an isolated system, the total momentum of the system of objects is conserved. The colliding bodies are regarded as the initial population. The mass of each colliding body is computed based on the following equation:

\[ M(i) = \frac{1}{\text{OF}_i} \sum_{i=1}^{n} \text{OF}_i \]  

(4)

where \( \text{OF} \) is the objective function and \( M \) is the matrix of the objects. The algorithm, like other optimization problems, has multiple agents. Each solution is regarded as a colliding body. Each solution has a specified mass based on Equation (4). A solution with good value has larger mass than those of the bad ones.

The arrangement of the colliding bodies’ (CBs) fitness function values is carried out in ascending order. The CBs are divided into two classes: moving and stationary. The stationary CBs are good solutions and the moving solutions move toward the stationary group. The velocity of stationary CBs is zero before the collision. The change of body position indicates the velocity of CBs before the collision. The time steps change with the number of iterations. The velocity of the moving objects before the collision is computed based on the following equations:

\[ v_i = x_i - x_{i-\frac{n}{2}}, i = \frac{n}{2} + 1, .., n \]  

(5)

where \( v_i \) is the velocity of \( i \)th CB in the moving object groups, \( x_i \) is the position of \( i \)th CB in the moving object groups, and \( x_{i-\frac{n}{2}} \) is the \( i \)th CB pair position of \( x_i \) in the previous group. The initial condition of positions is computed as follows:

\[ x^0_i = x^\min + \text{rand} \cdot (x^\max - x^\min), i = 1, 2, .., n \]  

(6)

where \( x^\min \) is the minimum value of the decision variable, \( x^\max \) is the maximum value of the decision variable, \( \text{rand} \) is a random number, and \( x^0_i \) is the initial value of the CBs.

The moving CBs collide with stationary CBs to enhance their locations. After the collision, the new velocities of objects are computed based on the following equation:

\[ v'_i = \frac{(m_i + \varepsilon m_{i-\frac{n}{2}}) v_{i-\frac{n}{2}}}{(m_i + m_{i-\frac{n}{2}})}, i = \frac{n}{2}, \frac{n}{2} + 1, .., \frac{n}{2} + \varepsilon \]  

(7)

The velocity of CBs in moving groups is as follows:

\[ v'_i = \frac{(m_i - \varepsilon m_{i-\frac{n}{2}}) v_{i}}{(m_i + m_{i-\frac{n}{2}})}, i = \frac{n}{2}, \frac{n}{2} + 1, \frac{n}{2} + 2, \frac{n}{2} + 3, .., n \]  

(8)

where \( m \) is the mass of each colliding body and \( \varepsilon \) is the coefficient of restitution of the two colliding bodies:

\[ \varepsilon = 1 - \frac{\text{iter}}{\text{iter(max)}} \]  

(9)

where \( \text{iter} \) is the current iteration number and \( \text{iter(max)} \) is the total number of iterations.

The new position for each stationary object is as follows:

\[ x'^{\text{new}}_i = x_{i-\frac{n}{2}} + \text{rand} \cdot v'_i \]  

(10)
where $x_{i}^{new}$ is the new position after the collision of the $i$th moving CB, $x_{i}^{old}$ is the old position of the $i$th stationary object, and $rand$ is a random number. The sign $(.)$ denotes an element-by-element multiplication.

The new position for each moving object is as follows:

$$x_{i}^{new} = x_{i} + rand \cdot v_{ii}'$$

(11)

where $x_{i}^{new}$ is the new position after the collision of the $i$th stationary CB, $x_{i}$ is the old position after the collision of the $i$th stationary CB, and $v_{ii}'$ is the velocity after the collision of the $i$th stationary object. The sign $(.)$ denotes an element-by-element multiplication.

The CBO is designed based on the physical laws of collision between two objects that use the concept of conversation of energy; as certain objects collide with another object, the collided system moves in concordance with the principle of conservation of energy. In fact, in most collisions of two objects, one object loses momentum and hence slows down while the other gains momentum and speeds up, as presented in the following formula and in Figure 3.

$$m_{1}u_{1} + m_{2}u_{2} = m_{1}v_{1} + m_{2}v_{2}$$

(12)

where $m_{1}$ is the mass of the first object, $m_{2}$ is the mass of the second object, $u_{1}$ is the initial velocity of the first object, $u_{2}$ is the initial velocity of the second object, $v_{1}$ is the final velocity of the first object, and $v_{2}$ is the final velocity of the second object.

**Figure 3.** The structure of the colliding body (CB) algorithm (NFE: Maximum Number of Objective Function Evaluations).
2.3. Particle Swarm Optimization (PSO)

The PSO algorithm was inspired by the social behavior of particles in the search space. The PSO is made of two components: position and velocity. The position vectors are the values of decision variables. The velocity vectors are utilized to update the position vectors:

\[ \vec{X}_i(t + 1) = \vec{X}_i(t) + \vec{V}_i(t + 1) \]  
\[ \vec{V}_i(t + 1) = w\vec{V}_i + c_1r_1(\vec{P}_i(t) - \vec{X}_i(t)) + c_2r_2(\vec{G}(t) - \vec{X}_i(t)) \]

where \( \vec{X}_i(t + 1) \) is the position of the \( i \)th particle at the \( t \)th iteration, \( \vec{V}_i \) is the velocity of \( i \)th particle at the \( t \)th iteration, \( w \) is inertia weight, \( c_1 \) is the individual coefficient, \( c_2 \) is the social coefficient, \( r_1 \) is a random number, \( r_2 \) is a random number, \( \vec{P}_i(t) \) is the best solution gained by the \( i \)th particle until the \( t \)th iteration, and \( \vec{G}(t) \) is the best solution obtained by all particles until the \( t \)th iteration (Figure 4).

\[ \text{Figure 4.} \] The details of the particle swarm optimization (PSO) components including location and velocity of particles.

2.4. Whale Algorithm (WA)

The WA was inspired from the bubble-net foraging trait of whales in nature. Whales utilize bubbles to trap prey. In the algorithm, each whale (candidate solution) can contain the \( n \) decision variables. The position of each whale is updated based on the following equation:

\[ \vec{Y}_{i+1} = \vec{Y}(t) - \vec{A.D} \]
\[ \vec{D} = \left| \vec{C.\vec{Y}(t) - \vec{Y}_i} \right| \]

where \( \vec{Y}_i \) is the solution vector in the \( i \)th iteration, \( \vec{\gamma}(t) \) is the possible location of prey in the \( i \)th iteration, \( \| \) is the absolute value, and the sign (·) is an element-by-element multiplication. \( \vec{A} \) and \( \vec{C} \) are parameter vectors computed for each dimension:

\[ \vec{A} = 2\vec{\alpha}.\vec{R}_1 - \vec{a} \]  
\[ \vec{C} = 2.\vec{R}_2 \]
where \( \vec{R}_1 \) is a random vector, \( \vec{R}_2 \) is a random vector, and \( \vec{a} \) is a linearly decreasing coefficient from 2 to 0.

The whale algorithm uses the above equations to relocate to different origins around a given prey. Whales use a spiral movement to attack prey. The shape of this movement and its effect on the movement of whales is shown in Figure 5. The spiral movement is updated based on the following equation:

\[
\vec{Y}_{t+1} = \vec{D} e^{bt} \cos(2\pi nt) + \vec{Y}_t^* \tag{18}
\]

![Figure 5. Bubble-net feeding behavior of humpback whales.](image)

Moreover, the whales encircle prey during the hunt. In order to mathematically simulate the encircling trait, the following equation is used:

\[
\vec{Y}(t+1) = \begin{cases} 
\vec{Y}(t+1) - \vec{A} \vec{D} & r_3 \leq 0.5 \\
\vec{D} e^{bt} \cos(2\pi nt) + \vec{Y}(t) & r_3 > 0.5
\end{cases}
\tag{19}
\]

where \( r_3 \) is a random number and \( \vec{D} = |\vec{C} \vec{Y}(t) - \vec{Y}_t^*| \), \( \vec{D} : |\vec{Y}(t) - \vec{Y}_t| \), and \( b \) are constants for determining the shape of spiral movement (Figure 5). \( |\vec{C} \vec{Y}(t) - \vec{Y}_t^*| \) indicates the distance of the \( i \)th whale to the prey (best solution obtained so far).

2.5. Optimization Algorithms for Training MLPs

To combine the MLP and optimization algorithms, two technical aspects should be determined: how to encode the solutions and how to determine the objective function. In applied optimization algorithms, all particles, whales, and colliding bodies can be encoded as one-dimensional vectors, including random numbers in \([-1, 1]\), where all are developed using MATLAB and R software (the codes are not available). Each solution shows a candidate MLP. Figure 6 shows the details of the experimental model. As in Figure 7, the encoded agent contains a set of bias terms and weight connections. The number of weights and biases in the MLP is used to calculate the length of this vector. To calculate the objective function, all agents (particles, whales, and colliding bodies) are sent to the MLP model as the connection weights and bias terms. The MLP model assesses those agents based on
the used training datasets. Then, the MLP indicates the fitness of all agents. The following objective function is used:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (z_i - \hat{z}_i)^2$$

(20)

where $MSE$ is the mean squared error, $n$ is the number of data points, $z_i$ is the observed data, and $\hat{z}_i$ is the estimated data. The following indices also are used to evaluate the performance of the models:

$$PBIAS = 100 \times \frac{\sum_{i=1}^{N} (Y_{obs} - Y_{sim})}{\sum_{i=1}^{N} Y_{obs}}$$

(21)

$$NSE = 1 - \frac{\left[ \sum_{i=1}^{n} (Y_{obs} - Y_{sim})^2 \right]}{\left[ \sum_{i=1}^{n} (Y_{obs} - Y_{mean})^2 \right]}$$

(22)

$$MAE = \frac{1}{N} \sum_{i=1}^{n} |Y_{obs} - Y_{sim}|$$

(23)

where $MAE$ is mean absolute error, $NSE$ is Nash–Sutcliff efficiency, $Y_{obs}$ is the observed data, $Y_{mean}$ is the average of observed data, $Y_{sim}$ is the simulated data, and $PBIAS$ is percent bias.

![Figure 6. The details of experimental model [33].](image)

![Figure 7. Structure of a decision variable in the Salp Swarm Algorithm (SSA) including bias and weight.](image)

3. Datasets

In recent years, many studies have analyzed pipeline (PL) scour by virtue of realizing the lateral propagation of scour holes along the PLs. The laboratory studies indicated that the scour rates rely on wave properties, physical properties of materials (such as soil (sandy or clay), erosive materials,
and the size of bed materials), and the PL geometry (PL arrangement). The following equations are used to define the relationship between scour propagation and effective parameters [34]:

\[
\begin{align*}
V^*(H) &= f(\theta_w, KC, \frac{e}{B}, \sin \alpha) \\
V^*(R) &= f(\theta_w, KC, \frac{e}{B}, \sin \alpha) \\
V^*(L) &= f(\theta_w, KC, \frac{e}{B}, \sin \alpha) \\
V^*(V) &= f(\theta_w, KC, \frac{e}{B}, \sin \alpha)
\end{align*}
\]  

(24)

where \(V^*(H)\) is the scour rate (SR) along the PL, \(V^*(R)\) is the SR to the right of the PL, \(V^*(L)\) is the SR to the left of the PL, \(V^*(V)\) is the vertical SR, \(e\) is the embedment depth, \(B\) is the pipe diameter, \(KC\) is the Keulegan parameter, \(\theta_w\) is the shields parameter, and \(\alpha\) is the flow incident angle to the PL.

\[
\begin{align*}
\theta_w &= \frac{\tau_w}{\rho g (\frac{\rho_s}{\rho} - 1)d_{50}} \\
KC &= \frac{U_w T}{B} \\
f_w &= 0.023r^{-0.52} \\
r &= \frac{U_w T}{4\pi d_{50}}
\end{align*}
\]  

(25)

where \(\rho\) is the mass density of water, \(d_{50}\) is the median sediment size, \(PL\) is the pipe diameter, \(U_w\) is the maximum undistributed orbital rate, \(T\) is the wave period, and \(\rho_s\) is the mass density of the bed material. Cheng et al. (2014) [33] carried out laboratory studies in a wave flume of 50 m in length and 2.5 m in depth and 4 m in width. In the current study, a concrete sandpit of 4 m length and 0.25 depth was constructed as the test section. The downstream end of the sandpit was 7 m from the perforated stone beach and 4 m from the wavemaker. Table 1 shows the range of parameters for this experiment. The live bed status was considered for all experiments. In this study, 38 datasets were used (25 datasets for training and 13 datasets for the testing level). The inputs were \(H, T, U_w, \) and \(e_m\) and the outputs were \(V^*_{H}, V^*_{L}, V^*_{R}, \) and \(V^*_{V}.\) The data published by [33] were used in this study.

| Parameters | Range          |
|------------|----------------|
| \(H\) (m) (input) | 0.13–0.17      |
| \(T\) (s) (input)  | 1.5–2.0        |
| \(U_w\) (m/s) (input) | 0.29–0.45     |
| \(e_m\) (mm) (input) | 5–20          |
| \(V^*_{H}\) (mm/s) (output) | 1.27–5.162  |
| \(V^*_{L}\) (mm/s) (output) | 1.19–4.527  |
| \(V^*_{R}\) (mm/s) (output) | 1.11–4.49   |
| \(V^*_{V}\) (mm/s) (output) | 0.592–2.405  |
| \(\theta_w\) (input) | 0.18–0.30     |
| \(KC\) (input)     | 8.7–18         |
| \(e/B\) (input)    | 0.10–0.40      |
| \(\alpha\) (rad) (input) | 0.0–0.70     |

4. Discussion and Results

4.1. The Selection of Optimization Algorithm Parameters

The performance of optimization algorithms rely on the correct selection of parameters. Among different models, the Taguchi model has been successfully applied in a wide range of studies. The Taguchi model utilizes an orthogonal array to categorize the experimental outputs. The Taguchi model attempts to find the best level of random parameters and also minimize the impact
of noise. Here, the pleasing value is known by the signal term (S). The noise (N) also shows the undesirable value. The \( S/N \) is computed by the following equation:

\[
\frac{S}{N} \text{ ratio} = \frac{1}{10} \log \left( \frac{\text{objective function}}{2} \right)
\]  

(26)

The aim is to maximize the signal-to-noise ratio. The minimum number of experiments to be conducted is based on the following equation:

\[
N = 1 + NV(L - 1)
\]

(27)

where \( N \) is the minimum number of experiments, \( L \) is the number of levels, and \( NV \) is the number of parameters. For example, The PSO algorithm was selected to explain the application of the Taguchi model because the PSO has more random parameters in comparison to the WA and CBO models.

The population size, individual coefficient, the social coefficient, and inertia weight were regarded as random parameters. Each parameter has four levels (Table 2a). According to the Taguchi model, the minimum number of experiments is equal to 13. The \( S/N \) ratio was computed for each level against each parameter. The larger \( S/N \) ratio corresponds to the best values of parameters. Similarity, the WA and CBO parameters were obtained in this article (Table 2b,c). In fact, random parameters such as population size influence the outputs of the algorithms. Table 2b shows the computed \( S/N \) rate versus the population size. Four levels were considered for the population size of the CBO. The \( S/N \) with the highest value shows the best size for the CBO.

Table 2. (a) The parameter levels and signal-to-noise (S/N) ratio for the PSO, (b) the best parameters for the colliding bodies’ optimization (CBO), and (c) the best parameters for the whale algorithm (WA).

| Parameter         | Population Size | Inertia Weight | Individual Coefficient | Social Coefficient |
|-------------------|-----------------|----------------|------------------------|--------------------|
| Level 1           | 100, S/N:0.87   | 0.2, S/N:0.21  | 1.6, S/N:0.32          | 1.6, S/N:0.30      |
| Level 2           | 200, S/N:0.76   | 0.40, S/N:0.15 | 1.8, S/N:0.54          | 1.8, S/N:0.41      |
| Level 3           | 300 S/N:0.82    | 0.60, S/N:0.42 | 2.0, S/N:0.30          | 2.0, S/N:0.24      |
| Level 4           | 400, S/N:0.87   | 0.80, S/N:0.55 | 2.2, S/N:0.45          | 2.2, S/N:0.32      |

b

| Parameter | Population Size |
|-----------|-----------------|
| Level 1   | 100, S/N:0.84   |
| Level 2   | 200, S/N:0.96   |
| Level 3   | 300 S/N:0.83    |
| Level 4   | 400, S/N:0.87   |

c

WA, population size: 100, the maximum number of iterations: 200

4.2. The Statistical Results for Different Soft Computing Models

The results for the testing and training levels are presented in Table 3. From the testing results, the longitudinal scour rate (LSR) showed better accuracy in comparison with vertical, left, and right scour rates (Table 3). In other words, the LSR was the most accurate rate as verified by NSE = 0.92, PBIAS = 0.14, and MAE = 0.367 mm/s calculated for the testing data. Table 3 indicates that the MLP-CBO provided the vertical scour rate with the highest accuracy in terms of MAE = 0.416, NSE = 0.90, and PBIAS = 0.20 compared to the scour rates to the right and left of the PLs. In the \( V_R^* \) forecast, outputs indicated that the MLP-CBO provided a more accurate estimation of scour rates than those using the \( V_L^* \).
Table 3. The statistical results for soft computing models. MAE: mean absolute error; PBIAS: percent bias; NSE: Nash–Sutcliffe efficiency; MLP: multilayer perceptron.

| Model          | MAE (mm/s) | PBIAS | NSE  |
|----------------|------------|-------|------|
| **Train**      |            |       |      |
| MLP-CBO ($V_{H^*}$) | 0.345      | 0.12  | 0.95 |
| MLP-WA ($V_{H^*}$)  | 0.389      | 0.17  | 0.93 |
| MLP-PSO ($V_{H^*}$) | 0.393      | 0.22  | 0.92 |
| **Test**       |            |       |      |
| MLP-CBO ($V_{H^*}$) | 0.367      | 0.14  | 0.92 |
| MLP-WA ($V_{H^*}$)  | 0.379      | 0.19  | 0.91 |
| MLP-PSO ($V_{H^*}$) | 0.391      | 0.23  | 0.90 |
| **Train**      |            |       |      |
| MLP-CBO ($V_{V^*}$) | 0.412      | 0.18  | 0.91 |
| MLP-WA ($V_{V^*}$)  | 0.422      | 0.22  | 0.89 |
| MLP-PSO ($V_{V^*}$) | 0.434      | 0.25  | 0.87 |
| **Test**       |            |       |      |
| MLP-CBO ($V_{V^*}$) | 0.416      | 0.20  | 0.90 |
| MLP-WA ($V_{V^*}$)  | 0.432      | 0.29  | 0.88 |
| MLP-PSO ($V_{V^*}$) | 0.449      | 0.32  | 0.86 |
| **Train**      |            |       |      |
| MLP-CBO ($V_{R^*}$) | 0.512      | 0.27  | 0.89 |
| MLP-WA ($V_{R^*}$)  | 0.522      | 0.32  | 0.87 |
| MLP-PSO ($V_{R^*}$) | 0.523      | 0.34  | 0.85 |
| **Test**       |            |       |      |
| MLP-CBO ($V_{R^*}$) | 0.534      | 0.29  | 0.87 |
| MLP-WA ($V_{R^*}$)  | 0.541      | 0.35  | 0.86 |
| MLP-PSO ($V_{R^*}$) | 0.555      | 0.37  | 0.85 |
| **Train**      |            |       |      |
| MLP-CBO ($V_{L^*}$) | 0.612      | 0.31  | 0.86 |
| MLP-WA ($V_{L^*}$)  | 0.621      | 0.39  | 0.84 |
| MLP-PSO ($V_{L^*}$) | 0.629      | 0.42  | 0.83 |
| **Test**       |            |       |      |
| MLP-CBO ($V_{L^*}$) | 0.714      | 0.33  | 0.85 |
| MLP-WA ($V_{L^*}$)  | 0.738      | 0.42  | 0.82 |
| MLP-PSO ($V_{L^*}$) | 0.742      | 0.45  | 0.80 |

At the testing level, Table 3 demonstrates that the MLP-WA produced a better estimation of the longitudinal scour rate than those gained using the $V_{R^*}$, $V_{V^*}$, and $V_{L^*}$. In addition, the error indices of MAE and PBIAS given by the vertical scour rate are lower than those obtained using the $V_{L^*}$ and $V_{R^*}$. The MAE value given for forecasting the $V_{R^*}$ was 3.12, while this parameter for estimation of the $V_{L^*}$ was 3.18. In the $V_{R^*}$ prediction, outputs indicated that the MLP-WA provided more accurate estimation of scour rates than those using the $V_{L^*}$.

In the $V_{H^*}$ prediction, outputs of statistical parameters showed that the MLP-PSO model provided a more accurate prediction of scour rate (MAE: 0.391, NSE: 0.90, and PBIAS: 0.23 in the testing level) than those obtained using the $V_{R^*}$, $V_{V^*}$, and $V_{L^*}$. The MAE value with high accuracy in terms of MAE: 0.449, PBIAS: 0.32, and NSE: 0.86 (test level) compared to the $V_{R^*}$ and $V_{L^*}$. In the $V_{R^*}$ prediction, the results indicate that the MLP-PSO provides a more accurate forecast of scour rates than those obtained using the $V_{L^*}$. 
4.3. Comparison of Soft Computing Models

Twenty-five datasets were used as the inputs to the models for the training level to fine-tune the ANN models. The optimization algorithms were used to find the optimal value of the ANN parameters such as weight and connection. Then, the prepared models in the training level with 13 new datasets were tested to evaluate the accuracy of the model with the new data.

Performance of the MLP-CBO model showed better accuracy compared to the MLP-PSO and MLP-WA models. For $V_L^*$ prediction, the suggested MLP-PSO model gave inaccurate results with MAE: 0.742 mm/s, PBIAS: 0.45, and NSE 0.80 (test level) compared with those of the MLP-CBO and MLP-WA models. The error indices given by the MLP-CBO model for the $V_V^*$ showed a significantly superior result to those of the MLP-PSO and MLP-WA models. In addition, the results showed that the MLP-WA model provided a better estimation than the MLP-PSO model. In Figure 8, a Taylor diagram was drawn for the final simulation outputs of all the applied computing models using three criteria of RMSE (root mean square error), standard deviation, and correlation. According to the obtained results for $V_H^*$, it can be observed that the obtained results of MLP-CBO model were close to and just below the semicircular zone RMSE < 0.5 and gave higher values of the correlation coefficient. The results indicate that the MLP-CBO model provided better results than the other models. Figure 9 indicates the convergence graph for the different optimization algorithms. The results indicate that the CBO model showed faster convergence in comparison to the other algorithms. However, the MLP-CBO model had more accuracy than the other models. The CBO model could find more accurate values for the weights and biases. Thus, the MLP-CBO model had the least value of error function among all the models. The accuracy level of MLP-CBO was high for all parameters $V_L^*$, $V_V^*$, $V_H^*$, and $V_R^*$. Additionally, the MLP-CBO model had faster performance than the other algorithms because of faster convergence. The general results indicate that all models had a better estimation for $V_H^*$ in comparison with parameters $V_L^*$, $V_V^*$, $V_H^*$, and $V_R^*$. It should be noted that all models carried out estimations based on four input datasets while the empirical models need more inputs. Although the ANN models gave more accurate estimations, they ignore the physical interactions in the pipes. The numerical models have to consider more detail for boundary conditions for estimation of scour parameters.
Figure 8. Taylor diagram for soft computing models. Based on correlation, RMSE, and standard deviation for $V_{R^*}$, $V_{V^*}$, $V_{L^*}$, and $V_{H^*}$.

Figure 9. Convergence curve for algorithms using the number of iterations and objective function value.
4.4. Comparison Analysis

Recently, Cheng et al. (2014) [33] proposed an empirical equation to estimate the value of the scour rate longitudinally as follows:

\[
V_H^* = 11.3 \left(1 - \frac{e}{B}(1 + \sin(\alpha))\right)\theta_w^{\frac{5}{3}} \sqrt{\frac{5}{3}}
\] (28)

The following equations have been proposed for the estimation of scour rates. They use the least square method to estimate scour rates [34].

\[
V_H^* = 256.7(\theta_w)^{1.31} KC^{-10.054} \left(\frac{e}{B}\right)^{-0.02} (1 + \sin \alpha)^{-0.103}
\] (29)

\[
V_L^* = 1.765(\theta_w)^{0.579} KC^{-0.37} \left(\frac{e}{B}\right)^{-0.308} (1 + \sin \alpha)^{-1.288}
\] (30)

\[
V_R^* = 0.468(\theta_w)^{0.579} KC^{-0.37} \left(\frac{e}{B}\right)^{-0.308} (1 + \sin \alpha)^{-1.288}
\] (31)

\[
V_V^* = 0.661(\theta_w)^{0.949} KC^{-0.397} \left(\frac{e}{B}\right)^{-0.355} (1 + \sin \alpha)^{-0.38}
\] (32)

Statistical results of the MLP-CBO indicate significantly lower error in prediction of \(V_H^*\) in comparison with those of Equations (28) and (29) (Table 4). Equation (28) was found to be the best model among other equations. Performance of the empirical equations showed lower accuracy in prediction of scour rates in comparison with those of the soft computing models. NSE, PBIAS, and MAE values given by the Equation (29) for the \(V_H^*\) forecast gave a superior result to the Equation (28) in this case. Performance of the MLP-PSO and MLP WA models showed notably accurate estimation of the scour rates compared with the results estimated using the empirical models.

Table 4. The results of models for all data.

| MODEL               | MAE (MM/S) | PBIAS | NSE  |
|---------------------|------------|-------|------|
| EQUATION (28)       | 1.59       | 0.55  | 0.87 |
| EQUATION (29)       | 1.34       | 0.52  | 0.86 |
| EQUATION (30)       | 1.37       | 0.56  | 0.85 |
| EQUATION (31)       | 1.42       | 0.55  | 0.81 |
| EQUATION (32)       | 1.45       | 0.54  | 0.80 |
| MLP-CBO (VH*)       | 0.367      | 0.14  | 0.90 |
| MLP-WA (VH*)        | 0.391      | 0.23  | 0.89 |
| MLP-PSO (VH*)       | 0.399      | 0.25  | 0.87 |
| MLP-CBO (VL*)       | 0.721      | 0.42  | 0.89 |
| MLP-WA (VL*)        | 0.745      | 0.45  | 0.87 |
| MLP-PSO (VL*)       | 0.814      | 0.49  | 0.86 |
| MLP-CBO (VR*)       | 0.621      | 0.32  | 0.85 |
| MLP-WA (VR*)        | 0.634      | 0.36  | 0.82 |
| MLP-PSO (VR*)       | 0.642      | 0.38  | 0.80 |
| MLP-CBO (VV*)       | 0.521      | 0.23  | 0.88 |
| MLP-WA (VV*)        | 0.555      | 0.25  | 0.89 |
| MLP-PSO (VV*)       | 0.591      | 0.27  | 0.89 |

4.5. Parametric Analysis

Figure 10 indicates the effect of \(\alpha\) on the scour propagation along the PL. The first result was related to \(\alpha = 0\) and \(KC = 8.7\). The MLP-PSO and MLP-WA models underestimated \(V_H^*\), while Equation (29) overestimated \(V_H^*\) (Figure 10a). The results indicate that the MLP-CBO model provided higher accuracy of \(V_H^*\) compared to those of the other models. For \(\alpha = 0\) and \(KC = 15.8\), the MLP-PSO and
MLP-WA models significantly overestimated the \( V_\text{H}^* \). The MLP-CBO model provided higher accuracy of scour rate estimation in comparison to those of the other models (Figure 10). From Figure 10c, the outputs indicate that MLP-PSO and MLP-WA models could not present an accurate estimation of \( V_\text{H}^* \) for \( e/B = 0.1 \). Figure 10d shows that the MLP-PSO and MLP-WA models indicated a downward trend, which overestimated the \( V_\text{H} \) in comparison to that of the MLP-CBO model.

\[
\alpha = 0, K_c = 8.7 \\
\alpha = 0, K_c = 15.8 \\
\alpha = 15, K_c = 15.8 \\
\alpha = 15, K_c = 18
\]

**Figure 10.** The variation of parameters versus scour rates. (a) \( \alpha = 0, K_c = 8.7 \); (b) \( \alpha = 0, K_c = 15.8 \); (c) \( \alpha = 15, K_c = 15.8 \); (d) \( \alpha = 15, K_c = 18 \).

The details of the uncertainty computation and the Latin hypercube sampling for the model are as follows:

1. At the beginning, the fitness \( g(b) \) and the uncertainty ranges for the parameters are determined, where mean square error has been chosen as the objective function.
2. The Latin hypercube is performed in the range of \([b_{\text{min}}, b_{\text{max}}]\), which is initially set to \([b_i, \text{abs}_{\text{mean}}, b_j, \text{abs}_{\text{max}}]\); the corresponding fitness functions are evaluated and the sensitivity matrix \( J \) and the parameters covariance matrix \( C \) are calculated.

It should be mentioned that the uncertainty of inputs affects the outputs. In fact, the data are experimental and the measurement of data may have errors. Thus, the errors in the input data transfer to the output data when the input data are used to simulate the outputs. Thus, the uncertainty boundary for the outputs should be computed. When the soft computing models are used, the error may propagate in the whole of structure of the model.

Thus, it is necessary to evaluate the uncertainty of applied soft computing models. The Monte Carlo (MC) method provides a perfect statistical explanation of changes in the output variables due to the uncertainties in the inputs, which could be present in the following equations at different stages in the computation of uncertainty:

\[
J_{ij} = \frac{\Delta g_i}{\Delta b_j}
\]  
(33)
\[ C = s^2 \left( (J^T J)^{-1} \right)^{-1} \]  

To evaluate the uncertainty of models, three indices were used:

\[ p = \frac{1}{k} \text{count}[X | X_L \leq X \leq X_U] \]  

\[ \bar{d} = \frac{\bar{d}_x}{\sigma_x} \]  

\[ \bar{d}_x = \frac{1}{k} \sum_{l=1}^{k} (X_u - X_l) \]

where \( k \) is the number of data points, \( X_u \) is the upper boundary of the input variable, \( X_L \) is the lower boundary of the variable, \( \sigma_x \) is the standard deviation, \( \bar{d}_x \) is the average distance between the upper and lower boundaries obtained, \( X_i \) is the input variable, and \( p \) is the percentage of the observed data bracketed by the 95% prediction uncertainty. A value of less than one is good for the \( d \) parameter. The models with larger \( p \) (maximum: 100%) values indicate better performance. The results of the uncertainty analysis are shown in Figure 11. It is clear that the MLP-CBO model increased the accuracy of the MLP-WA and MLP-PSO models. It can be concluded that the MLP-CBO and MLP-PSO models are the best and worst models, respectively. For \( V_l^* \) prediction, the suggested MLP-CBO model gave inaccurate results with \( p: 0.97 \) and \( d: 0.12 \) (test level) compared with those of the MLP-PSO and MLP-WA models. In the \( V_R^* \) forecast, outputs indicated that the MLP-CBO model provided higher \( p \) and lower \( d \) values than those using the \( V_L^* \).

Figure 11. Results of uncertainty analysis of soft computing models using \( d \) and \( p \) indices.
The RMSE-observations standard deviation ratio (RSR) is a useful index to evaluate the computational models as follows:

\[
RSR = \sqrt{\frac{\sum_{i=1}^{N} (Y_{\text{obs}} - Y_{\text{sim}}^i)^2}{\sum_{i=1}^{N} (Y_{\text{obs}} - Y_{\text{mean}}^i)^2}}
\]

(38)

where \(Y_{\text{mean}}^i\) is the average observed data. The general performance rating for the RSR index is shown in Table 5:

| Performance Rating | RSR Value |
|--------------------|------------|
| Very Good          | 0.00 \leq RSR \leq 0.5 |
| Good               | 0.500 \leq RSR \leq 0.600 |
| Satisfactory       | 0.600 \leq RSR \leq 0.700 |
| Unsatisfactory     | RSR > 0.7 |

Table 5. General performance rating for the RMSE-observations standard deviation ratio (RSR) index [35].

Figure 12 shows the RSR value for the soft computing models. The error indices given by the MLP-CBO model for \(V_R^*, V_V^*, V_H^*,\) and \(V_L^*\) showed very good performance compared with those of the MLP-PSO and MLP-WA models.

5. Conclusions

In this article, the MLP-CBO, MLP-PSO, and MLP-WA models were applied to forecast the scour rates below the PL due to waves. The PSO, CBO, and WA were used to update the weighting coefficients through the training level. The results indicate that the MLP-CBO model forecasted the longitudinal
scour rate with MAEs of 0.345 and 0.367 in the training and testing level, respectively. In the VR* prediction, the results reveal that the MLP-PSO model provides a more accurate scour rate forecast than those obtained using the VL*. The performance of the empirical equations showed a lower accuracy in the prediction of scour rates in comparison with those of the soft computing models. Considering the complexity of the problem, assessment of the scour rates could be carried out successfully using the design curves. However, there is a need to apply a particular pre-processing analysis for data in order to have a more appropriate input–output pattern that could lead to a more accurate prediction for scour rate. In addition, it would be useful to integrate the model with a multi-objective algorithm that might simultaneously identify the most effective inputs and find the optimal values of the ANN parameters.

In fact, some studies have a large number of inputs, so the selection of the appropriate inputs was complex. Thus, methods such as the principle component analysis or multi-objective algorithms can be used to enhance the accuracy of models with a selection of appropriate inputs.

Author Contributions: Conceptualization, M.E. and F.B.B.; Methodology, A.N.A. and L.L.; Software, S.D.L. and H.A.A.; Validation, C.M.F.; Writing—original draft, A.E.-S. All authors have read and agreed to the published version of the manuscript.

Funding: The authors appreciate the financial support received from the research grant coded GPD082A-2018 funded by the University of Malaya and from IPSR of the Universiti Tunku Abdul Rahman.

Acknowledgments: The authors gratefully acknowledge the technical facility support received from the University of Malaya, Malaysia.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

| Abbreviation | Full Form |
|--------------|-----------|
| ANFIS        | Adaptive Neuro Fuzzy Interface System |
| ANN          | Artificial Neural Network |
| BA           | Bat Algorithm |
| CBO          | Colliding Bodies’ Optimization |
| FFA          | Firefly Algorithm |
| FS           | Free Spans |
| GA           | Genetic Algorithm |
| GP           | Genetic Programming |
| MLP          | Multilayer Perceptron |
| MLP-CBO      | Multilayer Perceptron-Colliding Bodies’ Optimization |
| MLP-PSO      | Multilayer Perceptron-Particle Swarm Optimization |
| MLP-WA       | Multilayer Perceptron-Whale Algorithm |
| PL           | Pipe Line |
| PSO          | Particle Swarm Optimization |
| SA           | Shark Algorithm |
| WA           | Whale Algorithm |

References

1. Parsaie, A.; Haghiabi, A.H.; Moradinejad, A. Prediction of Scour Depth below River Pipeline Using Support Vector Machine. *KSCE J. Civ. Eng.* 2019, 23, 2503–2513. [CrossRef]
2. Peng, Z.; Zou, Q.-P.; Lin, P. A Partial Cell Technique for Modeling the Morphological Change and Scour. *Coast. Eng.* 2018, 131, 88–105. [CrossRef]
3. Mawat, M.J.; Khudier, A.S.; Hashim, S.J. Evaluation Study of Free Spanning Subjected to Hydrodynamic Loads. *J. Univ. Babylon* 2018, 26, 227–237.
4. Najafzadeh, M.; Shiri, J.; Rezaie-Balf, M. New Expression-Based Models to Estimate Scour Depth at Clear Water Conditions in Rectangular Channels. *Mar. Georesour. Geotechnol.* 2017, 36, 227–235. [CrossRef]
5. Jamei, M.; Ahmadianfar, I. Prediction of Scour Depth at Piers with Debris Accumulation Effects Using Linear Genetic Programming. *Mar. Georesour. Geotechnol.* 2019, 1–12. [CrossRef]
6. Sharafati, A.; Yasa, R.; Azamathulla, H.M. Assessment of Stochastic Approaches in Prediction of Wave-Induced Pipeline Scour Depth. J. Pipeline Syst. Eng. Pract. 2018, 9. [CrossRef]

7. Najafzadeh, M.; Kargar, A.R. Gene-Expression Programming, Evolutionary Polynomial Regression, and Model Tree to Evaluate Local Scour Depth at Culvert Outlets. J. Pipeline Syst. Eng. Pract. 2019, 10. [CrossRef]

8. Dang, N.M.; Tran Anh, D.; Dang, T.D. ANN Optimized by PSO and Firefly Algorithms for Predicting Scour Depths around Bridge Piers. Eng. Comput. 2019. [CrossRef]

9. Moradi, F.; Bonakdari, H.; Kisli, O.; Ebtehaj, I.; Shiri, J.; Gharabaghi, B. Abutment Scour Depth Modeling Using Neuro-Fuzzy-Embedded Techniques. Mar. Georesour. Geotechnol. 2018, 37, 190–200. [CrossRef]

10. Niazkar, M.; Afzali, S.H. Developing a New Accuracy-Improved Model for Estimating Scour Depth around Piers Using a Hybrid Method. Iran. J. Sci. Technol. Trans. Civ. Eng. 2018, 43, 179–189. [CrossRef]

11. Eghbalzadeh, A.; Hayati, M.; Rezaei, A.; Javan, M. Prediction of Equilibrium Scour Depth in Uniform Non-Cohesive Sediments Downstream of an Apron Using Computational Intelligence. Eur. J. Environ. Civ. Eng. 2016, 22, 28–41. [CrossRef]

12. Sreedhara, B.M.; Rao, M.; Mandal, S. Application of an Evolutionary Technique (PSO–SVM) and ANFIS in Clear-Water Scour Depth Prediction around Bridge Piers. Neural Comput. Appl. 2018, 31, 7335–7349. [CrossRef]

13. Hassanzadeh, Y.; Jafari-Bavil-Olyaie, A.; Aalami, M.-T.; Kardan, N. Experimental and Numerical Investigation of Bridge Pier Scour Estimation Using ANFIS and Teaching-Learning-Based Optimization Methods. Eng. Comput. 2018, 35, 1103–1120. [CrossRef]

14. Aamir, M.; Ahmad, Z. Estimation of Maximum Scour Depth Downstream of an Apron under Submerged Wall Jets. J. Hydroinform. 2019, 21, 523–540. [CrossRef]

15. Azamathulla, H.M.D.; Ghani, A.A. ANFIS-Based Approach for Predicting the Scour Depth at Culvert Outlets. J. Pipeline Syst. Eng. Pract. 2011, 2, 35–40. [CrossRef]

16. Haghiabi, A.H. Closure to “Prediction of River Pipeline Scour Depth Using Multivariate Adaptive Regression Splines” by Amir Hamzeh Haghiabi. J. Pipeline Syst. Eng. Pract. 2019, 10. [CrossRef]

17. Ahmad, N.; Bihs, H.; Myhrhaug, D.; Kamath, A.; Arntsens, Ø.A. Numerical Modeling of Breaking Wave Induced Seawall Scour. Coast. Eng. 2019, 150, 108–120. [CrossRef]

18. Cheng, L.; Yeow, K.; Zhang, Z.; Teng, B. Three-Dimensional Scour below Offshore Pipelines in Steady Currents. Coast. Eng. 2009, 56, 577–590. [CrossRef]

19. Ehteram, M.; El-Shafie, A.H.; Hin, L.S.; Othman, F.; Koting, S.; Karami, H.; Mousavi, S.-F.; Farzin, S.; Ahmed, A.N.; Zawawi, B.; et al. Toward Bridging Future Irrigation Deficits Utilizing the Shark Algorithm Integrated with a Climate Change Model. Appl. Sci. 2019, 9, 3960. [CrossRef]

20. Ehteram, M.; Binti Koting, S.; Afan, H.A.; Mohd, N.S.; Malek, M.A.; Ahmed, A.N.; El-shafie, A.H.; Onn, C.C.; Lai, S.H.; El-Shafie, A. New Evolutionary Algorithm for Optimizing Hydropower Generation Considering Multireservoir Systems. Appl. Sci. 2019, 9, 2280. [CrossRef]

21. Li, Y.; Jiang, P.; She, Q.; Lin, G. Research on Air Pollutant Concentration Prediction Method Based on Self-Adaptive Neuro-Fuzzy Weighted Extreme Learning Machine. Environ. Pollut. 2018. [CrossRef]

22. Ehteram, M.; Singh, V.P.; Ferdowsi, A.; Mousavi, S.F.; Farzin, S.; Karami, H.; Mohd, N.S.; Afan, H.A.; Lai, S.H.; Kisli, O.; et al. An Improved Model Based on the Support Vector Machine and Cuckoo Algorithm for Simulating Reference Evapotranspiration. PLoS ONE 2019, 14. [CrossRef] [PubMed]

23. Karami, H.; Ehteram, M.; Mousavi, S.-F.; Farzin, S.; Kisli, O.; El-Shafie, A. Optimization of Energy Management and Conversion in the Water Systems Based on Evolutionary Algorithms. Neural Comput. Appl. 2018, 31, 5951–5964. [CrossRef]

24. Zhang, R.; Wu, P. The Investigation of Shape Factors in Determining Scour Depth at Culvert Outlets. ISH J. Hydraul. Eng. 2019, 1–7. [CrossRef]

25. Das, B.S.; Devi, K.; Khatua, K.K. Prediction of Discharge in Converging and Diverging Compound Channel by Gene Expression Programming. ISH J. Hydraul. Eng. 2019, 1–11. [CrossRef]

26. Roushangar, K.; Foroudi Khour, A.; Saneie, M. Experimental Study and Artificial Intelligence-Based Modeling of Discharge Coefficient of Converging Ogee Spillways. ISH J. Hydraul. Eng. 2019, 1–8. [CrossRef]

27. Kaveh, A.; Dadras, A.; Montazeran, A.H. Chaotic Enhanced Colliding Bodies Algorithms for Size Optimization of Truss Structures. Acta Mech. 2018, 229, 2883–2907. [CrossRef]

28. Panda, A.; Pani, S. Determining Approximate Solutions of Nonlinear Ordinary Differential Equations Using Orthogonal Colliding Bodies Optimization. Neural Process. Lett. 2017, 48, 219–243. [CrossRef]
29. Kaveh, A.; Sabeti, S. Structural Optimization of Jacket Supporting Structures for Offshore Wind Turbines Using Colliding Bodies Optimization Algorithm. Struct. Des. Tall Spec. Build. 2018, 27, e1494. [CrossRef]

30. Kaveh, A.; Rezaei, M.; Shiravand, M.R. Optimal Design of Nonlinear Large-Scale Suspendome Using Cascade Optimization. Int. J. Space Struct. 2017, 33, 3–18. [CrossRef]

31. Ehteram, M.; Karami, H.; Mousavi, S.F.; Farzin, S.; Celeste, A.B.; Shafie, A.-E. Reservoir Operation by a New Evolutionary Algorithm: Kidney Algorithm. Water Resour. Manag. 2018, 32, 4681–4706. [CrossRef]

32. Ehteram, M.; Singh, V.P.; Karami, H.; Hosseini, K.; Dianatikhah, M.; Hossain, M.; Ming Fai, C.; El-Shafie, A. Irrigation Management Based on Reservoir Operation with an Improved Weed Algorithm. Water 2018, 10, 1267. [CrossRef]

33. Cheng, L.; Yeow, K.; Zang, Z.; Li, F. 3D Scour below Pipelines under Waves and Combined Waves and Currents. Coast. Eng. 2014, 83, 137–149. [CrossRef]

34. Najafzadeh, M.; Saberi-Movahed, F. GMDH-GEP to Predict Free Span Expansion Rates below Pipelines under Waves. Mar. Georesour. Geotechnol. 2018, 37, 375–392. [CrossRef]

35. Moriasi, D.N.; Arnold, J.G.; Van Liew, M.W.; Bingner, R.L.; Harmel, R.D.; Veith, T.L. Model Evaluation Guidelines for Systematic Quantification of Accuracy in Watershed Simulations. Trans. ASABE 2007, 50, 885–900. [CrossRef]