

A comprehensive study on the application of firefly algorithm in prediction of energy dissipation on block ramps

Indexed by:



Amin Mahdavi-Meymand^a, Wojciech Sulisz^{a,*}, Mohammad Zounemat-Kermani^{b,c}

^aPolish Academy of Sciences, Institute of Hydro-Engineering, ul. Kosciarska 7, 80-328 Gdansk, Poland

^bShahid Bahonar University of Kerman, Department of Water Engineering, Pajoohesh Sq, 76169-14111, Kerman, Iran

^cUniversity of Tabriz, Center of Excellence in Hydroinformatics, 29 Bahman Ave, 5166616471, Tabriz, Iran

Highlights

- Machine learning models were developed to predict energy dissipation on block ramps.
- Firefly algorithm was used as integrative method with machine learning methods.
- Computational time and stability of models in different runs were analyzed.
- Firefly algorithm increases the accuracy of standard machine learning models.

Abstract

In this study novel integrative machine learning models embedded with the firefly algorithm (FA) were developed and employed to predict energy dissipation on block ramps. The used models include multi-layer perceptron neural network (MLPNN), adaptive neuro-fuzzy inference system (ANFIS), group method of data handling (GMDH), support vector regression (SVR), linear equation (LE), and nonlinear regression equation (NE). The investigation focused on the evaluation of the performance of standard and integrative models in different runs. The performances of machine learning models and the nonlinear equation are higher than the linear equation. The results also show that FA increases the performance of all applied models. Moreover, the results indicate that the ANFIS-FA is the most stable integrative model in comparison to the other embedded methods and reveal that GMDH and SVR are the most stable technique among all applied models. The results also show that the accuracy of the LE-FA technique is relatively low, $RMSE=0.091$. The most accurate results provide SVR-FA, $RMSE=0.034$.

Keywords

firefly algorithm, machine learning, energy dissipation, block ramps.

This is an open access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/>) 

1. Introduction

Excessive kinetic energy causes damage to the hydraulic structures. Downstream of dams, the high flow velocity causes scours, which reduces the stability of a structure. To control the kinetic energy several techniques have been applied including designing flip buckets, designing stilling basins, designing stepped chutes, and designing block ramps [27]. A block ramp is a hydraulic structure that conducts the flow to a lower elevation producing high energy dissipation [3]. The dissipation causes the material of high roughness placed on the sloped of the structure. A typical block ramp is shown schematically in Fig. 2.

In Fig. 1 y_0 is the upstream flow depth, H is the block ramp height, ΔE is the energy dissipation, v is the flow velocity, $\frac{v^2}{2g}$ is the velocity head, Q is the flow discharge, a is the block ramp angle,

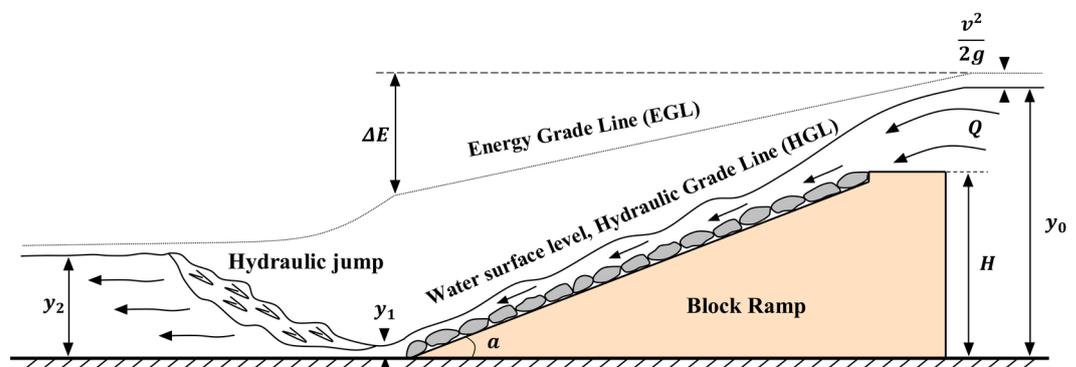


Fig. 1. A schematic view of block ramp

(*) Corresponding author.

E-mail addresses: A. Mahdavi-Meymand (ORCID: 0000-0002-9125-5214): amin.mahdavi1990@gmail.com, W. Sulisz (ORCID: 0000-0002-7484-5669): sulisz@ibwpan.gda.pl, M. Zounemat-Kermani (ORCID: 0000-0002-1421-8671): zounemat@uk.ac.ir

y_1 and y_2 are the upstream supercritical flow depth and the downstream subcritical flow depth of the hydraulic jump, respectively.

The design of block ramps must take into consideration the structure permeability and stability with a number of socio-economic and socio-ecologic aspects [3, 21, 27]. The block ramp purifies water as well as increases water aeration, which is important to decrease damages arising from cavitation [17]. The block ramps are resistant against destructive forces, especially forces arising from uplift pressure. The maintenance costs of block ramps are relatively low in comparison with corresponding maintenance costs of traditional hydraulic structures. The block ramps are considered to be eco-friendly type structures due to the possibility of building them from local material [23, 27].

The energy dissipation, ΔE , is an important parameter in the design of block ramps. In this regard, various experimental studies have been conducted to analyze different aspects of block ramp performance, especially the amount of dissipation energy under different hydraulic and geometric conditions [3, 23, 24, 27]. Pagliara and Chiavaccini [23] and Rahmanshahi and Shafai Bejestan [27] conducted some experiments and developed equations for the estimation of ΔE .

In recent years, the successful applications of machine learning methods in modeling engineering problems have been reported in numerous studies [4, 18, 19, 26, 33]. Despite the empirical formulas are easy to use, the outcome of research conducted in recent decades, including studies conducted in hydraulic engineering, indicate that the machine learning methods are more accurate [14, 16, 22, 29, 39]. Recently conducted studies have shown that the integration of machine learning models with meta-heuristic algorithms may drastically increase the accuracy of standard machine learning models. Firefly algorithm (FA) is an effective optimization approach developed by Yang [35]. The FA is a generalized form of the three popular meta-heuristic algorithms i.e. particle swarm optimization (PSO), differential evolution (DE), and simulated annealing (SA) [15]. Kumar and Kumar [15] discussed various aspects of FA in a study and concluded that the FA is a promising optimization algorithm. This motivated present work to derive novel and original techniques by developing integrative machine learning models embedded with the firefly algorithm, FA.

In this study, a number of machine learning techniques were applied to estimate the energy dissipation by block ramps. The applied models include multi-layer perceptron neural network (MLPNN), adaptive neuro-fuzzy inference system (ANFIS), group method of data handling (GMDH), support vector regression (SVR), and linear and nonlinear regression methods. First, the machine learning models are described. Then, a novel and original technique was derived by developing integrative machine learning models embedded with the firefly algorithm. Next, model evaluation criteria are introduced. Finally, the developed models are ranked based on innovative and original criteria including model accuracy, stability, and time duration in 30 consecutive runs, and conclusions are specified.

2. Material and methods

2.1. Empirical relations

Pagliara and Chiavaccini [23] developed the following relation by using regression methods to estimate ΔE based on their measured laboratory data sets:

$$\Delta E_r = \frac{\Delta E}{E_0} = A + (1 - A)e^{(B+C.S)y_c/H} \quad (1)$$

where ΔE_r is the relative energy dissipation, E_0 is the total energy upstream of the block ramp, S is the ramp slope, y_c is the critical water depth, and A , B and C are coefficients which are determined from Table 1.

Table 1. The value of coefficients A , B , and C

y_c / d_{50}	A	B	C
$y_c / d_{50} < 2.5$	0.33	-1.3	-14.5
$2.5 < y_c / d_{50} < 6.6$	0.25	-1.2	-12
$6.6 < y_c / d_{50} < 42$	0.15	-1	-11.5
$42 < y_c / d_{50}$	0.2	-0.9	-25

Note: d_{50} is the particle size at which 50% of ramp material by weight is finer

Rahmanshahi and Shafai Bejestan [27] conducted some experiments and developed two equations for estimating ΔE for smooth (Equation 2) and rough (Equation 3) ramps based on the gene-expression programming regression method:

$$\Delta E_r = e^{-49.06 \frac{y_c}{L}} + S + S^2 \left(1 - 35.54 \frac{y_c}{L} \right) \quad (2)$$

$$\Delta E_r = \left(S^{6.14} \frac{y_c}{L} \right)^L + \frac{y_c}{L} \left(2.25 - 0.3 \frac{y_c}{d_{50}} \right) \quad (3)$$

where L is the ramp length.

2.2. Machine learning methods

2.2.1. Multi-layer perceptron neural network

The MLPNN is one of the artificial neural network branches that can model complex problems. Like regression methods, MLPNN determines a relation between inputs and outputs and may be recommended to be applied even when a relationship is complex. Many studies report the successful application of MLPNN in engineering problems [8, 9, 11, 28]. The MLPNN also is a type of deep neural network [37]. In the MLPNN algorithm, there are three main layers including the input layer, middle layers, and output layer. Each layer is composed of several neurons connected to each other. The number of neurons in inputs and outputs layers is equal to the number of inputs and outputs parameters. The number of layers and neurons of the middle layers may vary. The input vector of neurons in the middle layers is calculated from the following equation:

$$A_i^{k+1} = \sum_{j=1}^n w_{ij} \times x_j^k + b \quad (4)$$

where n is the total number of middle layer neurons, x_j is the output of j th neuron of k layer, w_{ij} is the weight between j th neuron of layer k and i th neuron of $k+1$, and b is the bias. The output of the neurons of the middle layer may be written in the following form:

$$y_i = f(A_i) \quad (5)$$

where f is the activation function. The MLPNN can be trained by applying different optimization approaches such as conjugate gradient, gradient descent, and meta-heuristic algorithms. In this study, the Levenberg-Marquardt algorithm (LM) and the firefly algorithm (FA) were used to determine the network coefficients.

2.2.2. Adaptive neuro-fuzzy inference system

The adaptive neuro-fuzzy inference system (ANFIS) is a kind of artificial neural network introduced by Jang [12]. The ANFIS is based on the Takagi–Sugeno fuzzy inference system (FIS) that has the advantages of both neural networks and fuzzy logic principles in a single structure [12]. The previous studies approved the capability of ANFIS to approximate nonlinear functions in engineering problems [1, 5, 7, 31]. Generally, the ANFIS network consists of five layers. In Fig. 2 a basic flow diagram of ANFIS with four inputs is depicted.

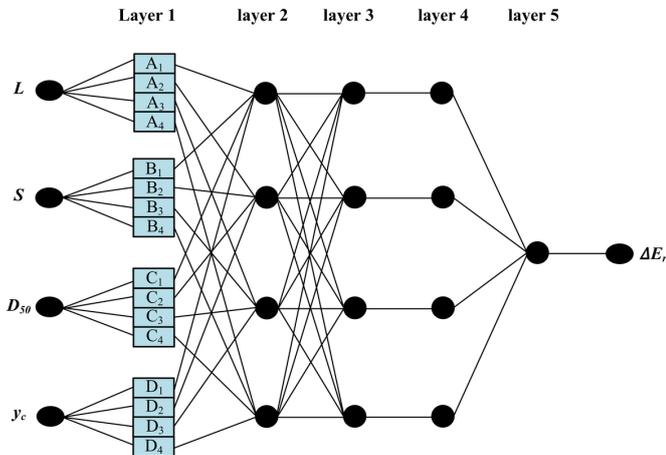


Fig. 2. A schematic view of the developed ANFIS structure

In Fig. 2 L , S , d_{50} , and y_c are inputs and ΔE_r is the output. The first layer, which is known as fuzzification, turns the input parameters into a fuzzy set by member functions as follow:

$$O_{A_i}^1 = \mu_{A_i}(L) \quad i = 1, 2, 3, 4 \quad (6)$$

$$O_{B_i}^1 = \mu_{B_i}(S) \quad i = 5, 6, 7, 8 \quad (7)$$

$$O_{C_i}^1 = \mu_{C_i}(D_{50}) \quad i = 9, 10, 11, 12 \quad (8)$$

$$O_{D_i}^1 = \mu_{D_i}(y_c) \quad i = 13, 14, 15, 16 \quad (9)$$

where μ is the membership function (type A, B, C. or D). In the second layer i.e. the multiplication layer, the weight of each rule is calculated from the following equation:

$$O_i^2 = w_i = \mu_{A_i}(L) \cdot \mu_{B_i}(S) \cdot \mu_{C_i}(D_{50}) \cdot \mu_{D_i}(y_c) \quad i = 1, 2, 3, 4 \quad (10)$$

The third layer is the normalization layer:

$$O_i^3 = \bar{w} = \frac{w_i}{\sum_i^4 w_i} \quad (11)$$

The fourth layer is the defuzzification layer. The output of this layer is obtained as below:

$$O_i^4 = \bar{w} f_i \quad i = 1, 2, 3, 4 \quad (12)$$

where f is a linear regression function. In the fifth layer, which is the summation layer, the output of the network can be calculated from the following equation:

$$O_i^5 = \sum_i \bar{w}_i f_i \quad i = 1, 2, 3, 4 \quad (13)$$

2.2.3. Group method of data handling

The group method of data handling (GMDH) belongs to the neural network branch of machine learning methods. The GMDH can be used to solve different problems such as classification, prediction, etc. Numerous studies have reported the successful application of GMDH in engineering problems [2, 13, 38]. Like other neural networks, the GMDH structure also includes the input layer, middle layers, and output layer. Each layer consists of several neurons. In Fig. 3 a schematic view of the developed GMDH with four inputs, five layers, and one output is shown.

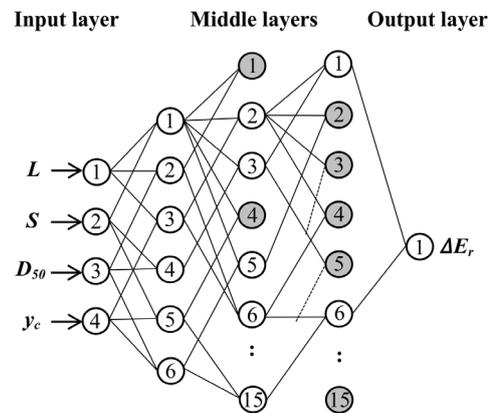


Fig. 3. A schematic view of the developed GMDH network

In the neurons of the GMDH network the connection between inputs and output variables can be expressed by Kolmogorov–Gabor polynomial equation [10]:

$$y = w_0 + \sum_{i=1}^N w_i x_i + \sum_{i=1}^N \sum_{j=1}^N w_{ij} x_i x_j + \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N w_{ijk} x_i x_j x_k + \dots \quad (14)$$

where x is the input vector, y is the output, and w is the weight vector. Usually, the second-order form of this equation is used as a transform function:

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2 \quad (15)$$

The weight coefficients can be determined by regression methods as well as optimization algorithms. In the GMDH network, the number of middle layers neurons increases from layer to layer. The number of middle layers neurons of quadratic polynomial GMDH is calculated from:

$$N^{j+1} = \binom{N^j}{2} \quad (16)$$

where N^{j+1} and N^j are the numbers of neurons of $j+1$ th and j th middle layer, respectively. As can be seen, the number of neurons will increase from layer to layer, which increases the complexity of a network. In this study, the maximum number of the middle layers neurons is assumed to be 10. In this strategy, the neurons were ranked based on their $RMSE$, 10 of the best neurons were selected for creating the network, and the rest were eliminated (the gray neurons in Fig. 3).

2.2.4. Support vector regression

The support vector machine (SVM) is a type of machine learning method that was originally used for classification problems. The SVR is an adaptation of SVM developed by Cortes and Vapnik [6] for re-

gression problems. The wide applications of SVR are reported in different disciplines including engineering [20, 25, 30, 36]. In the SVR method, the original data are mapped into a feature space of a higher dimension where a linear equation is fitted to the data with minimum complexity [32]. The main equation describing the implementation of SVR may be written in the following form:

$$f(x) = w, \varphi(x) + b \quad (17)$$

where $f(x)$ is a nonlinear regression function, $\varphi(x)$ is the nonlinear mapping of data in a space of higher dimension, and the vector w and the scalar b are the weights. The w and b are determined by the following optimization procedure:

$$\text{Minimize: } \frac{1}{2} w^2 + C \sum_{i=1}^N \xi_i + \xi_i^* \quad (18)$$

$$\text{Subject to: } \begin{cases} y_i - (w, \varphi(x_i) + b_i) \leq \varepsilon + \xi_i \\ (w, \varphi(x_i) + b_i) - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (19)$$

where ξ_i and ξ_i^* are the slack variables, C is the regularization parameter, and ε is the insensitive-loss function. This equation can be solved by the method of *Lagrange* multipliers. The final nonlinear regression equation of SVR can be expressed in the following form:

$$f(x) = \sum_{i=1}^n (a_i - a_i^*) K(x_i, x) + b \quad (20)$$

where a and a^* are the Lagrange multipliers, and $K(x_i, x)$ is the nonlinear kernel function. In this study the Gaussian kernel function was applied:

$$K(x_i, x) = \exp\left(-\frac{x_i - x^2}{2\sigma^2}\right) \quad (21)$$

where σ is the kernel function parameter.

2.3. Regression methods

In this study, in addition to machine learning methods, two simple regression equations i.e. linear equation (LE) and nonlinear equation (EQ) were applied to predict ΔE_r . These equations may be expressed in the following form:

$$\Delta E_r = A_0 + A_1 L + A_2 S + A_3 D_{50} + A_4 v_c \quad (22)$$

$$\Delta E_r = A_0 + A_1 L^{A_2} + A_3 S^{A_4} + A_5 D_{50}^{A_6} + A_7 v_c^{A_8} \quad (23)$$

where A is the weight coefficient vector. In this study firefly algorithm (FA) was applied to estimate the values of A .

2.4. Firefly algorithm

The firefly algorithm (FA) is a nature-inspired swarm-type algorithm that mimics the flashing behavior of the fireflies (is a nature-inspired algorithm mimics the flashing behavior of a swarm of fireflies. Yang [34] considered three ideal assumptions for developing FA:

First rule: the fireflies are unisex. So, each firefly can attract other fireflies distributed in the search space.

Second rule: The attraction of fireflies depends on their brightness. Between two fireflies, the brighter one will attract the less bright firefly. This means in the algorithm process the less bright firefly will move toward the brighter one. The brightness depends on the distance between fireflies-it will decrease with increasing the distance. The fireflies will move randomly if there is no bright firefly.

Third rule: The brightness of the population is determined by the cost function.

In FA, the distance between fireflies i and j (r_{ij}) is defined via Euclidean distance:

$$r_{ij} = x_i - x_j = \sqrt{\sum_{k=1}^m (x_i^k - x_j^k)^2} \quad (24)$$

where x denotes the position of fireflies and m is the dimension of the problem under consideration. The attractiveness of the population is determined by the following equation:

$$\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^2} \quad (25)$$

In the above equation, β_0 is the attractiveness at $r_{ij} = 0$. In the end, the new position of fireflies is calculated by the following equation:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) \quad (26)$$

2.5. Integrative FA-machine learning methods

In this study, standard machine learning methods and integrative machine learning methods combined with the firefly algorithm (FA) were applied to predict the energy dissipation over block ramps. The firefly algorithm was incorporated in the process of the training of models. The training procedure with FA for all machine learning methods is the same. At first, it is necessary to determine which parameters should be optimized and introduce an objective/cost function. For MLPNN weights and biases (Eq. 4), for ANFIS membership function parameters, for SVR regularization (C), insensitive loss (ε), and kernel function (σ) parameters, for GMDH weights of quadratic polynomial function (Eq. 15), and for regression methods weight coefficients (Eq. 22 and 23) were optimized with the FA. The *RMSE* between measured and predicted values is considered as an objective/cost function. The FA is a population-based optimization algorithm. So before starting the optimization process, the number of population and other constant parameters must be initialized. In general, these parameters are initialized based on trial and error, user experiences, and previous studies. The considered values of FA parameters are presented in Table 2.

In the next stage, the population/fireflies must be distributed randomly in the search space and their positions are updated based on the FA equations. In the end, the optimized values must be set to the models which provide the predicted values. For a better view of these steps, in Fig. 4 the flowchart of the optimization process of the applied models by applying FA is illustrated.

2.6. Input data

In this study, ramp length (L), particle diameters (d_{50}), ramp slope (S), and critical flow depth (v_c) were considered as input parameters to model the energy dissipation on block ramps (ΔE_r). In total 465 data were used to create the models. These data were collected from two published recourses. Pagliara and Chiavaccini [23] carried out an experimental study on the water surface profile over the block ramps and determined the amount of energy dissipation. In these ex-

Table 2. The considered values of the initial parameters of FA

Parameter	Value
Mutation Coefficient Damping	0.97
Light Absorption Coefficient (γ)	1
Attraction Coefficient (β_0)	2
Mutation Coefficient (a)	0.2
Population (MLPNN)	100
Population (ANFIS)	200
Population (GMDH)	100
Population (SVR)	10
Population (Regression equations)	100

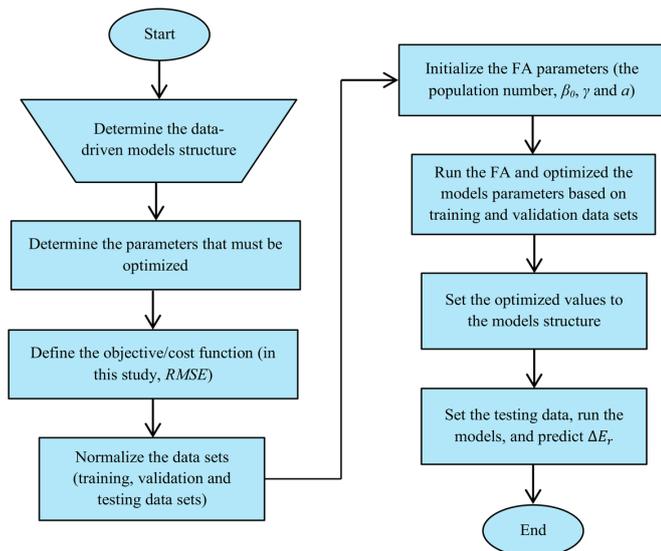


Fig. 4. Diagram of the training procedure of the applied integrative MLPNN, ANFIS, SVR, GMDH, and regression methods combined with FA

Table 3. The details of experimental data applied in the modeling

Parameter	Rahmanshahi and Shafai Bejestan [27]			Pagliara and Chiavaccini [23]		
	D	AVG	sd	D	AVG	sd
L (m)	1.9	1.9	0	[2.1, 6.91]	4.441	1.97
d_{50} (mm)	[0, 30]	15.52	10.031	[64, 133.8]	96.72	19.885
S	[0.083, 0.25]	0.174	0.063	[0.101, 0.333]	0.199	0.097
y_c (m)	[0.041, 0.114]	0.079	0.022	[0.022, 0.114]	0.070	0.021
ΔE_r	[0.104, 0.865]	0.539	0.166	[0.607, 0.965]	0.787	0.080
Number of data	210			255		

Note: D denotes the range; AVG denotes the average value; Sd denotes the standard deviation

Table 4. The ranges of the training, validation and testing data sets

Parameter	Training			Validation			Testing		
	D	AVG	sd	D	AVG	sd	D	AVG	sd
L (m)	[1.9, 6.91]	3.28	1.94	[1.9, 6.91]	3.26	1.88	[1.9, 6.91]	3.38	1.92
d_{50} (mm)	[0, 133.8]	60.72	43.14	[0, 133.8]	58.74	43.37	[0, 133.8]	58.25	45.41
S	[0.08, 0.33]	0.19	0.08	[0.08, 0.33]	0.18	0.09	[0.08, 0.33]	0.18	0.07
y_c (m)	[0.02, 0.11]	0.07	0.02	[0.03, 0.11]	0.07	0.02	[0.03, 0.11]	0.08	0.02
ΔE_r	[0.18, 0.96]	0.68	0.17	[0.10, 0.94]	0.66	0.19	[0.13, 0.93]	0.65	0.20

Note: D denotes the range; AVG denotes the average value; Sd denotes the standard deviation

periments, the discharge range was from 14 to 106 l/s and the ramp height was equal to 0.7 m. Different ramps geometry and hydraulic conditions were considered in hydraulic experiments conducted by Rahmanshahi and Shafai Bejestan [27]. A total of 255 data were extracted from this publication. Rahmanshahi and Shafai Bejestan [27] conducted laboratory experiments in a flume of 8 m length, 0.3 m wide, and 0.8 m height. The discharge varied from 8 to 36 l/s, and the block size varied from 0.0 mm to 30 mm. A total of 210 data were extracted from these experiments. The details of experimental data are presented in Table 3.

2.7. Modeling procedure

The data set was divided randomly into three categories including the training data set that comprises 70% of all data, the validation data set (15%), and the testing data set (15%). In Table 4 the ranges of the data sets are presented.

All models were trained based on the training data set. The validation data was used to prevent the overfitting of models in the training process. The testing data set was used to evaluate the performance of applied models. In the end, the results of models in 30 consecutive different runs were analyzed.

2.8. Model Evaluation Criteria

Five statistical parameters such as the root mean square error (RMSE), mean absolute error (MAE), coefficient of determination (R^2), Nash–Sutcliffe efficiency (NSE), and index of agreement (IA) were used to analyze and evaluate errors and output results. These parameters are calculated from the following equations:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\Delta E_{ri}^p - \Delta E_{ri}^o)^2} \quad (27)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |\Delta E_{ri}^p - \Delta E_{ri}^o| \quad (28)$$

$$R^2 = \frac{\left(\sum_{i=1}^N (\Delta E_{ri}^p - \overline{\Delta E_r^p}) (\Delta E_{ri}^o - \overline{\Delta E_r^o}) \right)^2}{\sum_{i=1}^N (\Delta E_{ri}^p - \overline{\Delta E_r^p})^2 \sum_{i=1}^N (\Delta E_{ri}^o - \overline{\Delta E_r^o})^2} \quad (29)$$

$$NSE = 1 - \frac{\sum_{i=1}^N (\Delta E_{ri}^p - \Delta E_{ri}^o)^2}{\sum_{i=1}^N (\Delta E_{ri}^o - \overline{\Delta E_r^o})^2} \quad (30)$$

$$IA = 1 - \frac{\sum_{i=1}^N (\Delta E_{ri}^p - \Delta E_{ri}^o)^2}{\sum_{i=1}^N \left(|\Delta E_{ri}^p - \overline{\Delta E_r^p}| + |\Delta E_{ri}^o - \overline{\Delta E_r^o}| \right)^2} \quad (31)$$

where ΔE_r^p is the predicted energy dissipation, ΔE_r^o denotes the observed values, and $\overline{\Delta E_r}$ is the average value of energy dissipation.

3. Results

The final performance of the applied machine learning models (ANFIS, ANFIS-FA, MLPNN, MLPNN-FA, GMDH, GMDH-FA, SVR, and SVR-FA) along with the regression-based (LE-FA and NE-

FA) machine learning models are presented in Table 5 for the training and validation stages.

As can be seen in Table 5, the models were evaluated based on the average performance and the best and worst performances. The analysis of the results of machine learning models based on the *RMSE*, *NSE*, and coefficient of determination show that the best training process for the simulation of the energy dissipation provides ANFIS and SVR-FA. These models provide the lowest values for the *RMSE* and the highest values for the R^2 and *NSE* during the training and validation stages. On the other hand, the regression-based models (LE-FA and NE-FA) show weaker efficiency than the machine learning models. Although the standard GMDH model acts better than the regression-based models, the integrative GMDH-FA exhibits weaker performance than the regression-based models.

The differences between the statistical measures for the training and validation sets of MLPNN, SVR, ANFIS, MLPNN-FA, SVR-FA, and ANFIS-FA models in Table 5 imply that there is an unexplained variance for all these models. However, by the use of the validation set, the overtraining problem has been resolved.

In Table 6, the final assessment of the predictive models is given using five performance criteria, including the *RMSE*, *MAE*, *NSE*, *IA*, and R^2 . Unlike Table 5, Table 6 comprises the results of two other models i.e. the empirical model of Pagliara and Chivaccini [23] and the machine learning model of GEP reported by Rahmanshahi [27], so that one can have a better and unbiased judgment over the final accuracy of the models. Taking into account the statistical measures,

Table 5. Performance of the applied machine learning models estimating the dissipation of energy for the training and validation stages

Method	Result type	Statistical indices					
		Training			Validation		
		RMSE	R ²	NSE	RMSE	R ²	NSE
ANFIS	Best	0.016	0.99	0.99	0.041	0.984	0.955
ANFIS	Worst	0.026	0.976	0.976	0.047	0.976	0.94
ANFIS	Average	0.019	0.986	0.986	0.044	0.983	0.949
ANFIS-FA	Best	0.017	0.989	0.989	0.043	0.97	0.952
ANFIS-FA	Worst	0.037	0.954	0.951	0.047	0.977	0.941
ANFIS-FA	Average	0.023	0.981	0.981	0.045	0.979	0.947
MLPNN	Best	0.018	0.988	0.988	0.045	0.977	0.947
MLPNN	Worst	0.025	0.978	0.978	0.057	0.949	0.914
MLPNN	Average	0.02	0.985	0.985	0.049	0.97	0.936
MLPNN-FA	Best	0.022	0.982	0.982	0.043	0.977	0.95
MLPNN-FA	Worst	0.036	0.955	0.953	0.049	0.969	0.938
MLPNN-FA	Average	0.027	0.974	0.974	0.046	0.973	0.943
GMDH	-	0.035	0.955	0.955	0.053	0.96	0.925
GMDH-FA	Best	0.039	0.946	0.944	0.044	0.967	0.949
GMDH-FA	Worst	0.057	0.892	0.884	0.061	0.933	0.901
GMDH-FA	Average	0.048	0.919	0.915	0.052	0.946	0.927
SVR	-	0.034	0.963	0.959	0.052	0.941	0.929
SVR-FA	Best	0.016	0.991	0.991	0.027	0.982	0.98
SVR-FA	Worst	0.037	0.973	0.965	0.045	0.954	0.948
SVR-FA	Average	0.017	0.99	0.989	0.038	0.965	0.96
LE-FA	Best	0.076	0.789	0.789	0.086	0.813	0.805
LE-FA	Worst	0.084	0.782	0.744	0.088	0.802	0.794
LE-FA	Average	0.077	0.787	0.784	0.087	0.808	0.799
NE-FA	Best	0.044	0.93	0.93	0.05	0.938	0.934
NE-FA	Worst	0.071	0.861	0.816	0.065	0.92	0.889
NE-FA	Average	0.053	0.902	0.896	0.059	0.933	0.907

Table 6. Performance of the applied machine learning models predicting the dissipation of energy for the testing set

Method	Result type	Statistical indices				
		Testing				
		RMSE	R ²	MAE	NSE	IA
ANFIS	Best	0.039	0.98	0.032	0.961	0.99
ANFIS	Worst	0.049	0.953	0.036	0.938	0.985
ANFIS	Average	0.041	0.974	0.033	0.957	0.989
ANFIS-FA	Best	0.038	0.975	0.029	0.964	0.991
ANFIS-FA	Worst	0.046	0.953	0.033	0.947	0.986
ANFIS-FA	Average	0.04	0.974	0.03	0.96	0.99
MLPNN	Best	0.041	0.974	0.033	0.958	0.989
MLPNN	Worst	0.049	0.956	0.035	0.94	0.985
MLPNN	Average	0.044	0.968	0.034	0.951	0.988
MLPNN-FA	Best	0.039	0.973	0.03	0.962	0.99
MLPNN-FA	Worst	0.048	0.956	0.037	0.943	0.985
MLPNN-FA	Average	0.04	0.974	0.032	0.959	0.99
GMDH	-	0.054	0.942	0.04	0.927	0.981
GMDH-FA	Best	0.045	0.95	0.034	0.949	0.987
GMDH-FA	Worst	0.066	0.914	0.052	0.891	0.972
GMDH-FA	Average	0.052	0.94	0.039	0.933	0.982
SVR	-	0.044	0.961	0.033	0.951	0.986
SVR-FA	Best	0.028	0.983	0.019	0.98	0.995
SVR-FA	Worst	0.043	0.957	0.028	0.952	0.987
SVR-FA	Average	0.034	0.977	0.021	0.971	0.992
LE-FA	Best	0.089	0.807	0.069	0.802	0.946
LE-FA	Worst	0.093	0.794	0.065	0.781	0.927
LE-FA	Average	0.091	0.802	0.066	0.793	0.934
NE-FA	Best	0.043	0.963	0.034	0.954	0.988
NE-FA	Worst	0.062	0.911	0.048	0.905	0.976
NE-FA	Average	0.046	0.955	0.036	0.948	0.986
Pagliara and Chiavaccini [23]	-	0.043	0.955	0.033	0.954	0.988
GEP-Rahmanshahi and Shafai Bejestan [27]	-	0.07	0.925	0.049	0.876	0.967

the SVR-FA ($RMSE=0.034$, $NSE=0.971$) provides the best predictions for the energy dissipation values. The ANFIS-FA and MLPNN-FA ($RMSE=0.040$) also show a reasonable performance for the prediction process.

In addition to the average values of the statistical criteria, Table 6 provides useful information regarding the ranges of the statistical measures especially information regarding the weakest outcome and the best results. Among a number of possible comparisons which may be conducted by applying between the weakest and the best results, it is worth comparing the variance in the predicted results. In this regard, it is worth noting that the NSE values for the SVR model range from 0.952 for the weakest results to 0.980 for the best results, while for the GMDH-FA model this range is from 0.891 to 0.949. In other words, the lowest NSE value of the SVR-FA model is greater than the highest NSE value of the GMDH-FA model. This fact indicates the absolute superiority of the SVR-FA model over the GMDH-FA model.

On the whole, for the SVR model, the integrative SVR-FA showed considerably better performance than its standard counterpart. For instance, the SVR-FA improved the $RMSE$ values up to 22% and the NSE value up to 2% in comparison to the standard SVR model.

The integration process improved the performances of all the applied machine learning models, however, the embedding advantage is not blatant for the other applied models like the SVR-FA model. As an illustration, the GMDH-FA slightly enhanced the statistical measures compared to the standard GMDH model i.e. the average $RMSE$ of the GMDH-FA is equal to 0.052 while the $RMSE$ of the GMDH is 0.054.

Interestingly, the integrative nonlinear regression model, NE-FA, ($RMSE=0.043$ and $R^2=0.963$) followed by the empirical relation suggested by Pagliara and Chiavaccini [23] ($RMSE=0.043$ and $R^2=0.955$) act reasonably in comparison to the sophisticated machine learning models. Although these nonlinear relations do not provide as good results as the SVR-FA, ANFIS-FA, and MLPNN-FA models, they surpassed the standard machine learning models such as the SVR, MLPNN, ANFIS, GEP, and GMDH. As could be expected, the linear regression-based model, LE-FA, demonstrated the weakest performance compared to the nonlinear and machine learning models.

Fig. 5 and Fig. 6 are scatterplots that compare the observed and predicted energy dissipation values (ΔE) from the best standard and integrative machine learning methods, respectively. Obviously, the predicted values of the LE-FA and GEP models are more scattered

than the other applied models. It can also be seen that the SVR models (both the standard SVR and SVR-FA) tend to over-predict smaller values of energy dissipation ($\Delta E_r < 0.5$). Although the SVR-FA model acts better than the other models (the highest R^2 value equal to 0.977), the ANFIS-FA is more successful in mapping the predicted values close to the line of agreement (1:1 line) with the highest trendline slope coefficient close to 1 ($m = 0.979$).

In Table 7, the computational costs (CPU time) of a single and 30 consecutive runs related to the applied machine learning models are presented. While the computational times of the standard machine learning models for single runs, including ANFIS, MLPNN, GMDH, and SVR, are less than 4 seconds, their integrative FA counterparts are not as timely-efficient models as the standard ones. This issue reveals

the fact that embedding the FA model to the machine learning models cannot be always considered as an ultimate technique. Although most of the published research papers – together with this study – have already claimed that using integrative models could improve the accuracy of the standard machine learning models, the majority of them have not reported and compared the computational cost of the standard versus integrative models. In other words, if the computational time is not a confining factor for the user/modeler, then the application of integrative machine learning models is recommended.

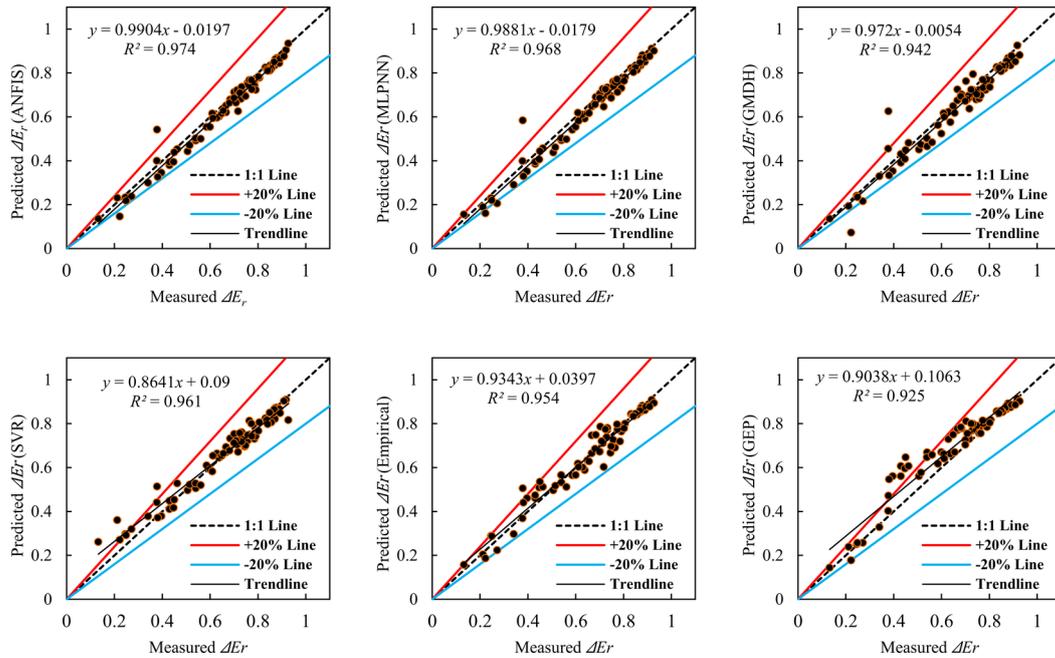


Fig. 5. Scatterplots comparing observed and predicted energy dissipation at the testing stage for standard machine learning models

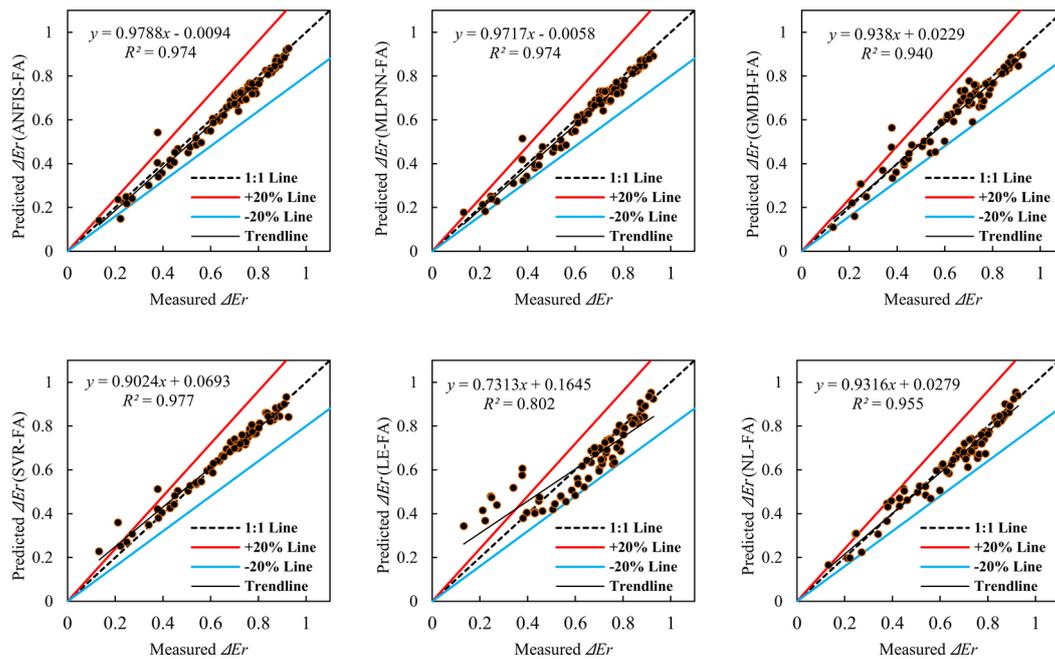


Fig. 6. Scatterplots comparing observed and predicted energy dissipation at the testing stage for integrative machine learning models

Table 7. Computational cost of a single and consecutive 30 runs-time duration (CPU: AMD Ryzen™ 7 PRO 4750U)

Method	Single run (s)	30 consecutive runs (s)
ANFIS	3	90
ANFIS-FA	11627	348810
MLPNN	3	90
MLPNN-FA	35978	1079340
GMDH	0.5	15
GMDH-FA	1124.8	33744
SVR	0.1	3
SVR-FA	138	4140
LE-FA	96	2880
NL-FA	580	17400
Summation	48436.6	1453098

4. Further Discussion

Fig. 7 depicts the box-whisker plots for the applied machine learning models and the testing set. The box-whisker plots illustrate the spreading of the dataset based on a five-element summary: the minimum, the first quartile, the median, the third quartile, and the maximum.

Such plots reveal two facts about the performance of the derived models: (i) the accuracy and (ii) the precision. As for the accuracy of the models, the SVR-FA shows the lowest values for the *RMSE* (e.g., the median equal to 0.038) and the highest values for the *NSE* (e.g., the median equal to 0.963). On the other hand, the LE model acts worst, in terms of accuracy, with the *RMSE* median equal to 0.091 and *NSE* median equal to 0.790. Regarding the precision of the models, the GMDH and SVR models were absolutely superior to the other applied models in such a way that no visual differences are observed between the minimum and maximum values. The corresponding box-whisker plots with the GMDH-FA and NE-FA show that these models have the widest maximum-minimum range (the distance between the lowest and highest parts of the box-whisker plots). This means that the GMDH-FA and NE-FA are the least precise applied models. Hence, an interesting fact can be noted from the plots in Fig 7. By embedding the FA algorithm to the MLPNN and SVR machine learning models, the accuracy of the models has been improved, while the precision of the models has not been enhanced. Embedding the FA to the ANFIS not only improved the accuracy of the model but also slightly upgraded the precision of the standard ANFIS. The GMDH-FA model was the only machine learning model whose performance has not been improved in comparison with the performance of a standard GMDH. Thus, unless more studies suggest using embedded GMDH models, it is not recommended to create an integrative version of the standard GMDH model. In the last decades, it has been shown that the machine learning models have better performance in comparison with the traditional methods. Nonetheless, the application of machine learning models is more difficult than empirical and regression equations. However, ΔE_r is an important parameter for the design of block ramps. Hence, the designers are recommended to consider the trained SVR-FA model to calculate ΔE_r in real water management projects.

5. Conclusion

In this study, comprehensive investigations were conducted to derive a novel and original technique by developing integrative machine learning models embedded with the firefly algorithm, FA, and apply-

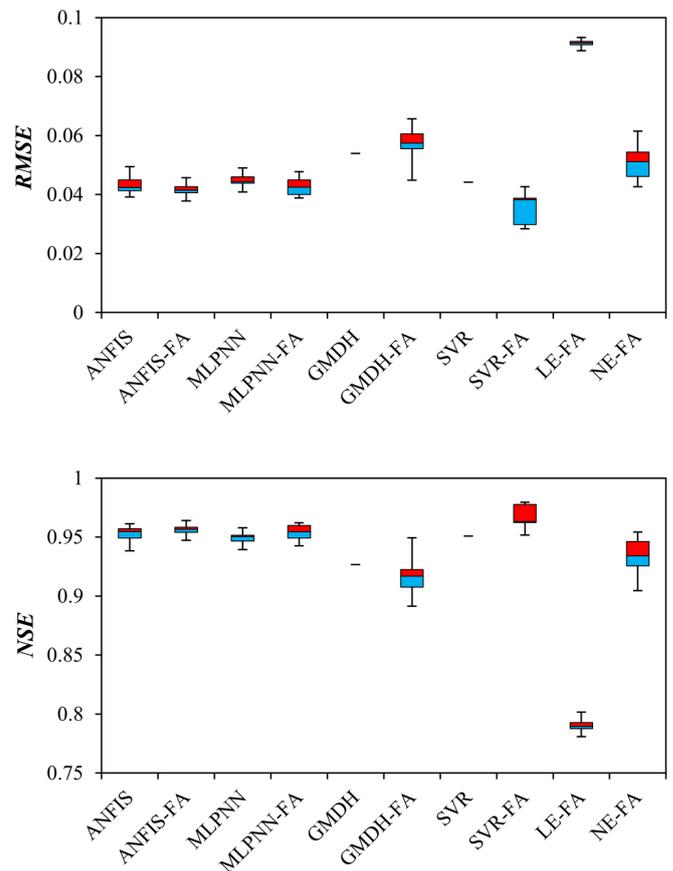


Fig. 7. Box-whisker plots for the *RMSE* and *NSE* values based on the 30 runs of the applied machine learning models

ing the derived models in the prediction of energy dissipation on block ramps (ΔE_r). Four standard machine learning models including multi-layer perceptron neural network (MLPNN), adaptive neuro-fuzzy inference system (ANFIS), group method of data handling (GMDH), and support vector regression (SVR), as well as linear and nonlinear regression equations, were chosen as the basic predictive methods. In addition to the aforementioned models, two empirically-based relations were applied for comparisons with the derive models. Four parameters including ramp length (L), particle diameter (d_{50}), ramp slope (S), and critical flow depth (y_c) were considered as inputs to the developed models. The performance of each model was evaluated in 30 consecutive runs. The effectiveness of the developed models was evaluated and ranked based on innovative and original criteria including model accuracy, stability, and time duration in 30 consecutive runs. The results show that the SVR-FA is the most accurate model among the developed models (*RMSE*=0.034). Moreover, the results show that the FA increases the accuracy of SVR, ANFIS, MLPNN, and GMDH by about 22.73%, 2.44%, 9.09%, and 3.70%, respectively. The results reveal that the standard GMDH and SVR are the most stable models, while their integration with FA reduces their stability. The computational cost analysis of the applied models reveals that the FA increases the time duration of the modeling process.

References

1. Abdulshahed AM, Longstaff AP, Fletcher S. The application of ANFIS prediction models for thermal error compensation on CNC machine tools. *Applied Soft Computing* 2015; 27: 158-168, <https://doi.org/10.1016/j.asoc.2014.11.012>.
2. Adnan RM, Liang Z, Parmar KS, Soni K, Kisi O. Modeling monthly streamflow in mountainous basin by MARS, GMDH-NN and DENFIS using hydroclimatic data. *Neural Computing and Applications* 2020, <https://doi.org/10.1007/s00521-020-05164-3>.
3. Ahmad, A, Srisvastava, D. Energy dissipation on block ramp with large scale roughness. In Proc., 5th Int. Symp. on Hydraulic Structures: Hydraulic Structures and Society-Engineering Challenges and Extremes. Barton, Australia, 2014, <https://doi.org/10.14264/uql.2014.41>.
4. Badora M, Sepe M, Bielecki M, Graziano A, Szolc T. Predicting length of fatigue cracks by means of machine learning algorithms in the small-data regime. *Eksploracja i Niezawodność - Maintenance and Reliability* 2021; 23(3): 575-585, <https://doi.org/10.17531/ein.2021.3.19>.
5. Bartoletti N, Casagli F, Marsili-Libelli S, Nardi A, Palandri L. Data-driven rainfall/runoff modelling based on a neuro-fuzzy inference system. *Environmental Modelling & Software* 2018; 106: 35-47, <https://doi.org/10.1016/j.envsoft.2017.11.026>.
6. Cortes C, Vapnik V. Support vector machine, *Mach Learn* 1995; 20: 273-297, <https://doi.org/10.1007/BF00994018>.
7. Dursun OF, Kaya N, Firat M. Estimating discharge coefficient of semi-elliptical side weir using ANFIS. *Journal of Hydrology* 2012; 426-427: 55-62, <https://doi.org/10.1016/j.jhydrol.2012.01.010>.
8. Feng X, Ma G, Su SF, Huang C, Boswell MK, Xue P. A multi-layer perceptron approach for accelerated wave forecasting in Lake Michigan. *Ocean Engineering* 2020; 211: 107526, <https://doi.org/10.1016/j.oceaneng.2020.107526>.
9. Freno BA, Carlberg KT. Machine-learning error models for approximate solutions to parameterized systems of nonlinear equations. *Computer Methods in Applied Mechanics and Engineering* 2019; 348: 250-296, <https://doi.org/10.1016/j.cma.2019.01.024>.
10. Ivahnenko AG. Polynomial theory of complex systems, *IEEE Trans. Syst, Man Cybern*, 1971.
11. Jaddi NS, Abdullah S. Optimization of neural network using kidney-inspired algorithm with control of filtration rate and chaotic map for real-world rainfall forecasting. *Engineering Applications of Artificial Intelligence* 2018; 67: 246-259, <https://doi.org/10.1016/j.engappai.2017.09.012>.
12. Jang JS. ANFIS: adaptive-network-based fuzzy inference system, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 1993; 23(3): 665-685, <https://doi.org/10.1109/21.256541>.
13. Jia X, Di Y, Feng J, Yang Q, Dai H, Lee J. Adaptive virtual metrology for semiconductor chemical mechanical planarization process using GMDH-type polynomial neural networks. *Journal of Process Control* 2018; 62: 44-54, <https://doi.org/10.1016/j.jprocont.2017.12.004>.
14. Kumar B. Neural network prediction of bed material load transport. *Hydrological Sciences Journal* 2012; 57(5): 956-966, <https://doi.org/10.1080/02626667.2012.687108>.
15. Kumar V, Kumar D. A Systematic Review on Firefly Algorithm: Past, Present, and Future. *Archives of Computational Methods in Engineering* 2020, <https://doi.org/10.1007/s11831-020-09498-y>.
16. Mahdavi-Meymand A, Sulisz W. 2022. Simulation of energy dissipation downstream of labyrinth weirs by applying support vector regression integrated with meta-heuristic algorithms. *Journal of Hydro-environment Research* 2022; 40: 91-101, <https://doi.org/10.1016/j.jher.2021.12.003>.
17. Mahdavi-Meymand A, Zounemat-Kermani M. A new integrated model of the group method of data handling and the firefly algorithm (GMDH-FA): application to aeration modelling on spillways. *Artificial Intelligence Review* 2020; 53: 2549-2569, <https://doi.org/10.1007/s10462-019-09741-4>.
18. Matuszczak M, Żbikowski M, Teodorczyk A. Predictive modelling of turbofan engine components condition using machine and deep learning methods. *Eksploracja i Niezawodność - Maintenance and Reliability* 2021; 23(2): 359-370, <https://doi.org/10.17531/ein.2021.2.16>.
19. Memar S, Mahdavi-Meymand A, Sulisz W. Prediction of seasonal maximum wave height for unevenly spaced time series by Black Widow Optimization algorithm. *Marine Structures* 2021; 78: 103005, <https://doi.org/10.1016/j.marstruc.2021.103005>.
20. Modaresi F, Araghinejad S, Ebrahimi K. A Comparative Assessment of Artificial Neural Network, Generalized Regression Neural Network, Least-Square Support Vector Regression, and K-Nearest Neighbor Regression for Monthly Streamflow Forecasting in Linear and Nonlinear Conditions. *Water Resources Management* 2018; 32: 243-258, <https://doi.org/10.1007/s11269-017-1807-2>.
21. Mohamed HI. Flow over Gabion Weirs. *Journal of Irrigation and Drainage Engineering* 2010; 136(8): 573-577, [https://doi.org/10.1061/\(ASCE\)IR.1943-4774.0000215](https://doi.org/10.1061/(ASCE)IR.1943-4774.0000215).
22. Najafzadeh M, Azamathulla HM. Neuro-Fuzzy GMDH to Predict the Scour Pile Groups due to Waves, *Journal of Computing in Civil Engineering* 2015; 29(5): 04014068, [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000376](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000376).
23. Pagliara S, Chiavaccini P. Energy Dissipation on Block Ramps. *Journal of Hydraulic Engineering* 2006; 132(1): 41-48, [https://doi.org/10.1061/\(ASCE\)0733-9429\(2006\)132:1\(41\)](https://doi.org/10.1061/(ASCE)0733-9429(2006)132:1(41)).
24. Pagliara S, Lotti I. Surface and sub-surface flow through the block ramps. *Journal of Hydraulic Engineering* 2009; 135(3): 366-374, [https://doi.org/10.1061/\(ASCE\)IR.1943-4774.0000070](https://doi.org/10.1061/(ASCE)IR.1943-4774.0000070).
25. Ping Z, Dongwei G, Tianyou C. Data-driven predictive control of molten iron quality in blast furnace ironmaking using multi-output LS-SVR based inverse system identification. *Neurocomputing* 2018; 308: 101-110, <https://doi.org/10.1016/j.neucom.2018.04.060>.
26. Qiu R, Wang Y, Wang D, Qiu W, Wu J, Tao Y. Water temperature forecasting based on modified artificial neural network methods: Two cases of the Yangtze River. *Science of The Total Environment* 2020; 737: 139729, <https://doi.org/10.1016/j.scitotenv.2020.139729>.
27. Rahmanshahi M, Shafai Bejestan, M. Gene-Expression Programming Approach for Development of a Mathematical Model of Energy Dissipation on Block Ramps. *Journal of Irrigation and Drainage Engineering* 2020; 146(2): 04019033, [https://doi.org/10.1061/\(ASCE\)IR.1943-4774.0001442](https://doi.org/10.1061/(ASCE)IR.1943-4774.0001442).
28. Reale C, Gavin K, Librić L, Jurić-Kaćunić D. Automatic classification of fine-grained soils using CPT measurements and Artificial Neural Networks. *Advanced Engineering Informatics* 2018; 36: 207-215, <https://doi.org/10.1016/j.aei.2018.04.003>.
29. Sasal M, Kashyap S, Rennie CD, Nistor L. Artificial neural network for bedload estimation in alluvial rivers. *Journal of Hydraulic Research* 2009; 47(2): 223-232, <https://doi.org/10.3826/jhr.2009.3183>.
30. Taravatroy N, Bahmanpouri F, Nikoo MR, Gualtieri C, Izady A. Estimation of air-flow parameters and turbulent intensity in hydraulic jump on rough bed using Bayesian model averaging. *Applied Soft Computing* 2021; 103: 107165, <https://doi.org/10.1016/j.asoc.2021.107165>.
31. Thakur AK, Kaviti AK, Singh R, Gehlot A. Specific Soft Computing Strategies for Evaluating the Performance and Emissions of an SI Engine Using Alcohol-Gasoline Blended Fuels-A Comprehensive Analysis, *Archives of Computational Methods in Engineering* 2020,

- <https://doi.org/10.1007/s11831-020-09499-x>.
32. Vapnik VN, Golowich S, Smola A. Support vector method for function approximation, regression estimation and signal processing. In: Mozer M, Jordan M, Petsche T (eds) *Advance in neural information processing system*, vol 9. MIT Press, Cambridge, 1997: 281-287.
 33. Wu DC, Bahrami Asl B, Razban A, Chen J. Air compressor load forecasting using artificial neural network. *Expert Systems with Applications* 2021; 168: 114209, <https://doi.org/10.1016/j.eswa.2020.114209>.
 34. Yang XS. Firefly algorithms for multimodal optimization. In: *International Symposium on Stochastic Algorithms 2009*: 169-178, https://doi.org/10.1007/978-3-642-04944-6_14.
 35. Yang XS. *Nature-inspired metaheuristic algorithms*, Luniver press, London, 2008.
 36. Yu K. Robust fixture design of compliant assembly process based on a support vector regression model. *The International Journal of Advanced Manufacturing Technology* 2019; 103: 111-126, <https://doi.org/10.1007/s00170-019-03488-6>.
 37. Zhang Z, Lei Y, Mao X, Yan M, Xu L, Zhang X. A study of effectiveness of deep learning in locating real faults. *Information and Software Technology* 2021; 131: 106486, <https://doi.org/10.1016/j.infsof.2020.106486>.
 38. Zounemat-Kermani M, Mahdavi-Meymand A, Hinkelmann R. Nature-inspired algorithms in sanitary engineering: modelling sediment transport in sewer pipes. *Soft Computing* 2021, <https://doi.org/10.1007/s00500-021-05628-1>.
 39. Zounemat-Kermani M, Mahdavi-Meymand A. Embedded fuzzy-based models in hydraulic jump prediction. *Journal of Hydroinformatics* 2021; 23(1): 151-170, <https://doi.org/10.2166/hydro.2020.347>.