water

MDPI

Article

Modeling of Monthly Rainfall–Runoff Using Various Machine Learning Techniques in Wadi Ouahrane Basin, Algeria

Mahdi Valikhan Anaraki [1], Mohammed Achite [2,3], Saeed Farzin [1], Nehal Elshaboury [3], Nadhir Al-Ansari [4,*] and Ismail Elkhrachy [5,*]

[1] Department of Water Engineering and Hydraulic Structures, Faculty of Civil Engineering, Semnan University, Semnan 35131-19111, Iran; mvalikhan@semnan.ac.ir (M.V.A.); saeed.farzin@semnan.ac.ir (S.F.)
[2] Water and Environment Laboratory, Faculty of Nature and Life Sciences, Hassiba Benbouali University of Chlef, Chlef 02180, Algeria; m.achite@univ-chlef.dz
[3] Construction and Project Management Research Institute, Housing and Building National Research Centre, Giza 12311, Egypt; nehal.elshabory@hbrc.edu.eg
[4] Department of Civil, Environmental and Natural Resources Engineering, Lulea University of Technology, 97187 Lulea, Sweden
[5] Civil Engineering Department, College of Engineering, Najran University, Najran 66291, Saudi Arabia
* Correspondence: nadhir.alansari@ltu.se (N.A.-A.); iaelkhrachy@nu.edu.sa (I.E.)

Abstract: Rainfall–runoff modeling has been the core of hydrological research studies for decades. To comprehend this phenomenon, many machine learning algorithms have been widely used. Nevertheless, a thorough comparison of machine learning algorithms and the effect of pre-processing on their performance is still lacking in the literature. Therefore, the major objective of this research is to simulate rainfall runoff using nine standalone and hybrid machine learning models. The conventional models include artificial neural networks, least squares support vector machines (LSSVMs), K-nearest neighbor (KNN), M5 model trees, random forests, multiple adaptive regression splines, and multivariate nonlinear regression. In contrast, the hybrid models comprise LSSVM and KNN coupled with a gorilla troop optimizer (GTO). Moreover, the present study introduces a new combination of the feature selection method, principal component analysis (PCA), and empirical mode decomposition (EMD). Mean absolute error (MAE), root mean squared error (RMSE), relative RMSE (RRMSE), person correlation coefficient (R), Nash–Sutcliffe efficiency (NSE), and Kling Gupta efficiency (KGE) metrics are used for assessing the performance of the developed models. The proposed models are applied to rainfall and runoff data collected in the Wadi Ouahrane basin, Algeria. According to the results, the KNN–GTO model exhibits the best performance (MAE = 0.1640, RMSE = 0.4741, RRMSE = 0.2979, R = 0.9607, NSE = 0.9088, and KGE = 0.7141). These statistical criteria outperform other developed models by 80%, 70%, 72%, 77%, 112%, and 136%, respectively. The LSSVM model provides the worst results without pre-processing the data. Moreover, the findings indicate that using feature selection, PCA, and EMD significantly improves the accuracy of rainfall–runoff modeling.

Keywords: water resources engineering; rainfall–runoff modeling; machine learning techniques; hybrid models

check for updates

Citation: Anaraki, M.V.; Achite, M.; Farzin, S.; Elshaboury, N.; Al-Ansari, N.; Elkhrachy, I. Modeling of Monthly Rainfall–Runoff Using Various Machine Learning Techniques in Wadi Ouahrane Basin, Algeria. *Water* **2023**, *15*, 3576. https://doi.org/10.3390/w15203576

Academic Editor: Marco Franchini

Received: 6 September 2023
Revised: 6 October 2023
Accepted: 9 October 2023
Published: 12 October 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Accurate rainfall–runoff modeling has been one of the most popular subjects for hydrology researchers because of its importance for water resources planning and management, including dam design, reservoir operation planning, and flood mitigation management [1,2]. In addition, the development of these models enhances comprehension of the ongoing hydrological processes in the watersheds [3]. This topic has gained paramount attention in recent years because of the world's declining water supply, which necessitates the development of accurate modeling techniques [4]. The intricate link between rainfall and runoff makes it difficult to estimate runoff accurately [5]. This can be attributed to the heterogeneous distribution and the spatial-temporal fluctuations of hydrological components [6]. In addition to rainfall, wind speed, temperature, solar radiation, evapotranspiration, and other meteorological factors, catchment-specific characteristics (e.g., land cover, topography, soil type, and slope) affect river runoff changes. As a result, developing accurate models to capture this dynamic and nonlinear natural phenomenon is challenging because these interrelated factors take place at many temporal and geographical scales [7]. Additionally, it is challenging to gather predictor variables from a catchment system using large samples. The difficulties of accurate and quantitative representation of the available data provide the key problems in the modeling process.

In general, there are two categories of hydrological models: (a) conceptual and physical-based models and (b) empirical or data-driven models. The former models need a lot of input parameters and a lot of hydro-meteorological information. The applicability of these models to represent hydrological processes is frequently limited by these constraints [8]. Also, in the absence of accurate data on meteorological and site-specific parameters, the data-driven models are suitable for modeling the rainfall–runoff process due to their minimal input dataset requirements [9]. Machine learning and data-driven models have been effectively used in recent years to simulate the nonlinear and nonstationary runoff phenomenon [10–12]. These approaches can be used to simulate hydrological processes due to various physical phenomena, such as the periodicity, pattern, or randomness of model input and target data [13,14].

Tikhamarine et al. [15] introduced the combination of Harris Hawks optimization (HHO) with a multi-layer perceptron neural network and least squares support vector machine (LSSVM) to predict the rainfall–runoff. Based on the autocorrelation function (ACF), partial ACF (PACF), and cross-correlation function, five alternative situations were explored. The performance of the suggested models was compared with data-driven methodologies integrated with particle swarm optimization (PSO). The findings showed that hybrid models trained using HHO exhibited better performance in forecasting runoff compared with integrated models with PSO. Additionally, coupling LSSVM with HHO resulted in a high degree of runoff prediction accuracy. Adnan et al. [16] examined the application of four machine learning techniques to estimate rainfall–runoff at an hourly timeframe in the Italian Samoggia River basin. The models included a multi-model simple averaging ensemble approach, multiple adaptive regression splines (MARS), an M5 model tree, as well as an adaptive neuro-fuzzy inference system (ANFIS) with fuzzy c-means (FCM) and the PSO algorithm. The outcomes of the developed models were compared with the theoretical EBA4SUB model using five statistics: mean absolute error (MAE), root mean squared error (RMSE), Nash–Sutcliffe efficiency (NSE), modified index of agreement, and scatter index. The MARS, ANFIS-FCM, and ANFIS-PSO offered equal accuracy, which was better than the M5 model. The machine approaches often outperformed the EBA4SUB when compared to the conceptual event-based method; however, in some instances, the latter method provided higher accuracy than the M5 model and MARS.

Mohammadi [11] reviewed the application of machine learning approaches (e.g., support vector machine (SVM), artificial neural network (ANN), and ANFIS) for hydrological subjects, including streamflow, rainfall–runoff, surface hydrology, and flood modeling. Furthermore, the benefits and drawbacks of popular machine learning models were critically examined in the field of runoff modeling. Okkan et al. [17] integrated ANN and support vector regression (SVR) into a conceptual rainfall–runoff model for monthly runoff simulation in the Gediz River Basin, Turkey. The nested hybrid models' parameters were all calibrated at once. The nested hybrid models outperformed the standalone models and linked model versions in terms of mean and high flows, according to the performance metrics. Thus, the research affirmed the credibility of a modeling approach that combined a conceptual model and several machine learning approaches. Roy et al. [18] applied a deep neural network (DNN) and EO-ELM model that integrated an equilibrium optimizer (EO) and an extreme learning machine (ELM) for rainfall–runoff modeling in the UK's River Fal at Tregony and the Teifi in Glanteifi. In order to deploy the suggested models, an ideal amount of lag inputs was determined using PACF. The proposed models were validated in terms of prediction accuracy using ELM, kernel ELM, PSO-based ELM, SVR, ANN, and gradient boosting machines. Additionally, the research applied a discrete wavelet-based dataset preprocessing approach to improve the performance of the suggested models. This research demonstrated how well EO-ELM and DNN may be used for rainfall–runoff modeling.

Waqas et al. [19] developed radial basis function (RBF)-SVM and M5 models to model the rainfall–runoff process in the Jhelum River Basin, Pakistan. The models were trained and tested using various combinations of datasets. Modeled and observed data were assessed using the coefficient of determination (R^2), normalized RMSE, MSE, and coefficient of efficiency for the training and testing phases. According to the findings, gene expression programming was found to be the most precise and highly effective technique. Xiao et al. [20] developed a backpropagation neural network, a generalized regression neural network (GRNN), an ELM, and a wavelet neural network (WNN) for runoff forecasting in the Xijiang River. The GRNN model performed better in runoff forecasting by considering flood propagation time. The WNN model exhibited the highest accuracy in the 7-day lead time for water level. This study suggested a machine learning-based runoff forecasting model would enhance flood and drought early warning systems. Singh et al. [21] used MARS, SVM, multiple linear regression (MLR), and random forest (RF) for rainfall–runoff prediction in the Gola watershed, Uttarakhand. The performance of models was assessed using numerical indices (i.e., R^2, RMSE, NSE, and percent bias) along with graphical charting (i.e., scatter plots, relative error plots, violin plots, line diagrams, and Taylor diagrams). In all case studies, the RF outperformed the other models in terms of daily runoff forecasting in both the training and testing phases.

After reviewing the literature, it is observed that many machine learning algorithms have been employed to mimic rainfall–runoff simulation. However, there is a lack of a comprehensive comparison of machine learning algorithms. In this regard, the main goal of this research is to simulate the rainfall–runoff phenomenon using standalone and hybrid machine learning models. ANN, LSSVM, K-nearest neighbor (KNN), M5 model, RF, MARS, and multivariate nonlinear regression (MNLR) are examples of conventional models. Meanwhile, hybrid models refer to LSSVM and KNN coupled with gorilla troop optimizer (GTO). Additionally, this study introduces a new combination of the feature selection method, principal component analysis (PCA), and empirical mode decomposition (EMD). The developed models are evaluated using MAE, RMSE, relative RMSE (RRMSE), person correlation coefficient (R), NSE, and Kling Gupta efficiency (KGE). The proposed models are applied to rainfall and runoff dataset records in Wadi Ouahrane, Algeria, because of the complex and nonlinear nature of runoff precipitation in this basin.

2. Materials and Methods

2.1. Multivariate Empirical Mode Decomposition (EMD)

EMD was introduced to decompose a signal of original data into finite and small oscillating modes. The oscillating methods are known as intrinsic mode functions (IMFs) and should meet the following criteria [22]:

1. Over the entire signal length, the number of zero-crossings and the number of local maxima and minima are either equal to or at least differ by one.
2. The average upper and lower envelopes calculated by local maxima and minima should be equal to zero.

EMD does not need to select the base function, and it is an alternative to signal decomposition methods such as the Fourier transform and the wavelet transform. In this process, the IMFs are obtained from the signal until they satisfy the above-mentioned criteria. The sifting method for extracting IMFs includes the following steps:

Step 1: Determine all the extreme points of the given signal.

Step 2: Use a cubic spline to fit the upper and lower envelopes of the signal.

Step 3: Calculate the average upper and lower envelopes using Equation (1) [23].

$$M(t) = \frac{e_{upper}(t) - e_{lower}(t)}{2} \tag{1}$$

Step 4: Subtract the average from the data to create the IMF candidate using Equation (2).

$$h(t) = y(t) - M(t) \tag{2}$$

Step 5: If $h(t)$ satisfies the two criteria for IMFs, it is considered the first IMF; otherwise, $y(t)$ is replaced with $h(t)$, and we go to step 1.

Step 6: The residual is regarded as new data, and steps 1–5 are applied. This process continues until the number of residues is constant or their trend is obtained. EMD is a simple and efficient method for the decomposition of signals. It is appropriate for identifying immediate frequency changes, especially for nonstationary signals.

2.2. Principle Component Analysis (PCA)

PCA is used for data pre-processing to identify the correlation among candidate factors. It converts the input variables into uncorrelated derived variables called principal components (PCs). Sums of PC variances are equal for the original and uncorrelated derived variables. PCs can be obtained using a linear function in Equation (3):

$$PC_i = \sum_{j=1}^{N} a_{i,j} X_j \tag{3}$$

where X_j is the original variable, j is the index of the input variable, and i is the index of PC; $a_{i,j}$ and PC_i are the eigenvalues and eigenvectors of the covariance matrix, respectively. The present study employs PCA because of the large size of the input dataset.

2.3. Multivariate Nonlinear Regression (MNLR)

MNLR is a nonlinear regression that estimates the nonlinear relationship between multiple inputs and output data. Equation (4) can be used for estimating the target variable.

$$R_{Output} = \sum_{i=1}^{N} W_i X_i^2 + \sum_{i=1}^{N} W_i X_i + b \tag{4}$$

where W and b are the weight and bias parameters, respectively.

2.4. Artificial Neural Networks (ANNs)

ANN is a machine learning algorithm that solves linear or nonlinear regression and classification problems. It processes input and output data in a multi-layer network to find the relationship between variables. It consists of one input layer, one or multiple hidden layers, and one output layer, in which each layer comprises one or several neurons. Neurons are simple mathematical models of biological neurons. In the hidden layer, the weighted summation of back layer neurons is imposed on one stimulation function, and the stimulation function generates one output signal, which is the input of the subsequent layer neurons.

2.5. K-Nearest Neighbor (KNN)

KNN is a nonparametric machine learning algorithm that solves regression and classification problems without presuppositions about training data distribution. In this algorithm, training data are considered neighbor points. The inverse Euclidean distance between the testing data and neighbor points is regarded as the weight of these points. The shorter the Euclidean distance, the greater the weight. The neighbor points are sorted based on their weights, and the K neighbor points with the highest weights are selected. Then, the KNN computes the output of each input dataset using the weighted average of the K neighbor (Equation (5)) [24]:

$$R_{Output,i} = \frac{\sum_{j=1}^{K} W_j R_j}{\sum_{j=1}^{N} W_j} \tag{5}$$

where R_j is the jth observed runoff in the training period, $R_{output,i}$ is the ith estimated runoff, and W_j is the jth weight of the neighbor that can be calculated in Equation (6):

$$W_j = \frac{1}{\|X - X_j\|} \tag{6}$$

where X and X_j are the testing and training input data, respectively. Figure 1 shows the KNN scheme for modeling runoff.

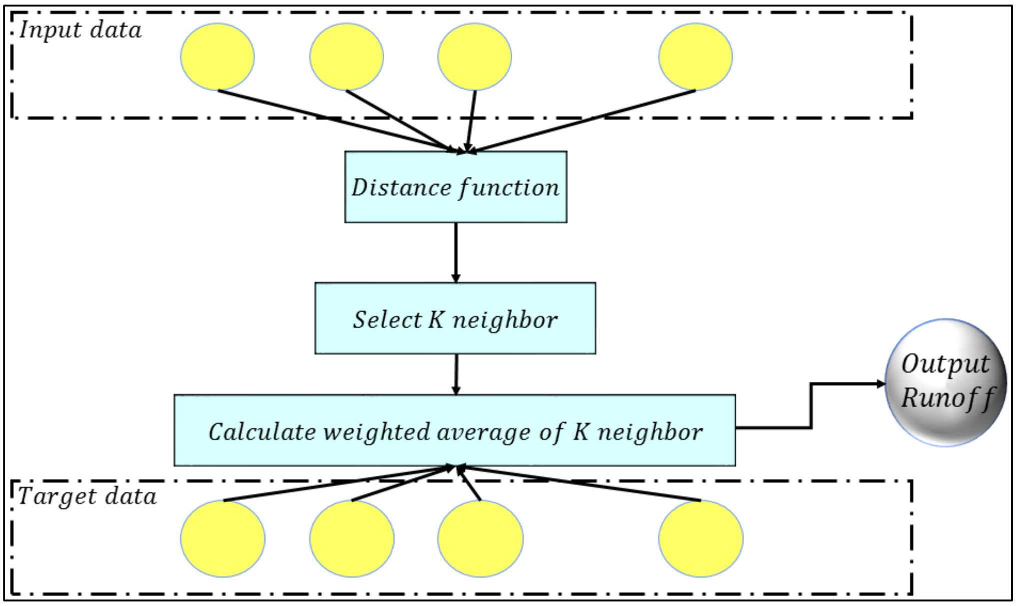


Figure 1. Structure of KNN.

2.6. Multivariate Adaptive Regression Spline (MARS)

MARS is a nonparametric and nonlinear machine learning algorithm for solving various regression and classification problems. MARS divides the original dataset into multiple sub-datasets. Then, for each sub-dataset, the target variable is fit using a spline regression. The formulation for this process is given by:

$$R_{Output,i} = b + \sum_{j=1}^{N} \beta_j h_j(X_i) \tag{7}$$

where b is the bias parameter, β is a constant coefficient, h is the basis function, and N is the number of basis functions [25].

2.7. M5 Model Tree (M5)

M5 is one of the tree-based machine learning algorithms used for modeling continuous variables. Its structure resembles a tree that consists of nodes, branches, and leaves. It splits the feature space into subsets, and a linear regression is fitted to the target variables of each subset. This process includes two steps: (1) growing the tree using input data and establishing linear regression at the end of each leaf, and (2) pruning extra branches to avoid overfitting. The splitting criterion is the maximum reduction in standard deviation, and it is calculated as follows [26]:

$$SDR = sd(S) + \sum_{i=1}^{N} \frac{S_i}{S} sd(S_i) \quad (8)$$

where S is a subset in the parent node, S_i is a subset in the child node, and sd is the standard deviation for the input data.

2.8. Least Square Support Vector Machine (LSSVM)

The LSSVM is a modified version of the standard SVM. Unlike SVM, LSSVM employs linear equations instead of quadric programming and modifies SVM's computation time efficiency and accuracy. LSSVM uses the following equation to estimate the output:

$$R_{Output,i} = \sum_{i=1}^{N} H(X_i,\ X)\alpha_i + b \quad (9)$$

where α and b are lagrangian coefficients and bias, respectively. H is a kernel function that maps the nonlinear relation between input and output variables in low and high-dimensional feature space. This helps LSSVM solve the nonlinear problems in linear form. The linear, polynomial, sigmoid, and RBF are different types of kernel functions. However, the RBF is the most accurate kernel function that has been used in many studies. The RBF kernel functions are estimated as follows [27]:

$$H(X,\ X_i) = \exp\left(\frac{-\|X - X_i\|^2}{2\sigma^2}\right) \quad (10)$$

where σ represents the width of the kernel function. The main parameters of LSSVM are the penalty coefficient $(gamma)$ and σ, in which $gamma$ is used for computing α and b. The LSSVM scheme, including one input, hidden input, and final output, is demonstrated in Figure 2.

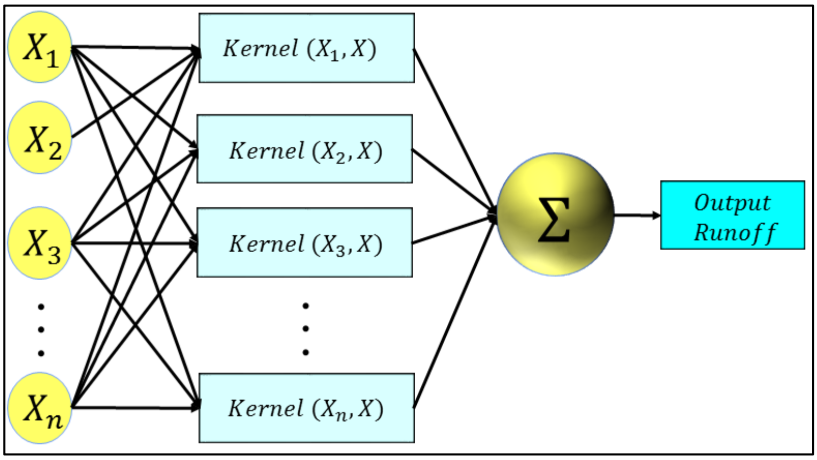


Figure 2. Scheme of the LSSVM structure.

2.9. Random Forest Regression (RF)

RF is one of the ensemble machine learning algorithms that solves decision trees' overfitting and instability problems. First, n random subsample from the original data is created. Then, for each subsample, one tree model is fitted, and RF integrates the generated results of all n trees into the outcome. In the present study, the M5 is considered an RF tree. For more information about RF, please see [28].

2.10. Gorilla Troop Optimizer (GTO)

GTO is based on the collaborative behavior of gorillas. This algorithm mimics five strategies of gorillas, including migration to unknown regions, migration to other gorillas, migration to other known locations, following the silverback, and competition for adult females [29]. The first three strategies are for exploration, and the remaining ones are for exploitation. Each artificial gorilla is considered one optimization problem solution, and the best gorilla in each iteration is regarded as a silverback. When $rand < p$, the first strategy of moving to an unknown region is selected. However, $rand < 0.5$ implies that the gorilla moves toward other gorillas, and if $rand > 0.5$, the gorilla shall migrate to known locations. The three exploration strategies are given by [30]:

$$GX_i^{iter+1} = \begin{cases} lb + rand_1 \times (ub - lb),\ rand < p \\ (rand_2 - C) \times X_r + L \times H\ rand \geq 0.5 \\ X_i^{iter} - L \times \left(L \times \left(X_i^{iter} - GX_r^{iter-1}\right) + rand_3 \times \left(X_t - GX_r^{iter-1}\right)\right),\ rand < 0.5 \end{cases} \tag{11}$$

In this context, GX_i^{iter+1} is a new candidate position vector of gorilla, X_i^{iter} is the current position of gorilla, $rand_1$, $rand_2$, and $rand_3$ are random numbers in the range between 0 and 1. The p variable represents the probability of migration to unknown regions. X_r and GT_r are members of artificial gorillas that are randomly selected from the whole population. ub and lb are the upper and lower bounds of decision variables. C, L, and H can be calculated in Equations (12) and (13):

$$C = F \times \left(1 - \frac{iter}{Max\_Iter}\right) \tag{12}$$

$$F = \cos(2 \times rand_4) + 1 \tag{13}$$

$$L = C \times l \tag{14}$$

$$H = Z \times X^{iter} \tag{15}$$

$$Z = [-C,\ C] \tag{16}$$

where $iter$ refers to the current iteration, Max_Iter is the maximum number of iterations, F is computed using Equation (8), cos is a cosine function, and $rand_4$ is a random number in the range of [0, 1]. L is calculated using Equation (9), l is a random number ranging from 0 to 1, H is computed using Equation (11), and Z is a random value in the range between –C and C. The fitness function of all GX is evaluated at the end of an exploration phase, and if the fitness function of GX^{iter} is less than X^{iter}, the GX^{iter} is used as X^{iter}. The best solution at this stage is the silverback gorilla.

GTO uses the silverback and competition for adult female strategies in the exploitation phase. Silverback is the head of the group that makes decisions and guides other gorillas to food sources. The young gorillas become mature and compete with other gorillas to select adult female gorillas. As per the below equation, these two strategies are mathematically modeled. If $C \geq W$, the first strategy is followed; otherwise, the second strategy is selected. W can be set before running GTO in Equation (17).

$$GX_i^{iter+1} = \begin{cases} L \times M \times \left(X_i^t - X_{silverback}\right),\ C \geq W \\ X_{silverback} - \left(X_{silverback} \times Q - X_i^t \times Q\right) \times A\ C < W \end{cases} \tag{17}$$

where $X_{silverback}$ is the position vector of the silverback, Q is the impact force, and A is the degree of violence in case of conflicts. Meanwhile, M, Q, and A are computed using the following Equations:

$$M = \left(\left|\frac{1}{N}\sum_{i=1}^{N} GX_i^t\right|^g\right)^{\frac{1}{g}} \tag{18}$$

$$Q = 2rand_5 - 1 \tag{19}$$

$$A = \beta \times E \tag{20}$$

where GX_i^{iter} is the current position of the candidate gorilla's vector, N is the number of gorillas, $rand_5$ represents a random number between 0 and 1, and E simulates the violence effect on the solution's dimensions. The values of g and E are calculated as follows:

$$g = 2^L \tag{21}$$

$$E = \begin{cases} N_1, \ rand \geq 0.5 \\ N_2 \ rand < 0.5 \end{cases} \tag{22}$$

where N_1 is a normal value with a normal distribution in the problem's dimensions and N_2 is a random number with a normal distribution.

2.11. Hybrid of LSSVM and KNN with Gorilla Troop Optimizer

Both LSSVM and KNN have essential parameters that should be selected before maneuvering them. However, choosing these parameters is still challenging for scientific societies. Using nature-based optimization algorithms can be an excellent solution to this challenge. Hence, in the present study, the GTO algorithm, as an efficient optimization algorithm, is used to determine the optimal LSSVM and GTO values. In this regard, the two-hybrid algorithms called KNN–GTO and LSSVM–GTO are defined. In KNN–GTO, the numbers of neighbors and input weight vectors are considered decision variables, whereas in LSSVM–GTO, penalty coefficients $(gamma)$ and σ are decision variables. For finding the optimal parameters of KNN and LSSVM, GTO solves the following fitness function (Equation (23)) in a pre-defined maximum number of iterations:

$$fitness\ function = \frac{\sum_{i=1}^{N} \left(R_{output,i} - R_{observed,i}\right)^2}{N} \tag{23}$$

where $R_{observed,\ i}$ is the observed runoff. The pseudocodes of KNN–GTO and LSSVM–GTO are presented in Algorithm 1.

Algorithm 1. KNN–GTO and LSSVM–GTO

```
Initialize parameters of GTO
Load inputs and target variables dataset
Generate the initial population of GTO
Train and test KNN and LSSVM for each artificial gorilla
Calculate the fitness function (MSE) for each artificial gorilla
iter: =1
while iter < Max_Iter do
    Update the position of an artificial gorilla using Equations (10)–(19)
    iter: = iter + 1
end while
Return the best solution (optimal W and K for KNN, and gamma and σ for LSSVM)
```

2.12. Assessment Criteria

In this study, MAE, RMSE, RRMSE, R, NSE, and KGE metrics are used for assessing the performance of rainfall–runoff models using the following Equations [31,32]:

$$MAE = \frac{\sum_{i=1}^{N} \left|R_{output,i} - R_{observed,i}\right|}{N} \tag{24}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(R_{output,i} - R_{observed,i}\right)^2}{N}} \quad (25)$$

$$RRMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(R_{output,i} - R_{observed,i}\right)^2}{N * std\left(R_{observed}\right)^2}} \quad (26)$$

$$R = \frac{\sum_{i=1}^{N}\left(R_{output,i} - \overline{R}_{output,i}\right)\left(R_{observe,i} - \overline{R}_{observe,i}\right)}{\sum_{i=1}^{N}\left(R_{output,i} - \overline{R}_{output,i}\right)^2 \sum_{i=1}^{N}\left(R_{observe,i} - \overline{R}_{observe,i}\right)^2} \quad (27)$$

$$NSE = 1 - \frac{\sum_{i=1}^{N}\left(R_{observe,i} - R_{output,i}\right)^2}{\sum_{i=1}^{N}\left(R_{observe,i} - \overline{R}_{observe,i}\right)^2} \quad (28)$$

$$KGE = 1 - \sqrt{(R-1)^2 + \left(\frac{\overline{R}_{observe}}{\overline{R}_{output}} - 1\right)^2 + \left(\frac{std\left(R_{observe}\right)}{std\left(R_{output}\right)} - 1\right)^2} \quad (29)$$

where $R_{output,i}$, $R_{observe,i}$, $\overline{R}_{output}$, $\overline{R}_{observe}$, $std\left(R_{output}\right)$, $std\left(R_{observe}\right)$, and N are the output runoff, observed runoff, average output runoff, average observed runoff, standard deviation of output runoff, standard deviation of observed runoff, and number of data, respectively. The desired values of MAE and RMSE are zeros, and their undesired values are $+\infty$. The desired values of RRMSE are in the range of [0, 0.5]. The R-value lies between -1 and 1, and R values close to 1 indicate good model performance. NSE = 1 denotes a perfect fit between the model and the data. KGE values range between $-\infty$ and 1, and values close to one indicate better model performance.

3. Case Study and Data Description

The study area is the Wadi Ouahrane basin in northern Algeria, which is located between 36°00′ N–36°24′ N and 01°00′ E–01°3′ E. This 270 km^2 region is a section of the Wadi Cheliff basin (Figure 3). The research area was mapped using a digital elevation model (12.5 m horizontal resolution), which displays a maximum altitude of 991 m and a minimum altitude of 165 m. A little, few kilometers long tributary of Wadi Cheliff is called Wadi Ouahrane. The flow of water in this basin is controlled by six pluviometric stations. The Wadi Ouahrane basin is constrained by the Wadi Allala basin to the north, the Wadi Sly basin to the south, the Wadi Fodda basin to the east, and the Wadi Ras basin to the west. With an average interannual rainfall of 333 mm from 1972 to 2018, evapotranspiration (ET) is 1050 mm, and the mean annual flow is equal to 0.472 m^3/s; this basin has a Mediterranean climate. The yearly average temperature is 18 Celsius. The monthly rainfall datasets were obtained at six stations between 1972 and 2018, and these dataset records are used in this study. The meteorological information was given by the National Meteorological Organization and the National Water Resources Agency of Algeria.

The correlation plot for the input and target variables is shown in Figure 4. In this figure, positive correlation shows a direct relationship between inputs and targets, negative correlation shows the inverse relationship, and close to zero correlation indicates no relation between inputs and targets. The maximum and minimum correlation between input and target variables in Figure 4 are related to R_S1 and T_{mean}, respectively. However, the correlation between inputs and targets is not close to 1 or -1. Also, the statistical criteria for input and target variables are presented in Table 1. According to this table, although the coefficient of variation in runoff data is lower than the inputs, its skewness coefficient is significantly higher than the inputs. Therefore, the runoff data studied do not follow the normal distribution and have high dispersion. These observations prove the nonlinear runoff production in this basin. Consequently, powerful nonlinear methods are expected to be needed for rainfall–runoff modeling in this basin.

Table 1. Statistical criteria for runoff modeling.

Statistics	Q (m^3/s)	S1	S2	S3	S4	S5	S6	T_{min} (°C)	T_{mean} (°C)	T_{max} (°C)	RH_{mean} (%)	WS (m/s)
		Rainfall (mm/Month)										
Mean	0.47	30.29	40.57	27.81	32.48	35.44	33.96	12.34	25.8	28.01	50.38	2.58
Standard deviation	1.54	32.15	48.01	30.3	34.2	38.44	34.63	6.09	7.07	9.2	26.63	0.71
Minimum	0	0	0	0	0	0	0	−1.5	0	0	0	0.6
Maximum	18.1	167.6	336.4	156.3	175.05	265.2	172.3	24.7	51.83	96.27	82.5	4.9
Coefficient of variation	0.31	0.94	0.85	0.92	0.95	0.92	0.98	2.03	2.69	2.8	1.89	3.63
Skewness coefficient	6.82	1.28	1.81	1.42	1.29	1.72	1.23	0.17	0.36	0.92	−1.09	−0.12

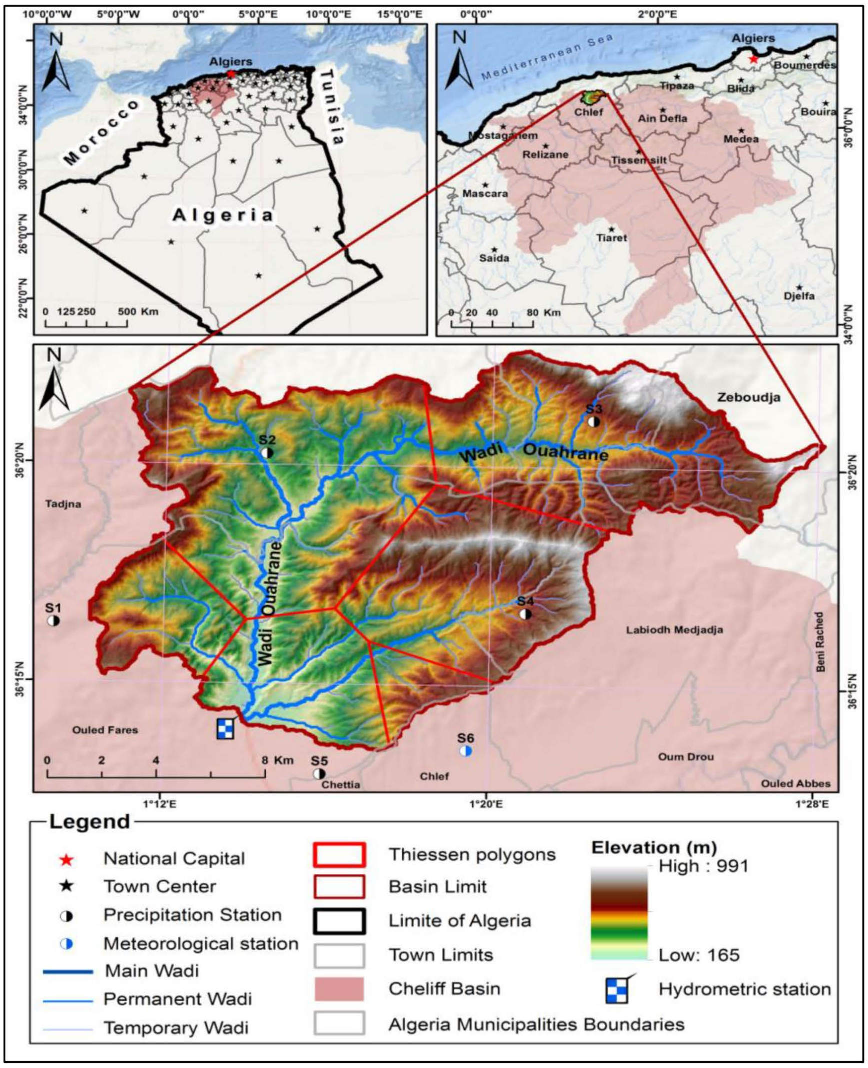


Figure 3. Map of the study area.

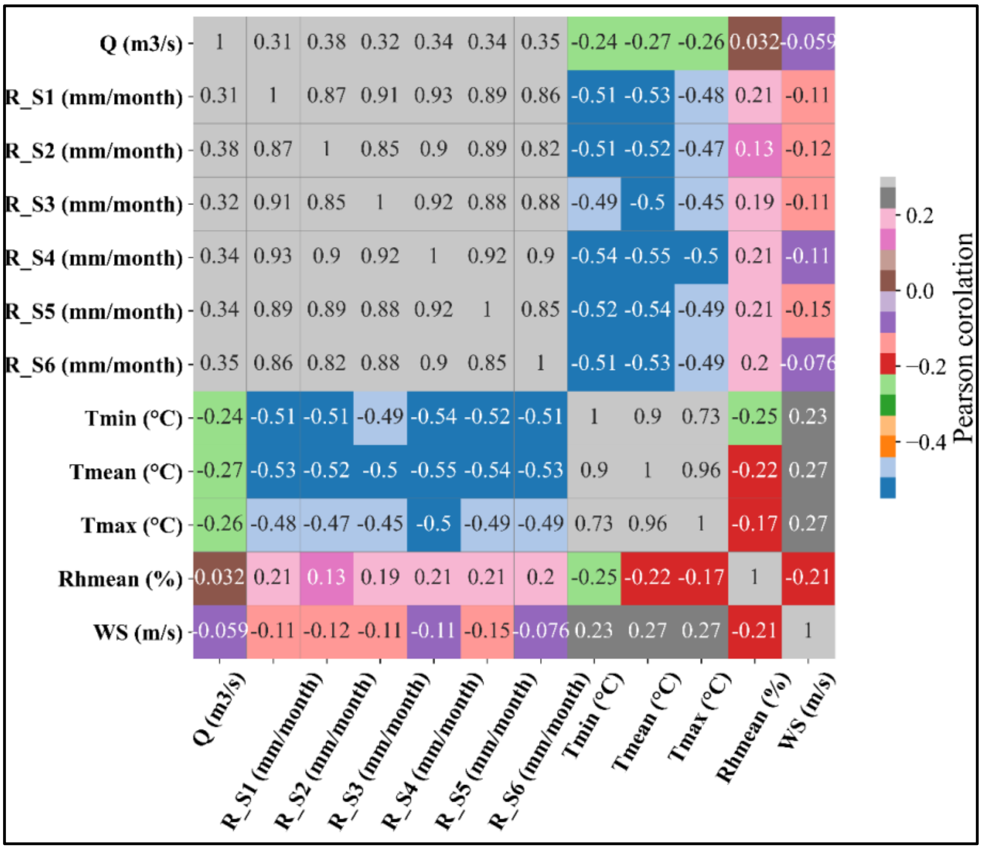


Figure 4. Correlation plot for the input and target variables.

4. Presented Framework for Modeling Rainfall–Runoff

The present study introduces a framework based on a combination of the feature selection method, PCA, EMD, and hybrids of KNN and LSSVM with GTO. In this framework, the most important inputs are selected using feature selection, and then the dataset is randomly divided into training and testing periods. The pseudocode of the applied feature selection method is illustrated in Algorithm 2. This feature selection method selects lagged inputs with a higher correlation with the target data.

Algorithm 2 feature selection

```
Load input data and target data
Apply lag times to input data
while i < number of input features do
    Calculate the Pearson correlation coefficient (R) between the feature and target data.
    If R < threshold of R
        Remove feature from the input data
    end if
    i: = i + 1
end while
Apply PCA to the remaining input data
Return the final inputs list
```

After feature selection, the size of the selected feature can be considerable; therefore, the PCA is used for dimension reduction. Then, the prepared dataset is used to apply the KNN–GTO and LSSVM–GTO models to simulate the rainfall–runoff phenomenon. Finally, the best of the results are selected according to different evaluation criteria. Furthermore, the results of the presented framework are compared with those of other machine learning algorithms, including MLR, KNN, ANN, M5, MARS, LSSVM, and RF, to validate the performance of the introduced framework. Figure 5 shows the scheme of the employed framework.

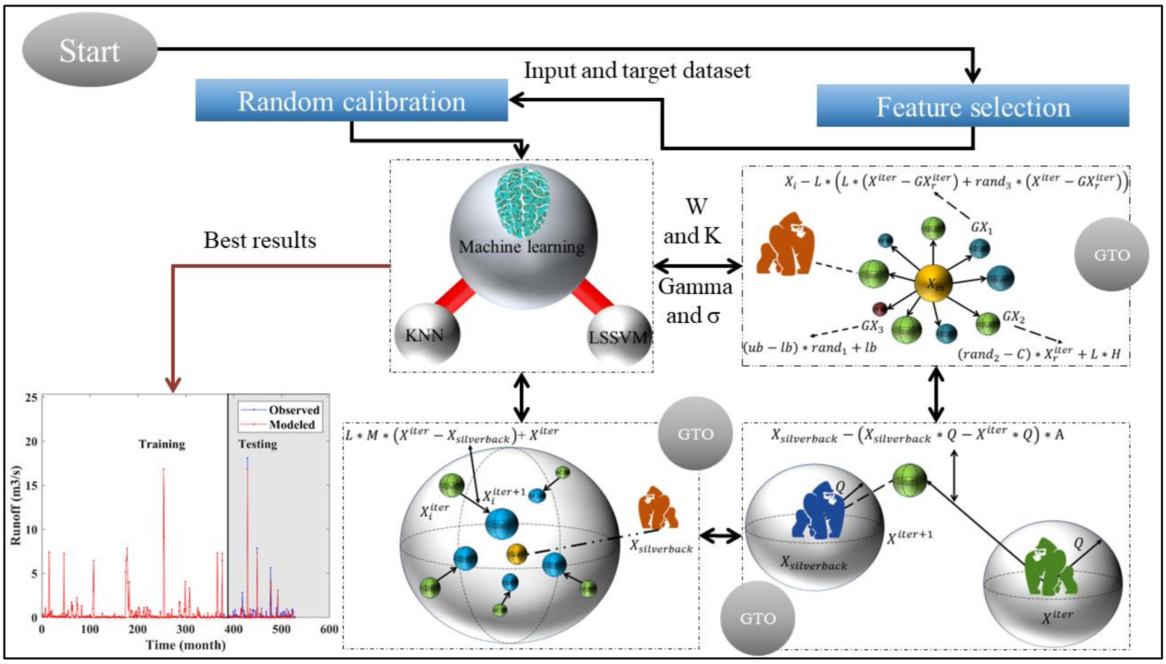


Figure 5. Scheme of the presented framework.

5. Results and Discussion

This study defines five scenarios for rainfall–runoff modeling (Table 2). In the first scenario, rainfall in six sections (T_{min}, T_{mean}, T_{max}, Rh_mean, and SW) is considered an input. The second scenario resembles the first scenario, with the difference that a 0 to 24-month lag time is imposed on input data and the R threshold is equal to 0.05. The third to fifth scenarios are the same as the second scenario in input data; however, the main difference is the application of IMF and the R threshold value of 0.1. The MaxNumIMF in the third to fifth scenarios equals 3, 4, and 5, respectively. Since the size of the input dataset in the third to fifth scenarios is large, the PCA is employed for dimension reduction.

Table 2. Characteristics of scenarios.

Scenarios	Inputs	Threshold of R	Pre-Processing	Post-Processing
1	R_1, R_2, R_3, R_4, R_5, R_6, T_{min}, T_{mean}, T_{max}, Rh_mean, SW	-	-	-
2	R_1, R_2, R_3, R_4, R_5, R_6, T_{min}, T_{mean}, T_{max}, Rh_mean, SW Lag = 0:24 month	0.05	-	-
3	R_1, R_2, R_3, R_4, R_5, R_6, T_{min}, T_{mean}, T_{max}, Rh_mean, SW Lag = 0:24 month	0.1	IMF (MaxNumIMF = 3)	PCA
4	R_1, R_2, R_3, R_4, R_5, R_6, T_{min}, T_{mean}, T_{max}, Rh_mean, SW Lag = 0:24 month	0.1	IMF (MaxNumIMF = 4)	PCA
5	R_1, R_2, R_3, R_4, R_5, R_6, T_{min}, T_{mean}, T_{max}, Rh_mean, SW Lag = 0:24 month	0.1	IMF (MaxNumIMF = 5)	PCA

The best parameters of the investigated algorithms are listed in Table 3. The grid search method estimates the parameters of ANN, LSSVM, M5, MARS, and RF. It is worth mentioning that MNLR does not have any parameters for implementation. The essential parameters of ANN are the number of neurons in the first and second layers. LSSVM and LSSVM–GTO can be implemented by defining gamma and sigma. The main essential parameters of M5 are min leaf size (minLSize) and split threshold (sThreshold), while MARS is developed by determining the maximum base function and model parameter (C). RF resembles the M5 tree, but it has another parameter called the number of trees (Num Tree). KNN and KNN–GTO are executed by selecting the K number of neighbors (K), but KNN–GTO has another main parameter, namely the weight of inputs (W).

Table 3. Optimal parameters of the investigated algorithms.

Scenarios	Algorithm	N1/N2	γ/σ	minLSize/sThreshold	mF/C	NumTree	K
1	ANN	1/5	-	-	-	-	-
	LSSVM	-	4.90/6.00	-	-	-	-
	M5	-	-	64/0.01	-	-	-
	MARS	-	-	-	5/4	-	-
	RF	-	-	4/0.05	-	100	-
	LSSVM–GTO	-	5.23/6.19	-	-	-	-
	KNN	-	-	-	-	-	13
	KNN–GTO	-	-	-	-	-	4
2	ANN	15/4	-	-	-	-	-
	LSSVM	-	10/5	-	-	-	-
	M5	-	-	64/0.01	-	-	-
	MARS	-	-	-	5/4	-	-
	RF	-	-	8/0.01	-	100	-
	LSSVM–GTO	-	100/8.16	-	-	-	-
	KNN	-	-	-	-	-	2
	KNN–GTO	-	-	-	-	-	2
3	ANN	10/7	-	-	-	-	-
	LSSVM	-	10/5	-	-	-	-
	M5	-	-	64/0.01	-	-	-
	MARS	-	-	-	5/4	-	-
	RF	-	-	32/0.1	-	100	-
	LSSVM–GTO	-	100/7.43	-	-	-	-
	KNN	-	-	-	-	-	3
	KNN–GTO	-	-	-	-	-	1
4	ANN	12/4	-	-	-	-	-
	LSSVM	-	10/5	-	-	-	-
	M5	-	-	64/0.1	-	-	-
	MARS	-	-	-	30/6	-	-
	RF	-	-	32/0.01	-	100	-
	LSSVM–GTO	-	1.38/2.33	-	-	-	-
	KNN	-	-	-	-	-	4
	KNN–GTO	-	-	-	-	-	1
5	ANN	7/7	-	-	-	-	-
	LSSVM	-	10/5	-	-	-	-
	M5	-	-	64/0.1	-	-	-
	MARS	-	-	-	30/4	-	-
	RF	-	-	8/0.01	-	100	-
	LSSVM–GTO	-	100/8.35	-	-	-	-
	KNN	-	-	-	-	-	5
	KNN–GTO	-	-	-	-	-	4

Figure 6 shows the weight of inputs (W) obtained by KNN–GTO. As seen, the importance of inputs is between 0 and 1, according to the base assumptions of KNN. The W in each scenario is different from that in another scenario, owing to the various amounts of input data in each scenario. Furthermore, the number of inputs in the first scenario is less than that in other scenarios. Therefore, it is expected that the accuracy of modeling rainfall–runoff will be lower in this scenario compared to other scenarios. Also, in the third, fourth, and fifth scenarios, the values of W are higher than in other scenarios, showing a greater correlation between these data and runoff data. The greater W value in the mentioned scenarios can lead to the high precision of KNN and KNN–GTO.

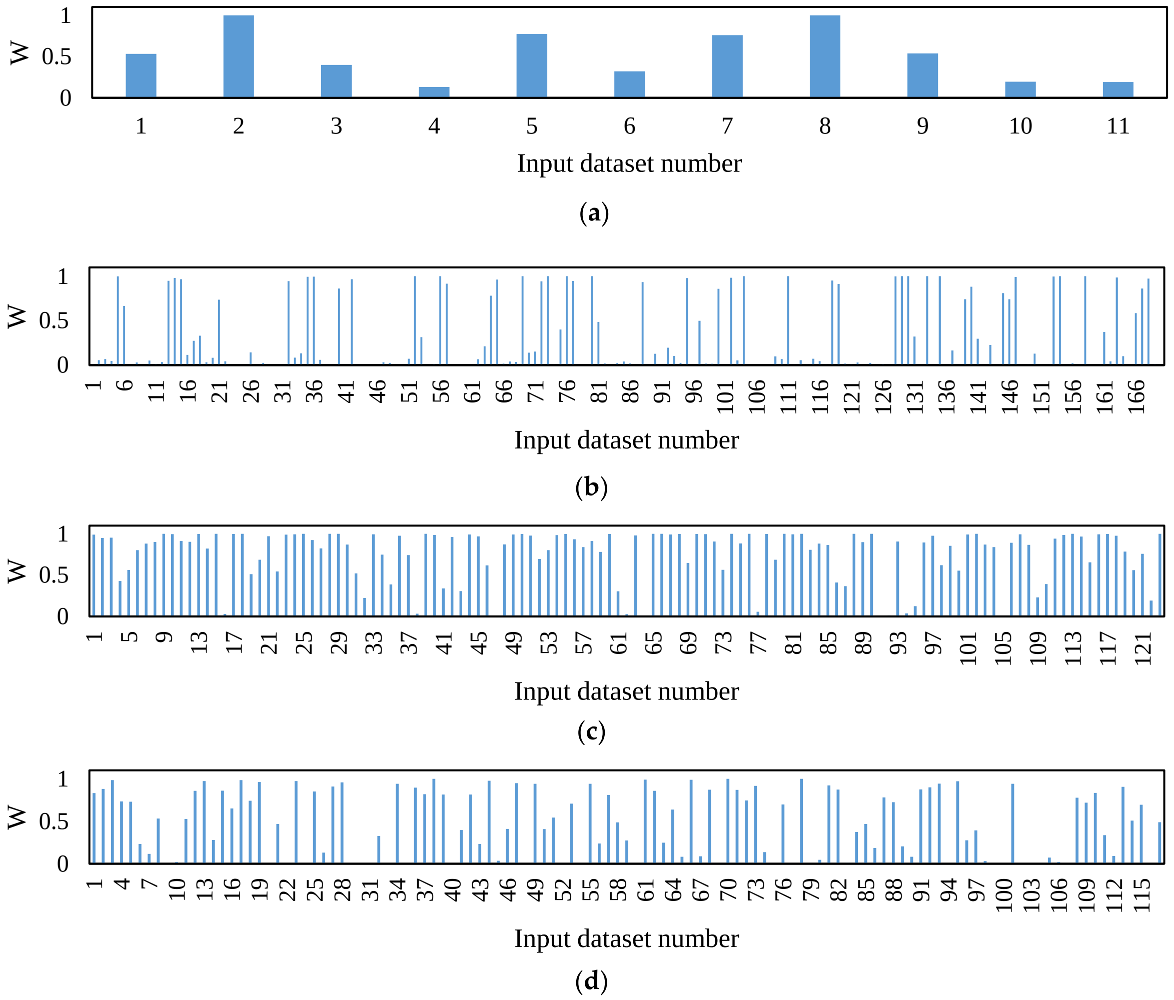


Figure 6. W values of input data for (**a**) scenario1, (**b**) scenario2, (**c**) scenario3, (**d**) scenario4.

Table 4 compares the accuracy of machine learning algorithms for rainfall–runoff modeling for the training period. According to this table, all algorithms have weak performance in the first scenario. This issue indicates the importance of selecting appropriate inputs and dataset processing. However, in other scenarios, other algorithms, such as ANN, LSSVM, KNN, LSSVM–GTO, and KNN–GTO, are trained with comparable accuracy. The best performance is associated with ANN in the third scenario and KNN–GTO in the fourth and fifth scenarios. For ANN, the MAE, RMSE, and RRMSE are equal to 0.000, while R, NSE, and KGE are equal to 1.0000. In addition, the metrics for KNN–GTO are specified to be 0.0001, 0.0016, 0.0011, 1.0000, 1.0000, and 0.9998, respectively.

Table 4. Results of rainfall–runoff modeling using machine learning algorithms for the training period.

Scenarios	Algorithm	MAE	RMSE	RRMSE	R	NSE	KGE
1	ANN	0.4540	1.3057	0.9052	0.4608	0.1786	−0.1042
	LSSVM	0.3175	0.7779	0.7240	0.7135	0.4745	0.3187
	M5	0.5356	1.4645	0.8855	0.4625	0.2139	0.0477
	MARS	0.5679	1.4487	0.8759	0.4804	0.2308	0.0717
	RF	0.3314	1.0234	0.6188	0.8354	0.6161	0.4567
	MNLR	0.4304	0.8965	0.8343	0.5497	0.3022	0.1695
	LSSVM–GTO	0.3174	0.7776	0.7237	0.7138	0.4749	0.3192
	KNN	0.5667	1.6160	0.9238	0.4109	0.1444	−0.1271
	KNN–GTO	0.5364	1.6365	0.9355	0.4277	0.1226	−0.1896
2	ANN	0.0209	0.0446	0.0345	0.9996	0.9988	0.9749
	LSSVM	0.1582	0.4317	0.3266	0.9827	0.8931	0.7108
	M5	0.4137	0.9160	0.7525	0.6574	0.4322	0.3368
	MARS	0.3545	0.8260	0.6380	0.7693	0.5918	0.5312
	RF	0.1968	0.7060	0.4956	0.9085	0.7537	0.6003
	MNLR	0.4526	0.6950	0.5709	0.8205	0.6731	0.6271
	LSSVM–GTO	0.0703	0.1859	0.1406	0.9972	0.9802	0.8773
	KNN	0.2498	0.8191	0.5750	0.8207	0.6685	0.5678
	KNN–GTO	0.0013	0.0127	0.0089	1.0000	0.9999	0.9969
3	ANN	0.0000	0.0000	0.0000	1.0000	1.0000	1.0000
	LSSVM	0.1268	0.3296	0.2015	0.9941	0.9593	0.8232
	M5	0.1856	0.7773	0.4796	0.8771	0.7694	0.7387
	MARS	0.4922	1.0055	0.8394	0.5418	0.2935	0.1580
	RF	0.2838	0.8688	0.5312	0.9236	0.7171	0.5305
	MNLR	0.4343	0.6578	0.5491	0.8353	0.6977	0.6557
	LSSVM–GTO	0.0529	0.1304	0.0797	0.9989	0.9936	0.9344
	KNN	0.3353	0.9785	0.5865	0.8206	0.6551	0.5454
	KNN–GTO	0.2839	0.8923	0.5348	0.8631	0.7132	0.5837
4	ANN	0.0277	0.0440	0.0300	0.9996	0.9991	0.9892
	LSSVM	0.1688	0.4213	0.2811	0.9874	0.9208	0.7525
	M5	0.3262	0.9896	0.6500	0.7592	0.5764	0.5127
	MARS	0.4541	0.7097	0.4662	0.8844	0.7821	0.7533
	RF	0.2547	0.8613	0.5068	0.9040	0.7424	0.5878
	MNLR	0.5617	0.9386	0.6262	0.7790	0.6068	0.5489
	LSSVM–GTO	0.0628	0.1525	0.1001	0.9981	0.9899	0.9184
	KNN	0.3389	0.9946	0.6533	0.7579	0.5721	0.4838
	KNN–GTO	0.0001	0.0016	0.0011	1.0000	1.0000	0.9998
5	ANN	0.0405	0.0693	0.0403	0.9994	0.9984	0.9498
	LSSVM	0.1523	0.3627	0.2499	0.9908	0.9374	0.7790
	M5	0.0795	0.4670	0.3217	0.9467	0.8962	0.8833
	MARS	0.5338	1.1197	0.6983	0.7149	0.5111	0.4340
	RF	0.2694	0.8038	0.4673	0.9604	0.7810	0.5769
	MNLR	0.4770	0.7890	0.5436	0.8389	0.7037	0.6627
	LSSVM–GTO	0.0588	0.1302	0.0897	0.9986	0.9919	0.9252
	KNN	0.2279	0.8323	0.4819	0.8875	0.7671	0.6455
	KNN–GTO	0.0001	0.0021	0.0012	1.0000	1.0000	0.9998

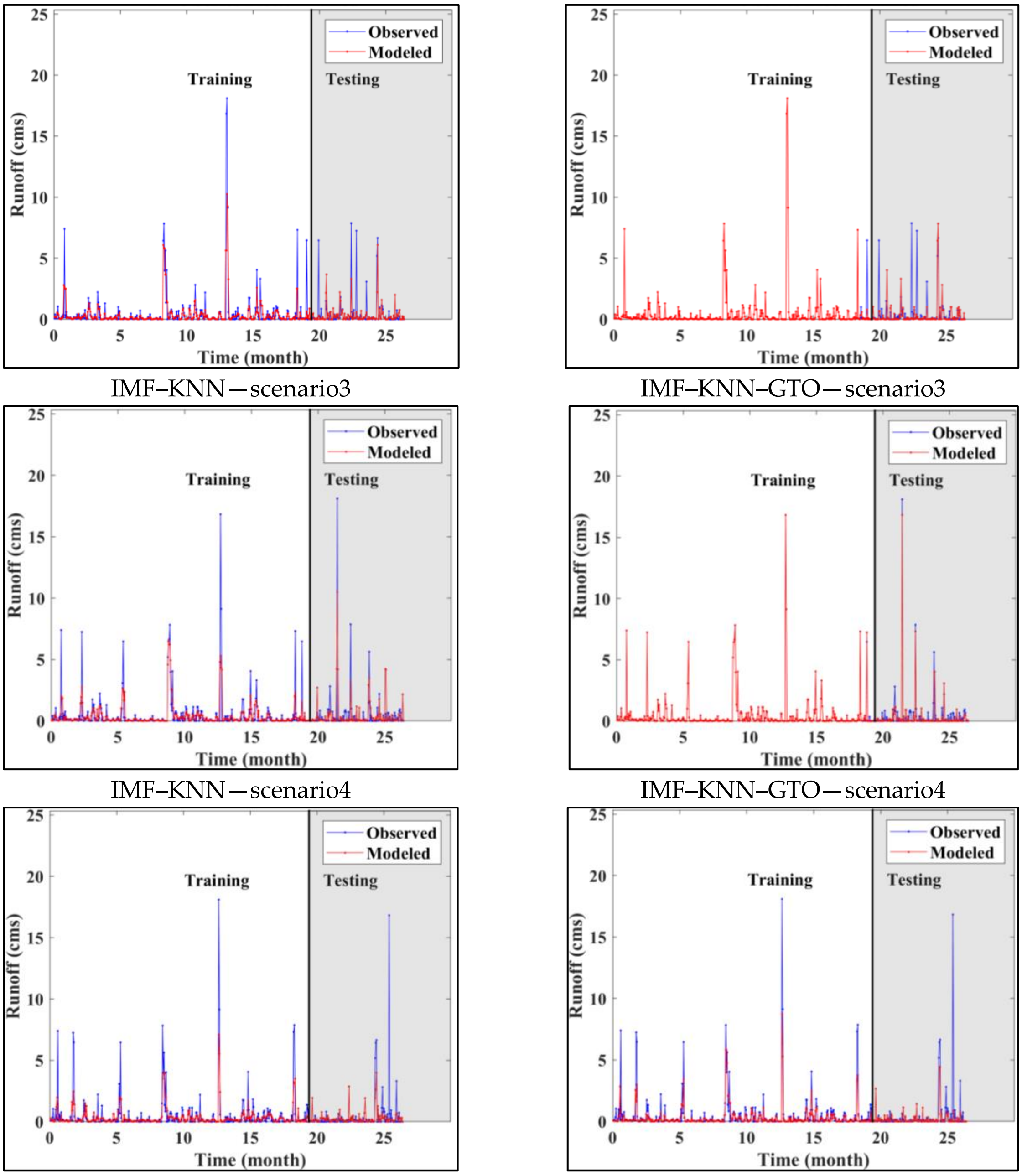
Table 5 compares the results of rainfall–runoff modeling by machine learning during the testing period. As seen in the first and second scenarios, machine learning algorithms produce low accuracy owing to poor training practices. Moreover, in the third scenario, the testing results are not as good as the training outcomes for ANN because of the overfitting problem of this algorithm. In contrast, KNN and KNN–GTO in the third, fourth, and fifth scenarios perform significantly better than the other algorithms. It can be noted that the best algorithm is KNN–GTO in the fourth scenario. The MAE, RMSE, RRMSE, R, NSE, and KGE for KNN–GTO are equal to 0.1640, 0.4741, 0.2978, 0.9607, 0.9108, and 0.7141, respectively. At the same time, MNLR in the first scenario is the worst algorithm, with

MAE, RMSE, RRMSE, R, NSE, and KGE equal to 0.8219, 2.2490, 0.9840, 0.2186, 0.0257, and −0.2600, respectively. On the contrary, KNN–GTO minimizes MAE, RMSE, and RRMSE by 80%, 79%, and 72% and maximizes R, NSE, and KGE by 77%, 112%, and 136% compared to the other algorithms. Moreover, Friedman test results show that KNN–GTO in the fourth, fifth, and third scenarios and KNN in the fourth and third scenarios are placed in the first to fifth ranking. However, MNLR and LSSVM in the first scenario have the worst ranking. Hence, in the following paragraphs, the accuracy of KNN and KNN–GTO is investigated.

Table 5. Results of rainfall–runoff modeling using machine learning algorithms for the testing period.

Scenarios	Algorithm	MAE	RMSE	RRMSE	R	NSE	KGE	Friedman Ranking
1	ANN	0.4827	1.5902	0.9017	0.4684	0.1820	−0.1069	35.3333
	LSSVM	0.7111	2.1998	0.9625	0.2941	0.0679	−0.2475	45.3333
	M5	0.5516	1.1938	0.9569	0.4099	0.0787	0.0760	35.6667
	MARS	0.5333	1.1759	0.9426	0.4172	0.1061	0.1119	33.3333
	RF	0.5239	1.2307	0.9865	0.3974	0.0208	0.1007	36.6667
	MNLR	0.8219	2.2490	0.9840	0.2186	0.0257	−0.2600	48.8333
	LSSVM–GTO	0.7111	2.1998	0.9625	0.2940	0.0679	−0.2474	45.3333
	KNN	0.3545	0.7886	0.9072	0.4470	0.1720	0.0744	27.6667
	KNN–GTO	0.3156	0.7537	0.8671	0.5006	0.2435	0.0502	24.8333
2	ANN	0.6039	1.8606	0.9227	0.4193	0.1432	0.0838	39.3333
	LSSVM	0.5587	1.6312	0.8277	0.6487	0.3105	0.0923	29.6667
	M5	0.4699	1.6769	0.7885	0.6787	0.3743	0.1388	23.3333
	MARS	0.4682	1.5109	0.7493	0.6625	0.4350	0.3069	17.8333
	RF	0.4893	1.5880	0.8834	0.4788	0.2147	−0.0066	34.0000
	MNLR	0.7810	1.7083	0.8033	0.5944	0.3507	0.1959	31.0000
	LSSVM–GTO	0.5491	1.5716	0.7975	0.6690	0.3599	0.1570	24.3333
	KNN	0.4925	1.6548	0.9205	0.4460	0.1472	−0.2137	37.5000
	KNN–GTO	0.3823	1.5340	0.8534	0.5365	0.2671	0.0273	29.6667
3	ANN	0.5885	1.1976	0.6946	0.7428	0.5144	0.5419	13.1667
	LSSVM	0.4661	0.9855	0.7543	0.6998	0.4274	0.2388	14.8333
	M5	0.4572	1.2876	0.9547	0.3579	0.0827	−0.1658	35.1667
	MARS	0.5875	1.6682	0.7769	0.6411	0.3925	0.3570	22.6667
	RF	0.5245	1.1745	0.8989	0.4489	0.1869	−0.0441	33.0000
	MNLR	0.7334	1.6685	0.7771	0.6350	0.3922	0.2339	26.3333
	LSSVM–GTO	0.4675	0.9404	0.7197	0.7167	0.4787	0.2911	13.0000
	KNN	0.2897	0.7242	0.6031	0.8053	0.6340	0.5264	5.0000
	KNN–GTO	0.2354	0.6521	0.5431	0.8746	0.7032	0.5316	3.1667
4	ANN	0.4257	1.2139	0.7069	0.7414	0.4971	0.5129	10.5000
	LSSVM	0.4576	1.3281	0.8070	0.6323	0.3446	0.1247	23.0000
	M5	0.5240	1.5408	0.9678	0.3241	0.0574	−0.0401	40.1667
	MARS	0.5581	1.0767	0.6763	0.7356	0.5397	0.4471	12.5000
	RF	0.4124	0.8672	0.7951	0.6471	0.3638	0.1145	17.8333
	MNLR	0.6971	1.1739	0.7133	0.7004	0.4880	0.4188	16.0000
	LSSVM–GTO	0.5097	1.2447	0.7818	0.6646	0.3849	0.0786	22.5000
	KNN	0.3052	0.9475	0.5951	0.8469	0.6436	0.4863	6.1667
	KNN–GTO	0.1640	0.4741	0.2978	0.9607	0.9108	0.7141	1.3333
5	ANN	0.2895	0.7241	0.7124	0.7193	0.4892	0.3207	7.8333
	LSSVM	0.4999	1.4501	0.8313	0.5962	0.3046	0.0991	28.1667
	M5	0.4582	1.4096	0.8080	0.5973	0.3429	0.1317	23.8333
	MARS	0.7892	1.4660	1.0491	0.2993	−0.1077	0.0406	43.8333
	RF	0.4198	0.8954	0.8810	0.4904	0.2190	−0.0040	27.3333
	MNLR	0.7695	1.4628	0.8385	0.5659	0.2924	0.2631	30.3333
	LSSVM–GTO	0.4979	1.4151	0.8112	0.6039	0.3378	0.1526	25.1667
	KNN	0.3212	0.7952	0.8114	0.5972	0.3374	0.3034	18.3333
	KNN–GTO	0.1728	0.4016	0.4098	0.9162	0.8310	0.7187	1.6667

The time series of rainfall–runoff modeling by KNN and KNN–GTO in the third, fourth, and fifth scenarios are compared in Figure 7. In the third and fifth scenarios, KNN and KNN–GTO have weaknesses in estimating peak runoff data. However, in the fourth scenario, KNN performs reasonably, and KNN–GTO is significantly better than the others. Additionally, KNN–GTO has higher accuracy than KNN, proving the capability of GTO to optimize and improve the precision of KNN.



IMF–KNN—scenario3 IMF–KNN–GTO—scenario3

IMF–KNN—scenario4 IMF–KNN–GTO—scenario4

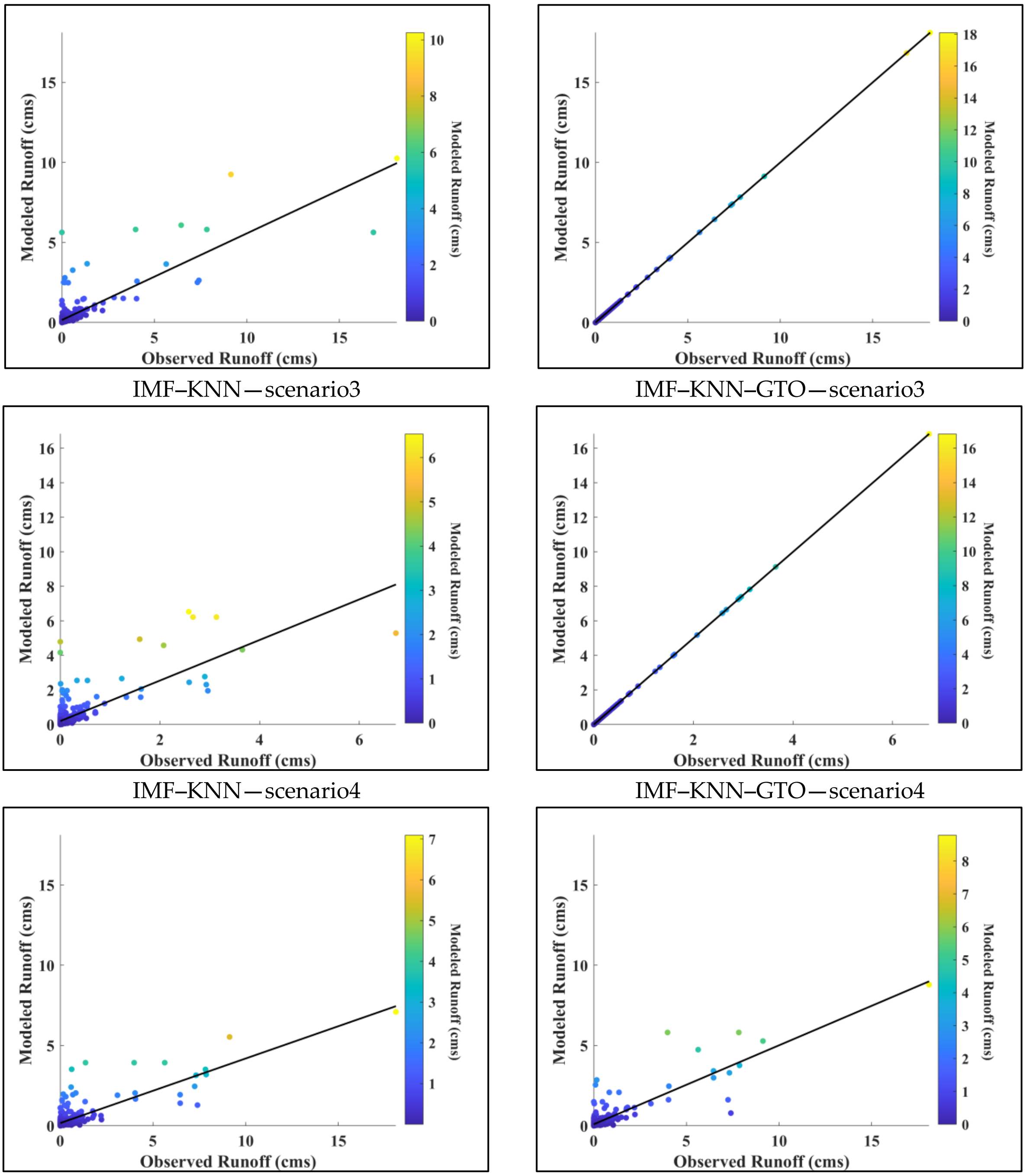
IMF–KNN—scenario5 IMF–KNN–GTO—scenario5

Figure 7. Time series plot of the best ML algorithms for modeling rainfall–runoff.

Figures 8 and 9 show the scatter plot representing the observed and modeled runoff with the line of a perfect fit at 45° during the training and testing periods. The results closer to the 45° line indicate more accurate machine learning algorithms.

KNN and KNN–GTO in the third, fourth, and fifth scenarios have closed results to the perfect fit line. In the testing period, KNN and KNN–GTO underestimated the runoff.

At the same time, the predicted outcomes by KNN–GTO in scenario 4 were close to the perfect line.



IMF–KNN—scenario3

IMF–KNN-GTO—scenario3

IMF–KNN—scenario4

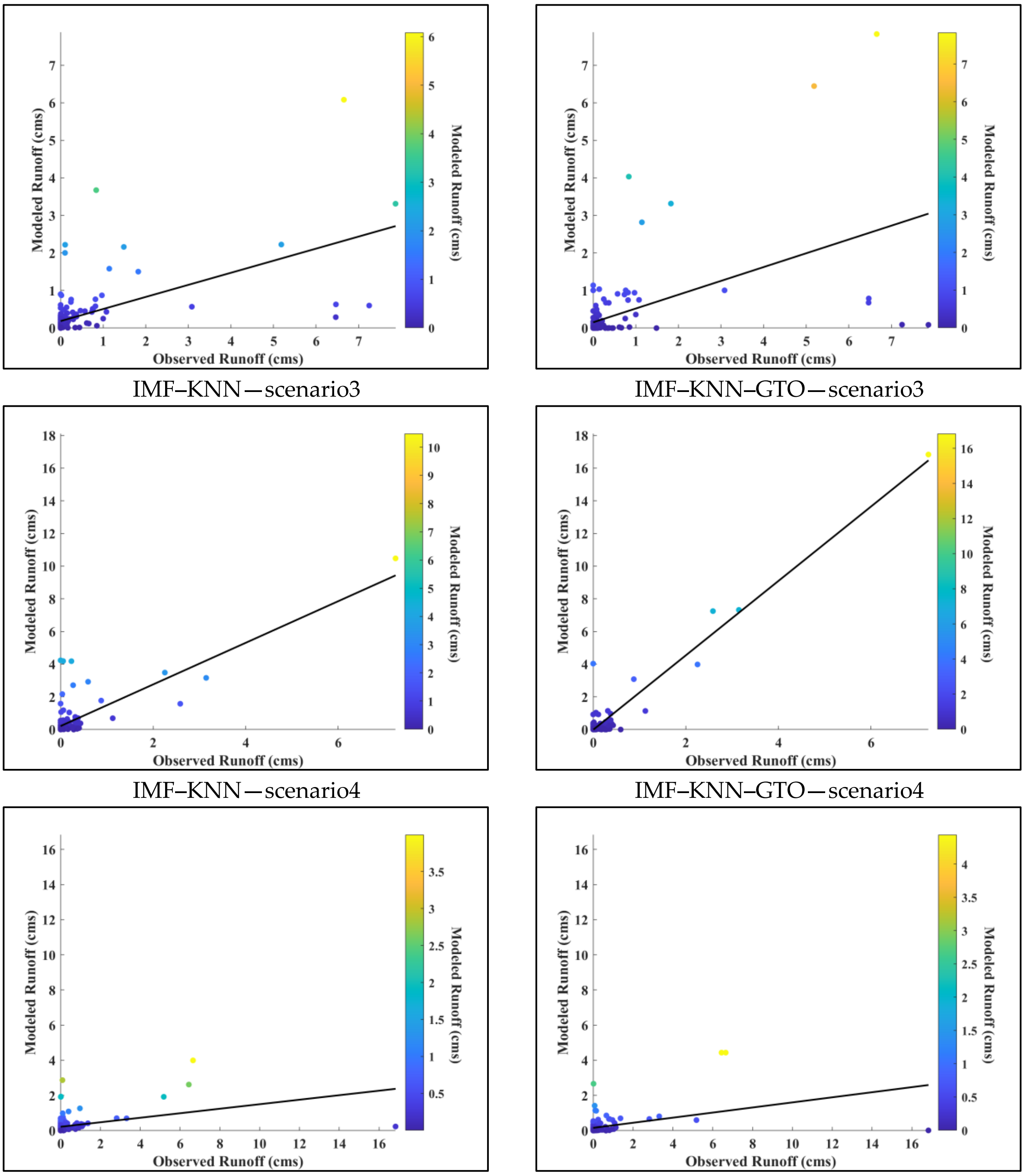
IMF–KNN-GTO—scenario4

IMF–KNN–scenario5

IMF–KNN-GTO—scenario5

Figure 8. Scatter plot of the best ML algorithms for modeling rainfall–runoff in the training period.

Model bias refers to the presence of systematic errors in a model that can cause it to consistently make incorrect predictions. Therefore, in this study, a PBias criterion is employed to analyze the bias of modeling in the best scenario, which means the fourth scenario. The estimated values for PBias are listed in Table 6. According to the results of this table, in the training period, M5, MARS, MNLR, and KNN_GTO had lower PBias. KNN and ANN have underestimated results. In contrast, LSSVM and LSSVM–GTO have overestimated results. During the testing period, the bias of all investigated algorithms increased. This is for using new data during the testing period. In this period, the less PBias is related to the MARS, and the maximum value of PBias is related to the RF. Considering all periods, MARS has fewer PBias, and LSSVM–GTO has more PBias. Moreover, KNN_GTO has reasonable PBias compared to other investigated algorithms. According to the study conducted by [33], the performance of the model for the PBias less than 10, between 10 and 15, and between 15 and 25 is very good, good, and fair, respectively. Hence, in terms of bias, according to all periods, it is very good.



IMF–KNN—scenario3

IMF–KNN–GTO—scenario3

IMF–KNN—scenario4

IMF–KNN–GTO—scenario4

IMF–KNN—scenario5

IMF–KNN–GTO—scenario5

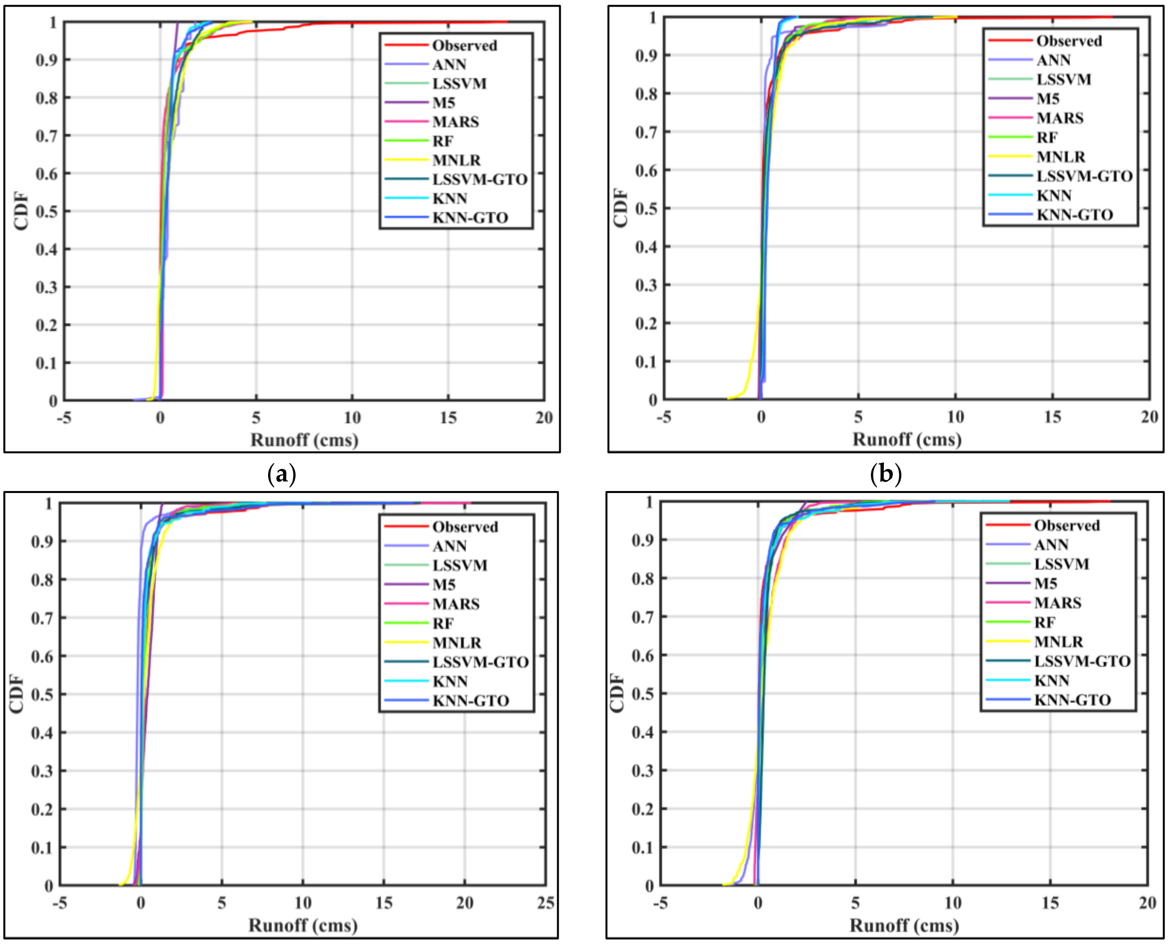
Figure 9. Scatter plot of the best ML algorithms for modeling rainfall–runoff in the testing period.

Table 6. Bias analysis in rainfall–runoff modeling using machine learning algorithms over training, testing, and all periods.

	ANN	LSSVM	M5	MARS	RF	MNLR	LSSVM–GTO	KNN	KNN–GTO
Training	−0.49	2.26	0.00	0.00	0.03	0.00	1.25	−4.68	0.02
Testing	18.23	22.21	25.41	9.00	36.54	−11.28	48.66	−6.36	−23.90
All	4.97	7.20	5.94	2.10	8.12	−2.79	12.33	−5.07	−5.57

Figure 10 shows the cumulative distribution function (CDF) for observed and modeled runoff under different scenarios. In this figure, the smaller the difference between observed and modeled CDF, the greater the accuracy. As seen, the maximum runoff modeling accuracy is related to the scenario3. In addition, in the mentioned scenario, the higher the accuracy is for KNN–GTO.

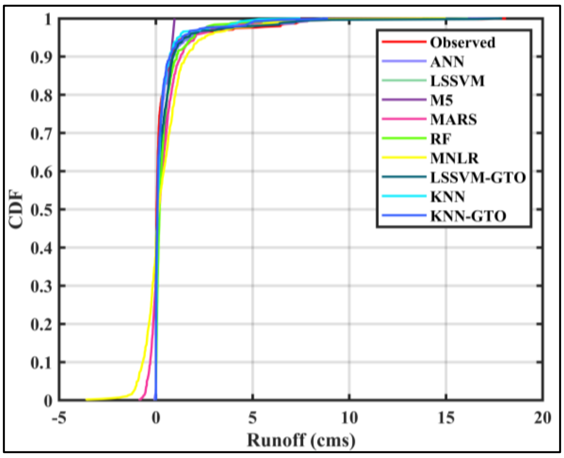
The convergence of GTO in optimizing KNN in the first to fourth scenarios is illustrated in Figure 11. In the fourth scenario, the minimum value of MSE is less than that in the other scenarios. The convergence speed in the fourth scenario is higher than that in the other scenarios. Therefore, using IMF improves the accuracy of rainfall–runoff modeling, and the optimal value of MaxNumIMF is equal to 4. The significant impact of using pre-processing and post-processing dataset methods and the use of time-lagged data show the effectiveness of input selection in modeling accuracy, which is accepted by the results of the third, fourth, and fifth scenarios in Tables 4 and 5. The role of dataset pre-processing and post-processing methods has been confirmed in other studies [34–36].



(**a**) (**b**)

(**c**) (**d**)

Figure 10. *Cont.*



(**e**)

Figure 10. Cumulative distribution function (CDF) for observed and modeled runoff under (**a**) scenario1, (**b**) scenario2, (**c**) scenario3, (**d**) scenatio4, and (**e**) scenatio5.

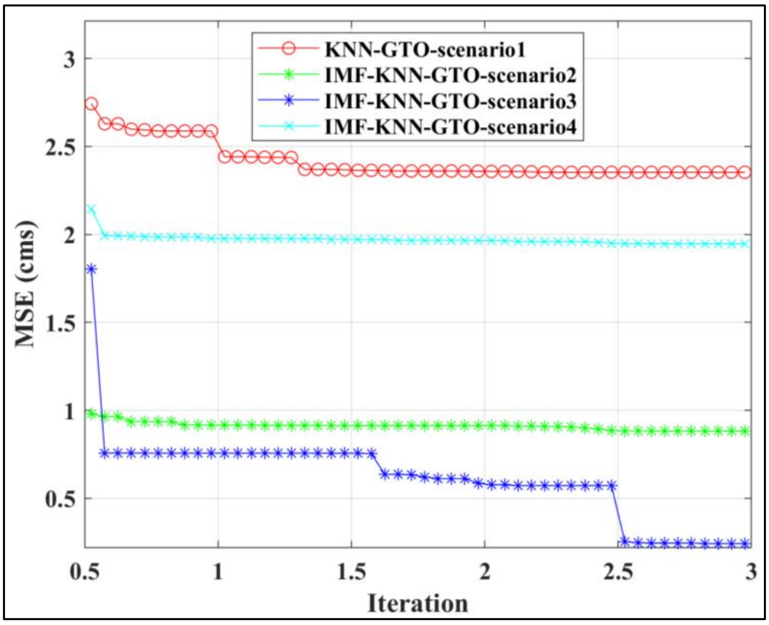


Figure 11. Convergence curve of KNN optimization.

Nevertheless, even in these scenarios, algorithms like LSSVM, M5, MARS, RF, and MNLR do not perform well, and ANN and LSSVM–GTO algorithms have moderate performance. The better accuracy of KNN and KNN–GTO algorithms is due to the kernel function, considering K nearest neighbor inputs. Also, the higher accuracy of KNN–GTO compared to KNN indicates the success of the GTO in finding the optimal parameters of the KNN algorithm.

6. Conclusions

In the present study, a new methodology was introduced for rainfall–runoff modeling. This methodology comprised dataset decomposition, feature selection, dataset reduction, and modeling by nine standalone and hybrid machine learning algorithms. The employed machine learning algorithms included neural network-based algorithms (ANNs), kernel-based algorithms (LSSVM and KNN), tree-based algorithms (M5 and RF), regression-based algorithms (MARS and MNLR), and hybrid algorithms (LSSVM–GTO and KNN–GTO). The reason for using this wide range of methods was the complex and nonlinear nature of runoff precipitation in Wadi Ouahrane, Algeria. Five scenarios were defined for selecting the input data. Results indicated that using EMD, feature selection, and PCA significantly improved the accuracy of rainfall–runoff modeling. KNN–GTO exhibited the best performance as it was associated with MAE, RMSE, RRMSE, R, NSE, and KGE of 0.1640, 0.4741, 0.2978, 0.9607, 0.9108, and 0.7141, respectively. It minimized MAE, RMSE, and RRMSE by 80%, 79%, and 72% and maximized R, NSE, and KGE by 77%, 112%, and 136% compared to the other algorithms. The worst algorithm was LSSVM without pre-processing data. A combination of data-processing methods and KNN–GTO performed accurately in estimating peak data. Comparing different scenarios showed that the machine learning algorithm had better performance when the maximum number of IMFs was equal to 4. Moreover, inputs with a correlation of greater than 0.1 were selected for rainfall–runoff modeling. In general, if high-quality data are available, there is no limitation to using the presented method (i.e., a combination of EMD, feature selection, PCA, and KNN–GTO) for predicting runoff and other hydrological parameters in other basins.

Author Contributions: M.V.A.: conceptualization, investigation, writing—original draft preparation, and writing—review and editing. I.E. and M.A.: supervision, conceptualization, investigation, writing—original draft preparation, and writing—review and editing. N.A.-A. and S.F.: conceptualization, investigation, writing—original draft preparation, and writing—review and editing. N.A.-A. and N.E.: supervision, conceptualization, investigation, and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data Availability Statement: The datasets generated during and analyzed during the current study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors are thankful to the Deanship of Scientific Research at Najran University for funding this work under the Research Groups Funding program grant code (NU/RG/SERC/12/21). We thank the ANRH agency for the collected data and the General Directorate of Scientific Research and Technological Development of Algeria (DGRSDT).

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Khan, M.T.; Shoaib, M.; Hammad, M.; Salahudin, H.; Ahmad, F.; Ahmad, S. Application of Machine Learning Techniques in Rainfall–Runoff Modelling of the Soan River Basin, Pakistan. *Water* **2021**, *13*, 3528. [CrossRef]
2. Bhusal, A.; Parajuli, U.; Regmi, S.; Kalra, A. Application of Machine Learning and Process-Based Models for Rainfall-Runoff Simulation in Dupage River Basin, Illinois. *Hydrology* **2022**, *9*, 117. [CrossRef]
3. Clark, M.P.; Kavetski, D.; Fenicia, F. Pursuing the Method of Multiple Working Hypotheses for Hydrological Modeling. *Water Resour. Res.* **2011**, *47*, W09301. [CrossRef]
4. Niu, W.; Feng, Z.; Zeng, M.; Feng, B.; Min, Y.; Cheng, C.; Zhou, J. Forecasting Reservoir Monthly Runoff via Ensemble Empirical Mode Decomposition and Extreme Learning Machine Optimized by an Improved Gravitational Search Algorithm. *Appl. Soft Comput.* **2019**, *82*, 105589. [CrossRef]
5. Li, H.; Zhang, Y.; Zhou, X. Predicting Surface Runoff from Catchment to Large Region. *Adv. Meteorol.* **2015**, *2015*, 1–13. [CrossRef]
6. Song, X.; Kong, F.; Zhan, C.; Han, J. Hybrid Optimization Rainfall-Runoff Simulation Based on Xinanjiang Model and Artificial Neural Network. *J. Hydrol. Eng.* **2012**, *17*, 1033–1041. [CrossRef]
7. Vafakhah, M.; Janizadeh, S. Application of Artificial Neural Network and Adaptive Neuro-Fuzzy Inference System in Streamflow Forecasting. In *Advances in Streamflow Forecasting*; Elsevier: Amsterdam, The Netherlands, 2021; pp. 171–191.
8. Liu, Z.; Todini, E. Towards a Comprehensive Physically-Based Rainfall-Runoff Model. *Hydrol. Earth Syst. Sci.* **2002**, *6*, 859–881. [CrossRef]
9. Xu, C.-Y.; Xiong, L.; Singh, V.P. Black-Box Hydrological Models. In *Handbook of Hydrometeorological Ensemble Forecasting*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 1–48.
10. Seo, Y.; Kim, S.; Singh, V.P. Machine Learning Models Coupled with Variational Mode Decomposition: A New Approach for Modeling Daily Rainfall-Runoff. *Atmosphere* **2018**, *9*, 251. [CrossRef]

11. Mohammadi, B. A Review on the Applications of Machine Learning for Runoff Modeling. Sustain. *Water Resour. Manag.* **2021**, *7*, 98.
12. Nourani, V.; Gökçekuş, H.; Gichamo, T. Ensemble Data-Driven Rainfall-Runoff Modeling Using Multi-Source Satellite and Gauge Rainfall Data Input Fusion. *Earth Sci. Inform.* **2021**, *14*, 1787–1808. [CrossRef]
13. Sharafati, A.; Khazaei, M.R.; Nashwan, M.S.; Al-Ansari, N.; Yaseen, Z.M.; Shahid, S. Assessing the Uncertainty Associated with Flood Features Due to Variability of Rainfall and Hydrological Parameters. *Adv. Civ. Eng.* **2020**, *2020*, 1–9. [CrossRef]
14. Mohammadi, B.; Guan, Y.; Moazenzadeh, R.; Safari, M.J.S. Implementation of Hybrid Particle Swarm Optimization-Differential Evolution Algorithms Coupled with Multi-Layer Perceptron for Suspended Sediment Load Estimation. *CATENA* **2021**, *198*, 105024. [CrossRef]
15. Tikhamarine, Y.; Souag-Gamane, D.; Ahmed, A.N.; Sammen, S.S.; Kisi, O.; Huang, Y.F.; El-Shafie, A. Rainfall-Runoff Modelling Using Improved Machine Learning Methods: Harris Hawks Optimizer vs. Particle Swarm Optimization. *J. Hydrol.* **2020**, *589*, 125133. [CrossRef]
16. Adnan, R.M.; Petroselli, A.; Heddam, S.; Santos, C.A.G.; Kisi, O. Short Term Rainfall-Runoff Modelling Using Several Machine Learning Methods and a Conceptual Event-Based Model. *Stoch. Environ. Res. Risk Assess.* **2021**, *35*, 597–616. [CrossRef]
17. Okkan, U.; Ersoy, Z.B.; Kumanlioglu, A.A.; Fistikoglu, O. Embedding Machine Learning Techniques into a Conceptual Model to Improve Monthly Runoff Simulation: A Nested Hybrid Rainfall-Runoff Modeling. *J. Hydrol.* **2021**, *598*, 126433. [CrossRef]
18. Roy, B.; Singh, M.P.; Kaloop, M.R.; Kumar, D.; Hu, J.-W.; Kumar, R.; Hwang, W.-S. Data-Driven Approach for Rainfall-Runoff Modelling Using Equilibrium Optimizer Coupled Extreme Learning Machine and Deep Neural Network. *Appl. Sci.* **2021**, *11*, 6238. [CrossRef]
19. Waqas, M.; Saifullah, M.; Hashim, S.; Khan, M.; Muhammad, S. Evaluating the Performance of Different Artificial Intelligence Techniques for Forecasting: Rainfall and Runoff Prospective. In *Weather Forecast*; IntechOpen: London, UK, 2021; p. 23.
20. Xiao, L.; Zhong, M.; Zha, D. Runoff Forecasting Using Machine-Learning Methods: Case Study in the Middle Reaches of Xijiang River. *Front. Big Data* **2022**, *4*, 752406. [CrossRef]
21. Singh, A.K.; Kumar, P.; Ali, R.; Al-Ansari, N.; Vishwakarma, D.K.; Kushwaha, K.S.; Panda, K.C.; Sagar, A.; Mirzania, E.; Elbeltagi, A. Application of Machine Learning Technique for Rainfall-Runoff Modelling of Highly Dynamic Watersheds. *arXiv* **2022**. [CrossRef]
22. Yang, M.-C.; Wang, J.-Z.; Sun, T.-Y. EMD-Based Preprocessing with a Fuzzy Inference System and a Fuzzy Neural Network to Identify Kiln Coating Collapse for Predicting Refractory Failure in the Cement Process. *Int. J. Fuzzy Syst.* **2018**, *20*, 2640–2656. [CrossRef]
23. Rouillard, V.; Sek, M.A. The Use of Intrinsic Mode Functions to Characterize Shock and Vibration in the Distribution Environment. *Packag. Technol. Sci.* **2005**, *18*, 39–51. [CrossRef]
24. Khorsandi, M.; Ashofteh, P.-S.; Azadi, F.; Chu, X. Multi-Objective Firefly Integration with the K-Nearest Neighbor to Reduce Simulation Model Calls to Accelerate the Optimal Operation of Multi-Objective Reservoirs. *Water Resour. Manag.* **2022**, *36*, 3283–3304. [CrossRef]
25. Guijo-Rubio, D.; Gutiérrez, P.A.; Casanova-Mateo, C.; Fernández, J.C.; Gómez-Orellana, A.M.; Salvador-González, P.; Salcedo-Sanz, S.; Hervás-Martínez, C. Prediction of Convective Clouds Formation Using Evolutionary Neural Computation Techniques. *Neural Comput. Appl.* **2020**, *32*, 13917–13929. [CrossRef]
26. Mohaghegh, A.; Farzin, S.; Anaraki, M.V. A New Framework for Missing Data Estimation and Reconstruction Based on the Geographical Input Information, Data Mining, and Multi-Criteria Decision-Making; Theory and Application in Missing Groundwater Data of Damghan Plain, Iran. *Groundw. Sustain. Dev.* **2022**, *17*, 100767. [CrossRef]
27. Chen, Y.; Chen, R.; Ma, C.; Tan, P. Short-Term Wind Speeds Prediction of SVM Based on Simulated Annealing Algorithm with Gauss Perturbation. *IOP Conf. Ser. Earth Environ. Sci.* **2019**, *267*, 042032. [CrossRef]
28. Breiman, L. Random Forests. *Mach. Learn* **2001**, *45*, 5–32. [CrossRef]
29. Ginidi, A.; Ghoneim, S.M.; Elsayed, A.; El-Sehiemy, R.; Shaheen, A.; El-Fergany, A. Gorilla Troops Optimizer for Electrically Based Single and Double-Diode Models of Solar Photovoltaic Systems. *Sustainability* **2021**, *13*, 9459. [CrossRef]
30. Pachpore, S.; Jadhav, P.; Ghorpade, R. Process Parameter Optimization in Manufacturing of Root Canal Device Using Gorilla Troops Optimization Algorithm. In *Computational Intelligence in Manufacturing*; Elsevier: Amsterdam, The Netherlands, 2022; pp. 175–185.
31. Daneshfaraz, R.; Aminvash, E.; Ghaderi, A.; Abraham, J.; Bagherzadeh, M. SVM Performance for Predicting the Effect of Horizontal Screen Diameters on the Hydraulic Parameters of a Vertical Drop. *Appl. Sci.* **2021**, *11*, 4238. [CrossRef]
32. Morshed-Bozorgdel, A.; Kadkhodazadeh, M.; Valikhan Anaraki, M.; Farzin, S. A Novel Framework Based on the Stacking Ensemble Machine Learning (SEML) Method: Application in Wind Speed Modeling. *Atmosphere* **2022**, *13*, 758. [CrossRef]
33. De Salis, H.H.C.; da Costa, A.M.; Vianna, J.H.M.; Schuler, M.A.; Künne, A.; Fernandes, L.F.S.; Pacheco, F.A.L. Hydrologic Modeling for Sustainable Water Resources Management in Urbanized Karst Areas. *Int. J. Environ. Res. Public Health* **2019**, *16*, 2542. [CrossRef]
34. Anaraki, M.V.; Farzin, S.; Mousavi, S.-F.; Karami, H. Uncertainty Analysis of Climate Change Impacts on Flood Frequency by Using Hybrid Machine Learning Methods. *Water Resour. Manag.* **2021**, *35*, 199–223. [CrossRef]

35. Jamei, M.; Ali, M.; Malik, A.; Prasad, R.; Abdulla, S.; Yaseen, Z.M. Forecasting Daily Flood Water Level Using Hybrid Advanced Machine Learning Based Time-Varying Filtered Empirical Mode Decomposition Approach. *Water Resour. Manag.* **2022**, *36*, 4637–4676. [CrossRef]
36. Zhou, R.; Zhang, Y. Reconstruction of Missing Spring Discharge by Using Deep Learning Models with Ensemble Empirical Mode Decomposition of Precipitation. *Environ. Sci. Pollut. Res.* **2022**, *29*, 82451–82466. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.