

Article

Precipitation Forecasting Using Multilayer Neural Network and Support Vector Machine Optimization Based on Flow Regime Algorithm Taking into Account Uncertainties of Soft Computing Models

Fatemeh Barzegari Banadkooki ^{1,*}, Mohammad Ehteram ², Ali Najah Ahmed ³ ,
Chow Ming Fai ⁴, Haitham Abdulmohsin Afan ^{5,*}, Wani M. Ridwan ⁶, Ahmed Sefelnasr ⁷ 
and Ahmed El-Shafie ⁸ 

¹ Agricultural Department, Payam Noor University, Tehran 19395-4697, Iran

² Department of Water Engineering and Hydraulic Structures, Faculty of Civil Engineering, Semnan University, Semnan 3513119111, Iran; mohammdehteram@semnan.ac.ir

³ Institute of Energy Infrastructure (IEI), Universiti Tenaga Nasional, Selangor 43000, Malaysia; Mahfoodh@uniten.edu.my

⁴ Institute of Sustainable Energy (ISE), Universiti Tenaga Nasional (UNITEN), Selangor 43000, Malaysia; Chowmf@uniten.edu.my

⁵ Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam

⁶ Department of Civil Engineering, Universiti Tenaga Nasional, (UNITEN), Selangor 43000, Malaysia; ridwaniyah.ridwan@uniten.edu.my

⁷ National Water Center, United Arab Emirate University, P.O. Box Al Ain 15551, UAE; ahmed.sefelnasr@uaeu.ac.ae

⁸ Department of Civil Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia; elshafie@um.edu.my

* Correspondence: barzegari@pnu.ac.ir (F.B.B.); haithamabdulmohsinafan@duytan.edu.vn (H.A.A.)

Received: 14 October 2019; Accepted: 21 November 2019; Published: 26 November 2019



Abstract: Drought, climate change, and demand make precipitation forecast a very important issue in water resource management. The present study aims to develop a forecasting model for monthly precipitation in the basin of the province of East Azarbaijan in Iran over a ten-year period using the multilayer perceptron neural network (MLP) and support vector regression (SVR) models. In this study, the flow regime optimization algorithm (FRA) was applied to optimize the multilayer neural network and support vector machine. The flow regime optimization algorithm not only identifies the parameters of the SVR and MLP models but also replaces the training algorithms. The decision tree model (M5T) was also used to forecast precipitation and compare it with the results of hybrid models. Principal component analysis (PCA) was used to identify effective indicators for precipitation forecast. In the first scenario, the input data include temperature data with a delay of one to twelve months, the second scenario includes precipitation data with a delay of one to twelve months, and the third scenario includes precipitation and temperature data with a delay of one to three months. The mean absolute error (MAE) and Nash–Sutcliffe error (NSE) indices were used to evaluate the performance of the models. The results showed that the proposed MLP–FRA outperformed all the other examined models. Regarding the uncertainties of the models, it was also shown that the MLP–FRA model had a lower uncertainty band width than other models, and a higher percentage of the data will fall within the range of the confidence band. As the selected scenario, Scenario 3 had a better performance. Finally, monthly precipitation maps were generated based on the MLP–FRA model and Scenario 3 using the weighted interpolation method, which showed significant precipitation in spring and winter and a low level of precipitation in summer. The results of the present study showed that MLP–FRA has high capability to predict hydrological variables and can be used in future research.

Keywords: precipitation forecasting; soft computing models; flow regime algorithm; multilayer perceptron neural network

1. Introduction

Consecutive droughts and increasing water demands mean that prediction and planning for the usage of precipitation are necessary for decision makers [1,2]. In addition, precipitation forecasting is necessary to prevent floods and construct flood-controlling structures. What is more, precipitation forecasting is one of the main issues of the water resources planning that can reduce the effects of drought [3]. Researchers have used various statistical and hydrological models to predict precipitation; however, it should be noted that precipitation patterns are dependent on various parameters and have a nonlinear behavior. Therefore, these complexities result in uncertainties in precipitation forecasting models. In recent years, researchers have used soft computing models to forecast hydrological variables such as precipitation [4]. Models such as artificial neural network (ANN), adaptive neuro-fuzzy inference system (ANFIS), support vector machine (SVM), or improved regression models and optimization algorithms not only are simple in structure and application but also have a wide range of capabilities due to high precision and short computation time [5]. The high flexibilities and prediction based on different kinds of data are another characteristic of soft computing models. Kisi and Cimen [6] used a wavelet neural network model to forecast precipitation. Daily precipitation was forecasted based on the hybrid model of support vector machine–wavelet transform approach. The results showed that the RMSE (root-mean-square error) index obtained by the hybrid model was reduced by 50% compared to that obtained by the simple support vector machine. Ramana et al. [7] used a wavelet neural network to forecast monthly precipitation. Input data included maximum temperature, precipitation, and humidity of previous months. The results showed that the wavelet neural network model had a larger Nash–Sutcliffe (NSE) coefficient compared to the neural network and regression models. Shamshirb et al. [8] used a support vector machine and a neural fuzzy model to forecast monthly precipitation. The results showed that the neural fuzzy model significantly reduced the mean absolute error (MAE) compared to the support vector machine model. Kisi and Sanikhani [9] used an adaptive neuro-fuzzy inference system (ANFIS) with grid partition (GP), ANFIS with subtractive clustering (SC), a support vector machine, and the neural network to predict monthly and annual precipitation in Iran. The results showed that the highest precipitation occurs in the north, west, and southwest of Iran and the lowest in the east and southeast, which was precisely modeled by an adaptive neuro-fuzzy inference system (ANFIS) with grid partition (GP). Shenify et al. [10] forecasted the precipitation based on a wavelet neural network–support vector machine, genetic programming, and a neural network. Their results showed that the wavelet neural network–support vector machine model is more accurate in precipitation forecast than the neural network and genetic programming. Amiri et al. [11] used ANN and wavelet neural network models for precipitation forecasting in the Aidoghush basin in Iran. The results showed an increase in the R^2 coefficient and a decrease in the relative error (RE) of the wavelet neural network model compared to the ANN model in precipitation forecasting. Du et al. [12] used a support vector machine with the particle swarm algorithm as a hybrid model to forecast the precipitation. The particle swarm algorithm increased the accuracy of the model by finding the optimal parameters of the support vector machine, so that the hybrid model of the support vector and particle swarm reduced the relative error rate by 20% compared to the support vector model.

Mirabbasi et al. [13] made a precipitation forecast based on tree modeling, genetic programming, and the least squares support vector machine. The results showed that the RMSE values of the least squares vector machine model, genetic programming, and decision tree model were 13.96, 36.74, and 37.22 mm, respectively. Therefore, the performance of the least squares support vector machine model was confirmed. Mehr et al. [14] used hybrid models of the support vector machine

and firefly algorithm (FFA) to forecast precipitation in Iran. The results showed that the improved support vector machine increased the accuracy of the results and reduced the relative error rate by up to 30% compared to the simple model of support vector machine. Azad et al. [15] used a fuzzy model along with the particle swarm algorithm, ant colony optimization algorithm, and the genetic algorithms to forecast the precipitation. The results showed that the neuro fuzzy–ant colony optimization algorithm not only decreases the relative error and mean absolute error (MAE) but also has a higher convergence rate than the other models. Kumar et al. [16] used recurrent neural network to forecast precipitation in India. Monthly precipitation data were considered from the years 1871 to 2016 to forecast monthly precipitation. The results showed high accuracy of a recurrent neural network based on a high NSE coefficient. Another study used a hybrid wavelet–M5 model tree to forecast precipitation [17]. Hossain et al. [18] used the neural network and regression models to forecast monthly precipitation. The result showed that the neural network model and the outputs had a high correlation with the precipitation model of Western Australia and a higher NSE coefficient than the multiple regression model. In any case, previous studies have shown that methods such as the neural network, fuzzy–neural network and support vector machine have good capabilities to forecast precipitation [19–24]. Models such as the neural network and support vector machine have a good performance, however, it is important to note that they have unknown parameters, the values of which should be obtained before precipitation forecast [25–27]. The number of hidden layers and neurons, values of weight and bias in neural networks, as well as the parameters associated with the kernel function should be quantified in the support vector machine, so that the accuracy of the forecasting models would be acceptable. Although there are training algorithms in the structure of neural network and other algorithms to accurately calculate the parameters of the support vector machine, optimization and evolutionary algorithms are a priority due to their ease of use, high convergence rate, and high accuracy in finding the optimal solution.

It should be noted here that the novelty of the current research study is to introduce a new prediction model for precipitation utilizing advanced machine learning model. In fact, the proposed model is a further enhancement for the traditional machine learning model after improving its backpropagation algorithm with a nature-inspired optimization algorithm. In addition, a new way of model performance analysis has been introduced, which is the uncertainty. Finally, the model outputs have been presented as precipitation forecasting zone mapping for case study area which is considered as new visual presentation for forecasting model.

The present study trains the new models of the multilayer neural network and support vector machine based on the new flow regime optimization algorithm and then compares the results with the MT5 tree model. These models are used to predict precipitation, and a comprehensive comparison is made between them considering the uncertainties of the models. The optimization algorithm is used to determine weights, bias, number of hidden layers and neurons, as well as parameters associated with the support vector model. The monthly temperature and precipitation are used as the inputs to the soft computing models. Section 2 explains the material and methods. Section 3 explains the used case study and source of data in the current article. The results and discussions are explained by the Section 4. Finally, Section 5 presents the general results and next steps of the current research.

2. Materials and Methods

2.1. Support Vector Machine

The support vector machine is one of the most widely used hydrological simulation models providing simple structure and acceptable results. This model is a regression analysis to predict or simulate time series. The linear form of the support vector machine is based on Equation (1) [23–27]:

$$f(x) = w^{Tr} \cdot x + b \quad (1)$$

where x is the input variable, b is the bias, w is the weight, and Tr is the transpose. The difference between the observed and simulated data is minimized by the calculation process of the support vector machine. Therefore, the support vector machine tries to minimize the error function of an optimization problem. If the prediction error values are within the permissible range (ε), the error is ignored. The mathematical form of the optimization problem is consistent with Equation (2) [28–30]:

$$\begin{aligned} & \text{Minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m (\xi_i^- + \xi_i^+) \\ & \text{subject to } (w_i x_i + b) - y_i < \varepsilon + \xi_i^+ \end{aligned} \quad (2)$$

where C is the penalty factor, ξ_i^- and ξ_i^+ are penalties of training data of which the error is outside of the ε range, w_i is the weight, x_i is the input variable, and y_i is the output variable. The values of w and b are calculated by Equation (2) and then substituted in Equation (1). There are multiple kernel functions for the support vector machine, and according to the previous studies, the radial kernel function (Equation (3)) is an effective and widely used function.

$$f(x) = w^{Tr} \cdot K(x, x_i) + b \quad (3)$$

$$K(x, x_i) = \exp\left(-\frac{|x - x_i|^2}{2\gamma^2}\right) \quad (4)$$

$K(x, x_i)$ is the kernel function, and γ is the radial kernel parameter. In addition to the radial kernel parameter, C and ε are parameters which have to be calculated precisely to achieve a suitable prediction model. In the present study, an optimization method was used to achieve these parameters and increase the accuracy of the support vector machine model. The flow regime optimization algorithm was used to obtain the best value of SVR parameters. The algorithm was created based on fluid flow concepts [30], which will be introduced later in the following section Section 2.4.

2.2. Multilayer Perceptron (MLP)

Previous studies have shown that the multilayer neural network is one of the most effective and accurate tools for predicting hydrological variables. It contains input, hidden, and output layers. The network works based on Equation (5) [29]:

$$y(k) = \sum_{j=1}^m w_j f_j \left(\sum_{i=1}^n w_{ij} x_i(k) + w_0 \right), \quad (5)$$

where $y(k)$ is the objective variable, w_j is the weight coefficient of neurons, f_j is the nonlinear activation function of the j th neuron, n is the number of neurons of input layers, m in the number of neurons of hidden layer, $x_i(k)$ is the input variable at time k , w_0 is the bias of the output layer, and T is the transpose of the matrix.

The performance of the multilayer neural network depends on a precise determination of the number of hidden layers, hidden neurons, and weight and bias values. Although multilayer neural networks have training algorithms such as the Levenberg–Marquardt in their structure, the training time and accuracy of these training algorithms does not reach an acceptable level in some problems. Starting the optimization from a point away from the ultimate minimum, the Levenberg–Marquardt algorithm tries to minimize the error measure to reach an ultimate minimum. As with other training algorithms, the mentioned algorithm has an iterative process, and the initial values of the neural network parameters are obtained by a primitive guess. Studies have shown that if neural network models use an accurate and fast algorithm to find their unknown values, they will have high accuracy. The present study used a fluid flow regime in addition to the Levenberg–Marquardt algorithm to train

the neural network. In fact, the aim of the paper is related to the development of an artificial neural network based on the flow regime optimization algorithm of a traditional training algorithm.

2.3. Decision Tree Model

The decision tree model is a simple model without high complexity and has a good accuracy that is widely used in the field of hydrological prediction [29]. The decision tree model creates a multivariate linear model for the data in each inner node. A dual stage method is considered for the decision tree model. The decision trees are generated based on splitting of input or output data into subsets in the first level.

Applying 'divide-and-conquer', a model is generated where N data are accompanied with a leaf or a test criterion that divides them into subsets corresponding to a test output. The splitting criterion depends on treating the standard deviation of class values and computing the decrement in error.

The values of the standard deviation reduction are calculated based on Equation (6):

$$SDR = sd(T) - \sum \frac{|T_i|}{|T|} sd(T_i), \quad (6)$$

where T is a series of samples that reach the node, T_i is the i th output of the model, i is the number of available data records and sd is the standard deviation. After the number of split branches is maximized, the tree model selects the split that maximizes the expected reduction. The high number of divisions and branches of the tree model causes overfitting in the tree model. Therefore, the pruning stage is performed for the tree model. Pruning causes the input model of the tree model to be divided into smaller areas. In addition, based on the greedy algorithm, the model eliminates the variables that have little to do with modeling.

2.4. Flow Regime Optimization Algorithm

The flow regime optimization algorithm is used as one of the new optimization algorithms in mathematical modeling, engineering problems, structural problems, and other optimization areas. It is known as a successful algorithm due to its fast convergence, high accuracy of the results, use of advanced operators to diversify solutions, and creating a great balance between diversification and intensification [30]. The algorithm was created based on fluid flow concepts. According to the laws of fluid mechanics and hydraulics, the fluid flow is divided into laminar and turbulent states based on the ratio of inertia to viscosity force. The laminar state of the flow regime is known as a local search and the turbulent state as a global search. The Reynolds number is also responsible for determining the type of flow. In the present study, a number similar to Reynolds determined the type of flow. In addition, another hypothesis of the algorithm is that the optimal response is the starting point of the movement of fluid in the boundary layer. As the fluid flows over a surface, due to the viscosity of the flow, a velocity profile is formed. The flow velocity is zero on the surface, and at the elevation known as the boundary layer thickness, it is equal to 99% of the velocity on the surface. Like other algorithms, the mentioned algorithm has an initial population. Here, the fluid particles are known as the initial population (Equation (7)):

$$fluid(particle) = [x_1, x_2, x_3, \dots, x_{Nvar}], \quad (7)$$

where x_i is known as the i th particle of the fluid and $Nvar$ is the number of decision variables. The objective function is then calculated for each fluid particle. After calculating the objective function, the best particle is selected as the global optimum. The best solutions replace the worst ones in each iteration of the algorithm. With the increase in the number of iterations, the algorithm changes the search from the global state to the local one. To distinguish the local search from the global one, a number similar to Reynolds number is used (Equation (8)):

$$STF^i = 3.2 \times 10^6 \times \frac{\maxit}{n} \times \frac{\|g_n - particle^j\|}{\|particle^k - particle^j\|} \quad (8)$$

where STF^i is the search type factor, n in the current iteration number, \maxit is the maximum iteration, i, j , and k are the number of solutions, and g_n is the global optimum solution. Then, Equation (9) is used to determine the type of search (local or global search):

$$\left[\begin{array}{l} STF \geq 3.2 \times 10^5 \rightarrow global(search) \\ STF < 3.2 \times 10^5 \rightarrow local(search) \end{array} \right]. \quad (9)$$

3.2×10^5 was used because the search type factor (STF) is similar to the Reynolds number used to determine the type of flow in the boundary layer. As mentioned, evolutionary algorithms have both local and global search types. The algorithms first make a global search for the global optimum and try to find the optimum solutions around the global optimum. The term $\frac{\maxit}{n}$ in Equation (8) is related to the iteration and search space, in such a way that in the initial iteration, the search is of the global type, and with the increase in the number of iterations, the search space becomes smaller and changes to the local type. The second term of Equation (8) is the ratio of two Euclidean distance terms. The first is the distance between the i th solution and the best solution, and the second is the distance between two random solutions. This term can be larger or smaller than one. If it is larger than 1, it means that the distance between two random solutions is smaller than the distance between the optimal and the random solution, thus reducing the global ability search. When the mentioned term (the second term in Equation (8)) is larger than 1, the STF value is increased to improve the global ability search. If the second term of Equation (8) is less than 1, it means that the algorithm searches the adjacent space of the best solution, and thus, the local search ability is increased by the algorithm. Figure 1 shows two possibilities for the solutions. STF is the search type factor. If $STF \geq 3.2 \times 10^5$, the i th particle is far from boundary layer starting point (global best solution). If $STF < 3.2 \times 10^5$, the particle will approach the boundary layer, which is the optimum solution. δ_l is the thickness of the stable boundary layer, and δ_t is the thickness of the turbulent boundary layer. The yellow curve shows the threshold of the thickness of stable boundary and the thickness of turbulent. The dashed lines depict the radius of turbulent and stable area.

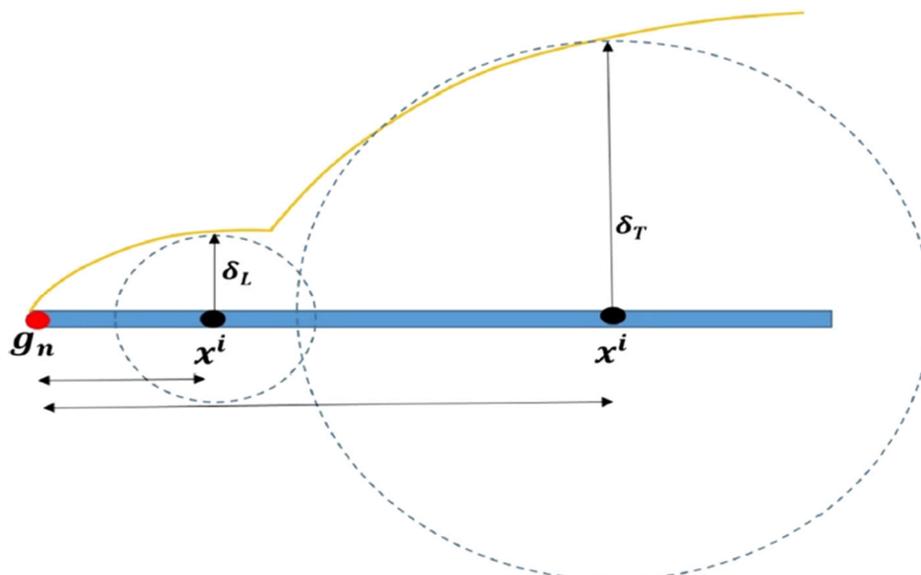


Figure 1. Two possibilities for each solution.

Finally, Equations (10) and (11) are used for global and local search in the following order:

$$x_i^{n+1} = x_i^n + \gamma(\text{Leavy}) \times (g_n - x_i^n) \times \frac{0.37}{\sqrt[3]{STF}}, \quad (10)$$

$$x_i^{n+1} = x_i^n + \gamma(\text{Rand}) \times (g_n - x_i^n) \times \frac{4.96}{\sqrt[3]{STF}}. \quad (11)$$

Leavy is the number created by *Leavy* distribution, *Rand* is a random number, and γ is the scaling factor equal to 0.30.

2.5. Construction of Hybrid Models (ANN–Flow Regime Optimization Algorithm (FRA) and SVM–FRA)

As mentioned above, the multilayer neural network model requires determining the number of hidden layers, the number of hidden neurons, the weight values, and bias values. In addition, the support vector machine requires determining the parameters of γ , C , and ε . These parameters were explained in the earlier sections and they were regarded as the kernel and SVR parameters.

In the present paper, the hybrid model of the neural network–flow regime and the support vector machine–flow regime is used as follows to find the above values:

1. The input data are identified.
2. The data are normalized.
3. The training phase is completed.
4. The criterion of stopping the modeling process is controlled. If satisfactory, go to step 10; otherwise, go to step 5.
5. The parameters of the multilayer neural network, including weight, bias, number of neurons and hidden layers, as well as the support vector model, are defined as the initial population of fluid particles. The above parameters are considered as decision variables in the flow regime algorithm.
6. The objective function is computed for the fluid particle population members. The present study considers the RMSE value as the objective function. Then, the best particle or optimal global solution is determined with the best objective function.
7. The two random particles of J and K are determined, and the STF value is calculated.
8. Particle movement is based on control of the STF value, and Equations (9)–(11) are used for the particle movement.
9. The maximum number of iterations is controlled. If satisfied, the algorithm is stopped and goes to step 3; otherwise, it goes to step 5.
10. The test phase is performed, and the output is provided, which is the monthly precipitation value.

3. Case Study

The present case study made a precipitation forecast in one of Iran's major basins called Aidogh mush. The River Aidogh mush is one of the main rivers of the Ghezel Ozen basins. The area of the basin is 1800 km². This river is located in the East Azarbaijan province in Iran. The study area is in the geographical coordinates of 46°52' to 47°45' E longitude and 36°43' to 37°26' N latitude (Figure 2). The annual discharge of the basin is 170 million m³. The average precipitation in the entire basin is 336.2 mm, and the length of the Aidogh mush River is 80 km. The elevation of the basin ranges from 1100 to 2500 m. In the present study, precipitation data from 2000 to 2010 were analyzed to predict precipitation. In total, 70% of data were used for the training phase and 30% for the test phase. This splitting percentage was selected upon the strong recommendations from previous studies. Therefore, the current setup for this model, 70% for training and 30% for testing, is suitable for this study [18–21]. The data were obtained from rain gauge stations that operated by the meteorological department in Iran. Furthermore, the model has been structured to provide one-month-ahead forecasting for precipitation.

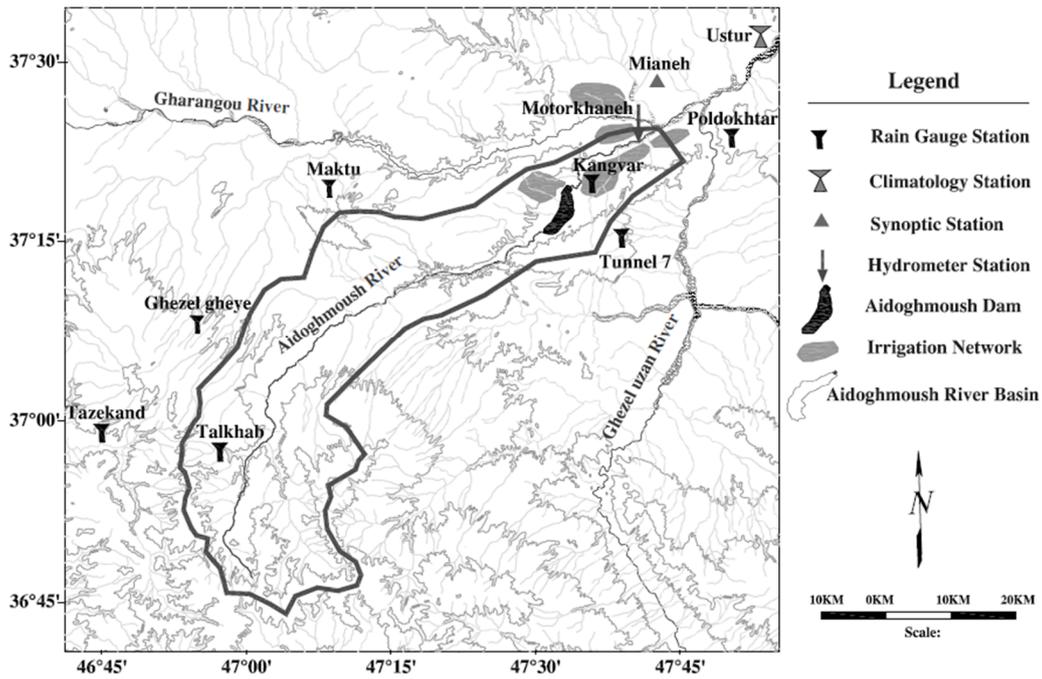


Figure 2. Map illustrates the study domain and locations of multiple observing stations.

The average annual precipitation of the basin is 236.2 mm, and the maximum of monthly average precipitation is in April. The basin has an average annual temperature of 11.6 °C. July has the absolute maximum temperature of 21.9 °C. The minimum temperature is also −16.8 °C in January. The average normal monthly temperature was below zero in January for the basin, with a maximum of −1.79 °C. In summer, temperatures above 25 °C were observed. Precipitation exceeds 30 in winter and spring (monthly value). Winter and spring have the highest precipitations, and the precipitation in summer is nearly zero. Figure 3 demonstrates the monthly average of temperature and precipitation in the research region.

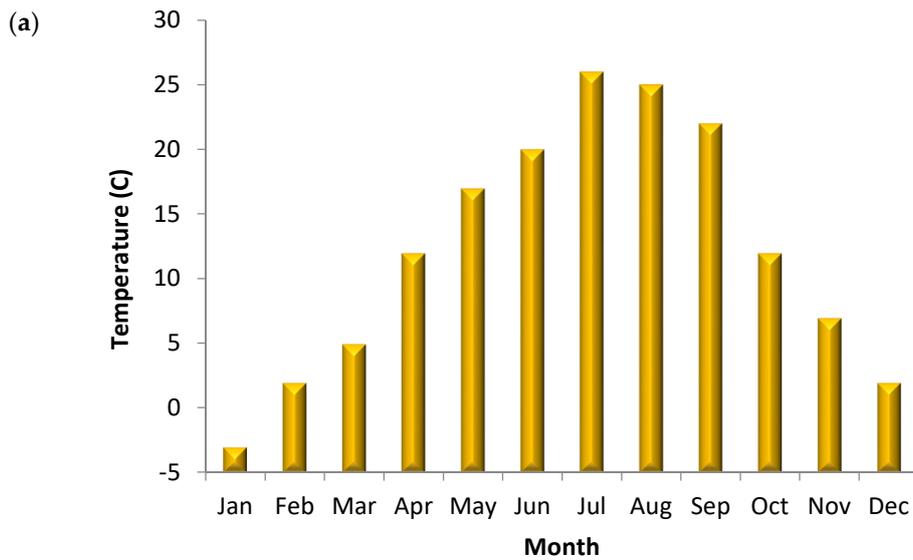


Figure 3. Cont.

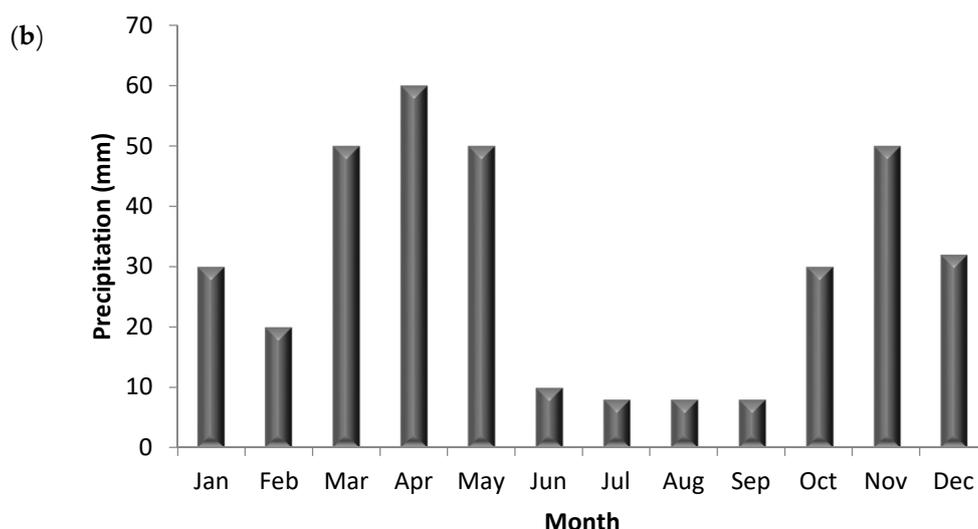


Figure 3. (a): Monthly average of temperature and (b): monthly average of precipitation (2000–2010).

Three scenarios were taken into account to forecast the precipitation. The lead times were suggested after initial screening of correlation between lag times and precipitation values to be one-month-ahead forecasting. The precipitation and the temperature values are selected to be the input variables for the model because the high correlation between these two variables has been proved, and, hence, it is necessary while developing the forecasting model for precipitation to consider the temperature as one of the input variables [18–21].

1. Input of precipitation forecasting models is the average temperature based on various time delays from 1 to 12 months.
2. Input of precipitation forecasting models is the average precipitation based on various time delays from 1 to 12 months.
3. Input of precipitation forecasting models is the average temperature and precipitation based on various time delays from 1 to 12 months.

When the number of input variables is high as in these three scenarios, the principal component analysis method can be used. In this method, the initial variables of the problem are transformed into new and independent components. The new components are a linear combination of the initial variables. Given that all variables are used in the formation of components, the components are able to provide preliminary information on all variables without losing details. First, the Kaiser–Meyer–Olkin (KMO) coefficient was used to determine the applicability of the principal component analysis method (Equation (12)). If the coefficient is greater than 0.5, the implementation of the proposed method is allowed. Each component has a percentage of the information provided by the initial variables. The component that is most important in providing the data has the highest variance, and the component that has the least variance is considered as the last component. If all the initial variables are used in component generation, it will be difficult to analyze the components. Therefore, component rotation is usually selected to simplify component analysis. The orthogonal component rotation maintains the independence between the components. Therefore, the most influential variables are expressed in every component after each rotation.

$$KMO = \frac{\sum_{i=1}^p \sum_{j=1}^p r_{ij}^2}{\sum_{i=1}^p \sum_{j=1}^p r_{ij}^2 + \sum_{i=1}^p \sum_{j=1}^p \alpha_{ij}^2}, \quad (12)$$

where r_{ij} is the correlation coefficient and α_{ij} is the partial correlation coefficient between i and j variables.

The error indices presented in Equations (13) to (15) are used to evaluate different models' efficiency.

$$NSE = 1 - \frac{\left[\sum_{i=1}^n (Y_i^{obs} - Y_i^{sim})^2 \right]}{\left[\sum_{i=1}^n (Y_i^{obs} - Y^{mean})^2 \right]}, \quad (13)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i^{obs} - Y_i^{sim}|, \quad (14)$$

$$RSR = \frac{RMSE}{STDEV} = \frac{\left[\sqrt{\sum_{i=1}^n (Y_i^{obs} - Y_i^{sim})^2} \right]}{\left[\sqrt{\sum_{i=1}^n (Y_i^{obs} - Y^{mean})^2} \right]}, \quad (15)$$

where Y_i^{obs} is observed data, Y_i^{sim} is simulated data, Y^{mean} is the mean data, MAE is the mean absolute error of data, RSR is the root mean square error-observations standard deviation ratio, and NSE is the Nash–Sutcliffe efficiency coefficient. The above indexes are widely used to evaluate the performances of the soft computing models [15–21,30].

4. Results and Discussion

All the proposed models have been developed based on the suggested structure that has been explained in the previous sections using the collected data. Herein, detailed explanations about the procedure for initiating the models and examine their performance have been introduced. At first, in order to confirm the applicability of the principal component analysis method, the KMO coefficient should be examined first to check the suitability of the used data. Applying Equation (12) on the used data showed that the KMO value is about 0.89 which confirm the suitability of the used data. In the following sub-sections, further analysis on the sensitivity, model input selection, and comparison between different models for each scenarios will be presented.

4.1. Sensitivity Analysis of FRA Parameters

Evolutionary algorithms have random parameters, and the exact values of random parameters must be determined to begin the optimization processed by the algorithm. FRA is one of the algorithms that needs the determination of the maximum number of iterations and the initial population of particles but has fewer random parameters than other evolutionary algorithms. First, an objective function was considered for the optimization process. The objective function of the present study was RMSE. The population size was selected, and then an iteration number was determined. To combine these two parameters, the objective function was computed. Then, for a population size of 100 and other values of the iteration number, a separate compound was considered, and the value of the objective function was calculated based on Table 1. Similarly, such a process was then followed for a population of 200 iterations, and different combinations were generated for a population of 200 based on the different iteration numbers. Finally, the combination with the least error (objective function) was selected. After this process, a population size of 300 and an iteration number of 200 were selected. In fact, Table 1 indicates the variation of parameters versus the objective function to show the way of the selection of the best parameters.

Table 1. Sensitivity analysis of flow regime optimization algorithm (FRA) parameters.

Population	Number of Iteration
100	200
200	400
300	600
400	800

Population	Number of Iteration
100	200
200	400
300	600
400	800

Population	Number of Iteration
100	200
200	400
300	600
400	800

Population	Number of Iteration
100	200
200	400
300	600
400	800

4.2. Selection of Input Model Parameters

Figure 4 shows the eigenvalues of the components. For example, if we consider Scenario 1 (the average temperature data with delays of 1 to 12 months), 12 different components are available. As observed, the first three components in Scenario 1 have the highest eigenvalue, accounting for 85% of the total eigenvalue, which is equivalent to 8. Table 1 shows the parameters used in different scenarios. In addition, the first three components comprise 87% of the total eigenvalue. Similarly, this is the case for the third scenario. Another important point is to select the influential parameters; for example, in Scenario 1, there are 12 parameters consistent with Table 2. Figure 4 shows the loading values for the various parameters in the components. Results show that the first six input data have loading coefficients between 0.9 and 0.8, which are more important than other input data. Thus, there are three components with six input data for Scenario 1. The same procedure was repeated for Scenarios 2 and 3, and the values of the loading coefficients were calculated for these scenarios, too. To avoid duplication of results, the values of the loading coefficients for the inputs of Scenario 1 are shown. The results showed that Scenarios 2 and 3 have three influential components according to Figure 4, in which five initial parameters among them have an impact factor between 0.8 and 0.9. Therefore, Table 2 shows the input parameters in all three components of the three scenarios. According to Table 2, Scenario 1, with three components and temperature parameters with a delay of one to six months, Scenario 2, with precipitation parameters with a delay of one to six months, and Scenario 3, with precipitation parameters with a delay of one, two, and three months and temperature with a delay of two and three months are the five main parameters of the three components of Scenario 3. The significant parameters were identified by the load coefficient. Each principal component has consisted of multiplying significant parameters (lagged temperature and rainfall inputs) by load coefficient. For example, the required coefficients for the third scenario were extracted by Figure 4d. Regarding the similar process, other required coefficients were extracted. Figure 4a–c makes it possible to know the number of effective PCs on the results. Each PC is an independent variable. Finally, the PCs were used by the soft computing models to find a relationship between them and the dependent variable (monthly rainfall).

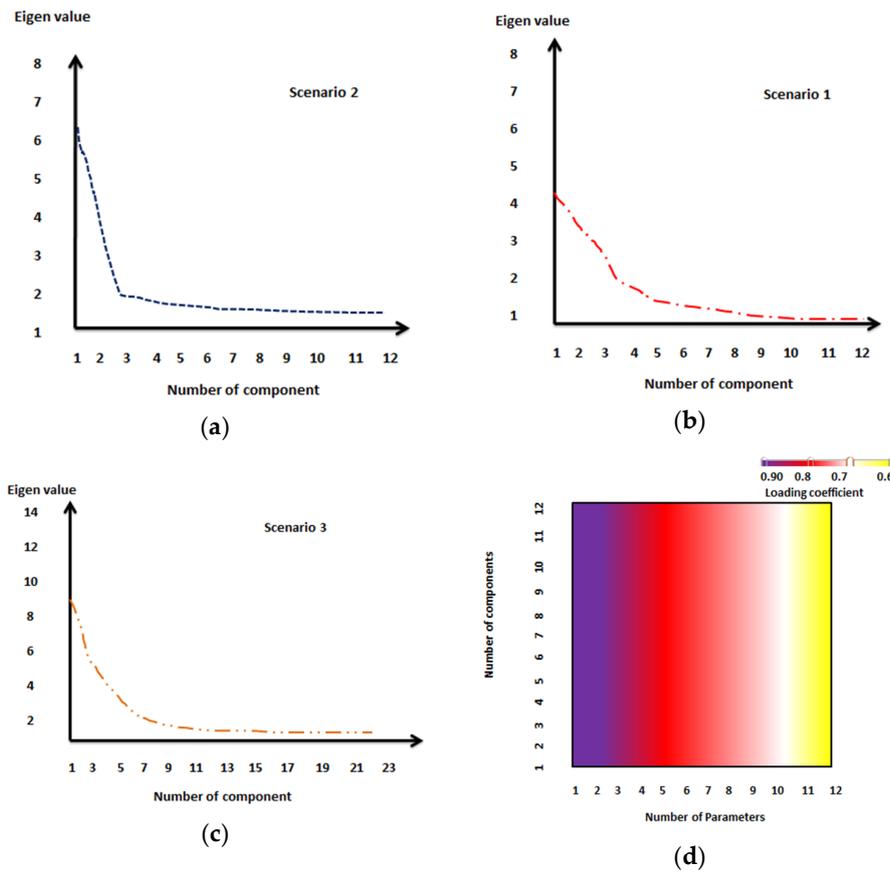


Figure 4. (a) Variation of Eigen value for Scenario 2, (b) variation of Eigen value for Scenario 1, (c) variation of Eigen value for scenario 3, (d) the load coefficient for Scenario 3.

Table 2. (a) Effective input parameters of different scenarios, (b) range of input variables.

(a)		
Selected Parameters for Three Components	Input Parameters	Scenario
T (t-1), T (t-2), T (t-3), T (t-4), T (t-5), T (t-6)	T (t-1) (t-12)	1
R (t-1), R (t-2), R (t-3), R (t-4), R (t-5), R (t-6)	R (t-1) (t-12)	2
R (t-1), R (t-2), T (t-1), R (t-3), T (t-2)	T (t-1) (t-12) And R (t-1) (t-12)	3
(b)		
Parameter	Rainfall (mm)	Temperature (°C)
Training		
Minimum	10	−3
Maximum	70	27
Average	35.2	19
Testing		
Minimum	12	−2
Maximum	65	26
Average	34.90	21

T: Temperature, R: precipitation and t-1: a lag of one month, t-2: a lag of two months, And t-12: a lag of 12 months.

4.3. Comparison of Results of Different Scenarios

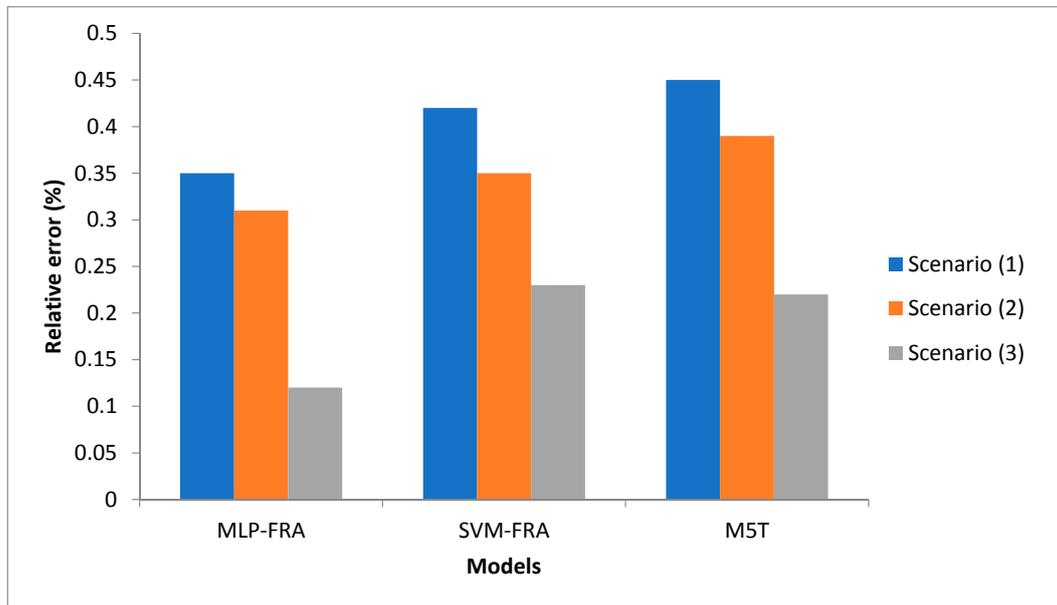
Table 3 shows the results of applying different models to the selected scenarios. It is clear that the MLP–FRA model performs better than other models in all scenarios. For example, if Scenario 1 and the MAE index are considered in the test phase, the MLP–FRA model reduces the MAE error rate by 18% and 8%, respectively, compared to the support vector regression (SVR)–FRA and decision tree (M5T) models. Also considering Scenario 2 and the NSE index, the MLP–FRA model has an NSE equal to 0.80, while this index equals 0.79 and 0.75 for the SVR–FRA and M5T models. In addition, as per Scenario 3, the value of the RSR index for the MLP–FRA model is 0.14 in the test phase, while the index equals 0.25 and 0.42 for the two models of SVR–FRA and M5T, respectively. Further, according to the results of Table 3, Scenarios 3, 2, and 1 are prioritized in that order.

Table 3. Analysis of scenario results and different models.

Train			
Scenario (1)			
Scenario	MLP–FRA	SVM–FRA	M5T
MAE	1.230	1.432	1.512
RSR	0.24	0.35	0.37
NSE	0.78	0.72	0.70
Scenario (2)			
MAE	1.002	1.212	1.456
RSR	0.21	0.29	0.34
NSE	0.83	0.80	0.72
Scenario (3)			
MAE	0.912	1.004	0.988
RSR	0.12	0.20	0.27
NSE	0.93	0.88	0.85
Test			
Scenario (1)			
MAE	1.334	1.445	1.612
RSR	0.29	0.37	0.39
NSE	0.76	0.70	0.68
Scenario (2)			
MAE	1.112	1.219	1.545
RSR	0.25	0.31	0.38
NSE	0.80	0.79	0.75
Scenario (3)			
MAE	0.941	1.114	1.547
RSR	0.14	0.25	0.42
NSE	0.92	0.86	0.74

Figure 5 illustrates the performance of different models and scenarios. Taylor’s diagram was plotted based on the values of standard deviation, correlation coefficient, as well as RMSE—noted on the circles from 0.4 to 1.6. The best-performing model is closest to the reference point. Therefore, in all scenarios, the SLP–FLA model is closer to the reference point than the other two models. On the other hand, in Scenario 3, all models perform better than other scenarios [18–21]. Figure 5a illustrated the maximum relative error that have been achieved from different models. It could be observed that the MLP–FRA could provide lowest relative error (The ratio between the Absolute Error and the Actual Value of the item) compared to the other models for all scenarios. However, performance of models

can be different during the testing level and training level. It is related to the nature of data but the MLP-FRA outperformed other models during both levels.



(a)

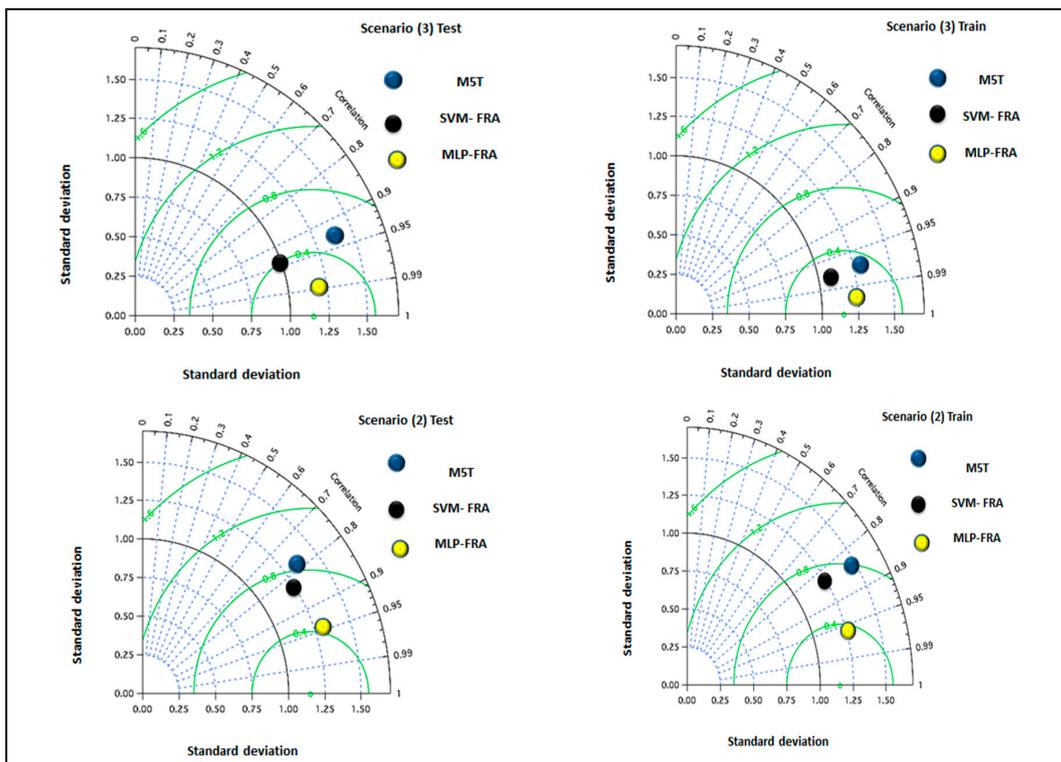


Figure 5. Cont.

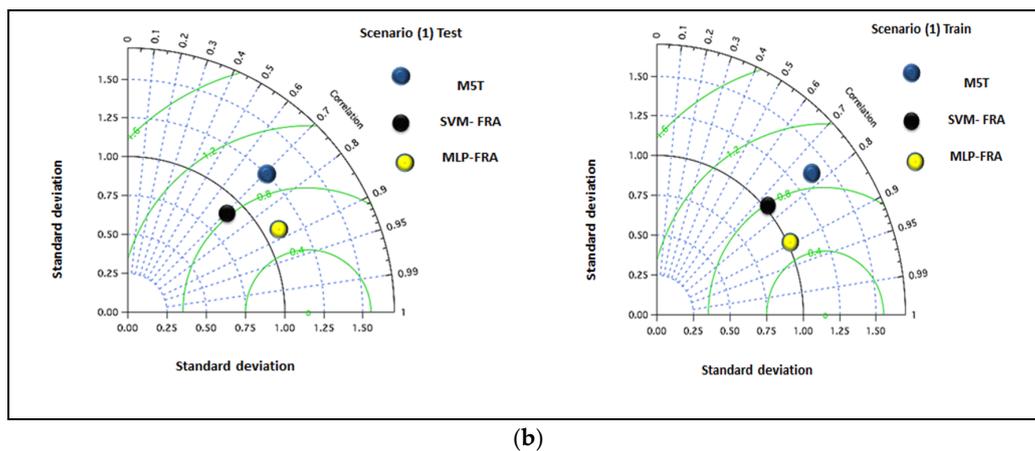


Figure 5. (a) Computed relative error for soft computing models and (b) performance of different model displayed in the form of Taylor's diagram.

The input data to various soft computing models include uncertainties, which in turn make the computational models uncertain in their outputs and performance. Therefore, the uncertainties should be taken into account in all models. The Monte Carlo simulation was used to generate random numbers [18–21]. Random numbers were used to generate each variable, and then the probability function that better fits the distribution of the intended variable was simulated and generated. At this stage, when the distribution function of each input variable was determined, the necessary samples were generated using Simlab 2.2. This software is able to use different distribution functions based on seven sampling methods. Then the distribution equations of each parameter were used to generate samples. The results showed that 2000 samples can provide the best conditions for convergence. Then, high limits (97.5%) and low limits (2.5%) of the generated time series were specified, and 2000 time series were generated for 10 years. To measure the uncertainty of the models, the model uncertainty bandwidth index (denoted by d) and percentage of the data replacement within the range of confidence band (denoted by p) were used. The results of the test phase are presented in Figure 6. The results showed that MLP-FRA model and Scenario 3 have a better performance than other models with $d = 0.10$ and $p = 95\%$ in the test phase. Further, following the neural network model, the SVM-FRA model also performs best in Scenario 3 with $d = 0.14$ and $p = 94\%$ (Figure 6). Two equations were used to calculate the indices:

$$p = \frac{1}{n} \text{count}(R|X_l \leq R \leq X_u) \quad (16)$$

$$d = \frac{\bar{d}_x}{\sigma_x}$$

where p is the range of confidence band; X_u is the upper bound of the domain; X_l is the lower bound of the domain; σ_x is the standard deviation of the forecasted variable R (precipitation); d is the model uncertainty bandwidth index; and \bar{d}_x is the average distance between the upper and lower bound.

Figure 7 shows the R^2 coefficient for all models under Scenario 3. The previous results showed that the best performance of the models is related to Scenario 3. The results of this phase show that the R^2 value is higher for the MLP-FRA model than the other models in the test and training levels, which indicates the superiority of the model.

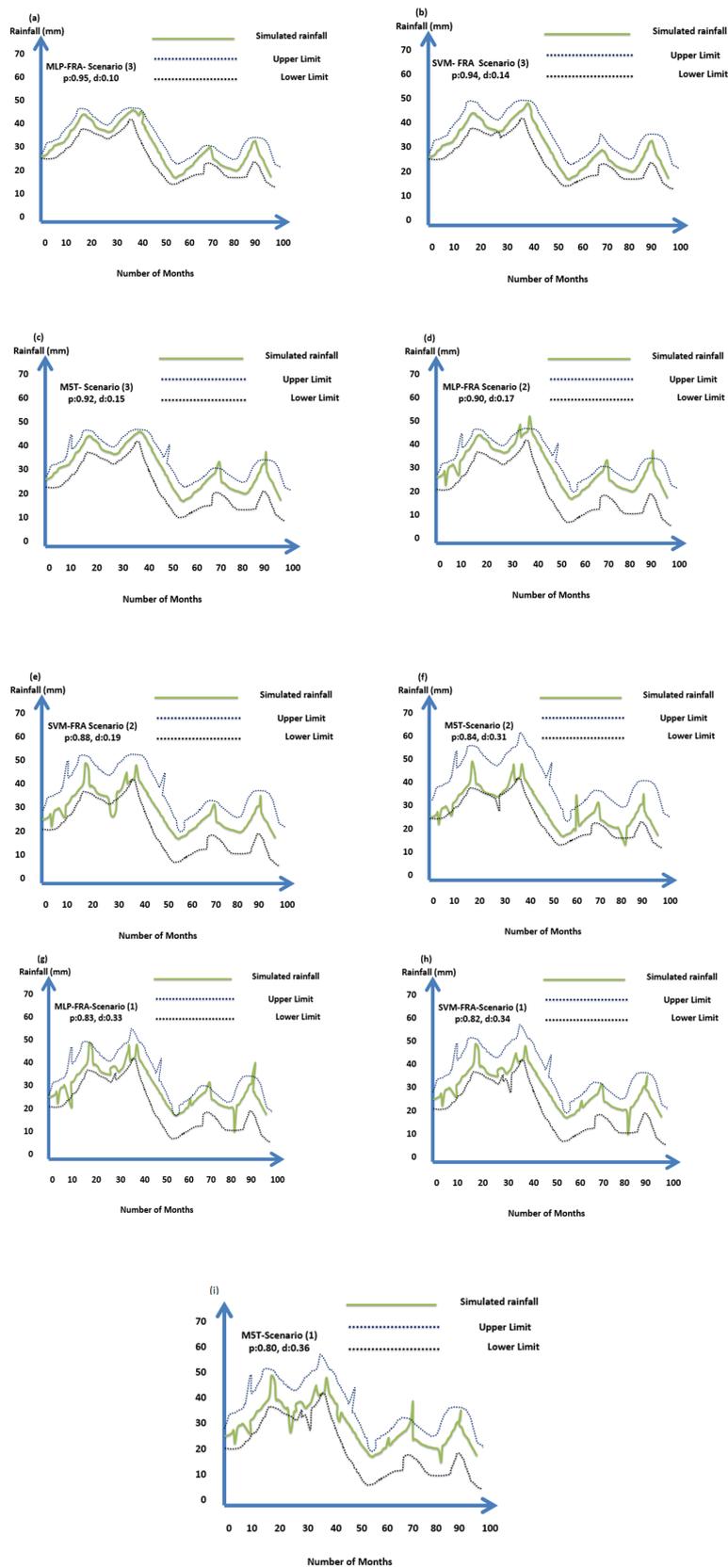


Figure 6. Performance of models under uncertainty for (a) MLP-FRA (Scenario 3), (b) SVM-FRA (Scenario 3), (c) M5T (Scenario 3), (d) MLP-FRA (Scenario 2), (e) SVM-FRA (Scenario 2), (f) M5T (Scenario 2), (g) MLP-FRA (Scenario 1), (h) SVM-FRA (Scenario 1), and (i) M5T (Scenario 1).

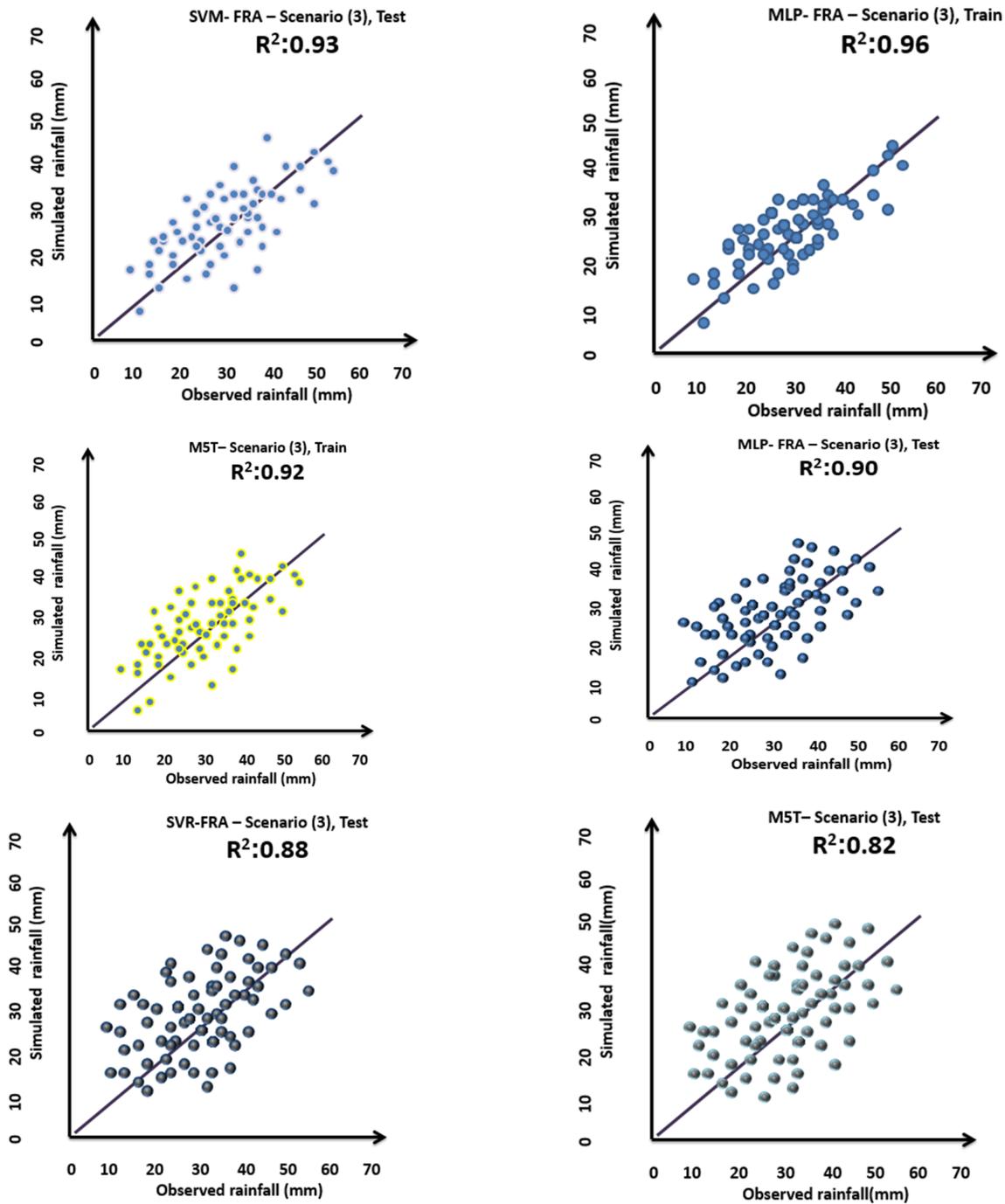


Figure 7. Performance of models according to the R^2 coefficient.

4.4. Hydrological Analysis of Results

Figure 8 shows the precipitation maps based on the MLP-FRA model. Two months were selected as samples to evaluate the models and compare with the observed data. The results indicated the good agreement between forecasted and based data. The Kappa coefficient was used to measure the agreement between observed data and forecasted data. The coefficient was computed based on the following equation:

$$Kappa : 1 - \frac{1 - p_0}{1 - p_e} \tag{17}$$

where p_0 is the relative measured agreement among raters and p_e is the hypothetical probability of chance agreement.

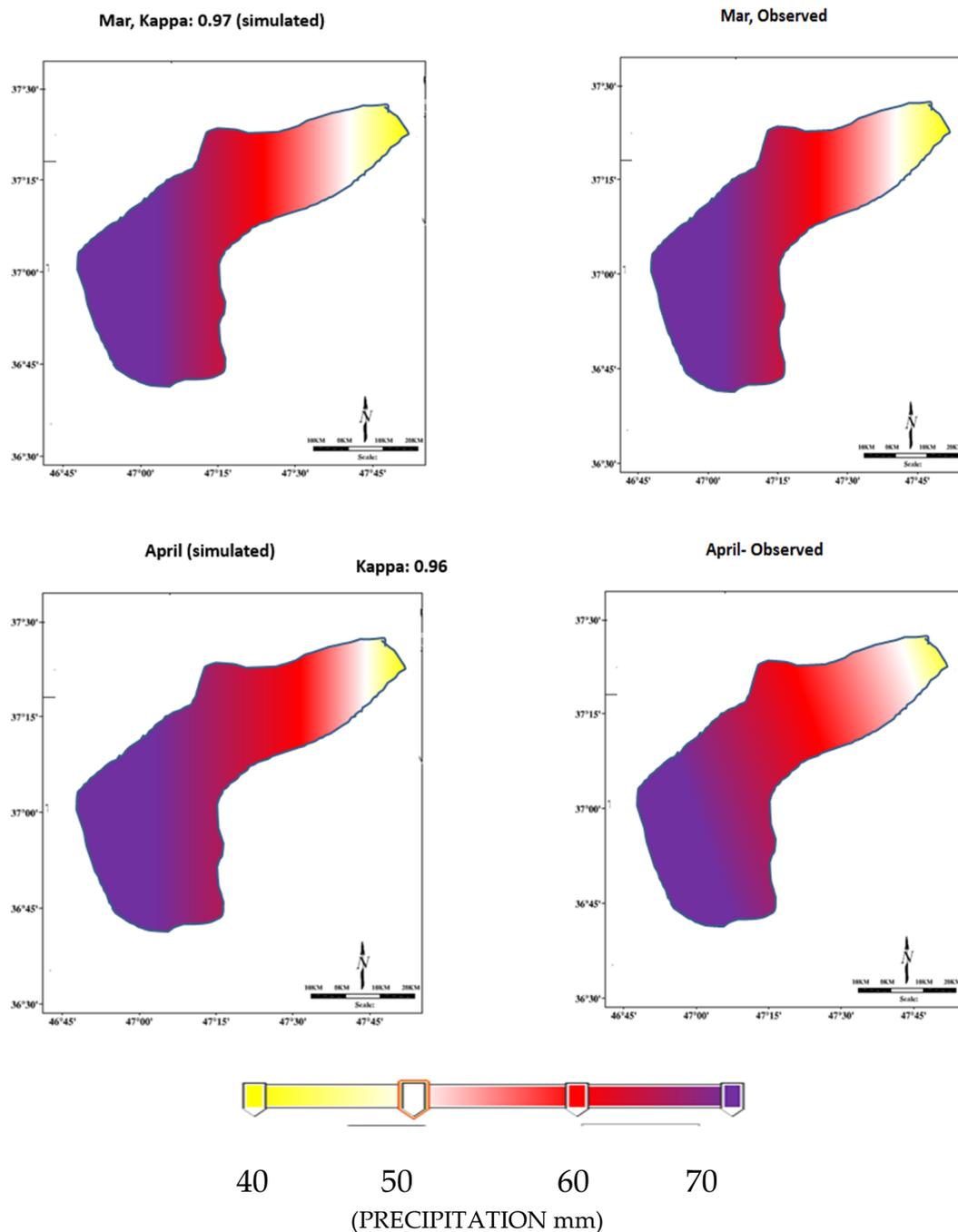


Figure 8. Precipitation zoning map 2000 to 2010 for months March and April.

P_0 is the value that has been calculated based on Equation (16) while P_e is the (assumed or supposed) probability of chance agreement, then finally we can calculate the Kappa coefficient, to represent the uncertainty pattern of the model output. In fact, the higher the value of the Kappa coefficient, the better the overall agreement level between the model output and the observed data. Keep in mind that the ratio of the agreement level in each category (prevalence) range should be computed at each observer’s accuracies.

In addition, it is observable that the shaded pattern for Figure 8 for the observed and simulated data for both months March and April are relatively very smooth. This is due to the fact that the

collected data is of relatively high-resolution which lead to less uncertainty for both the observed and simulated shaded pattern representation for the study area. It should be considered that the uncertainty analysis and spatial distribution of rainfall is necessary to prove that the MLP–FRA can be considered as an effective model. In addition, when the model has been examined using different inputs with different time series it achieved outstanding performance, proving that the model can act well with different and new input data.

5. Conclusions

Consecutive droughts and the need for water resources management make precipitation forecasting very important. The present study introduced two improved models of a multilayer neural network and support vector machine with a flow regime algorithm for precipitation forecasting. In addition, the results were compared with the tree model. Three scenarios were considered, including input temperature data with a delay of 1 to 12 months, input precipitation data with a delay of 1 to 12 months, and input temperature and precipitation data with a delay of 1 to 12 months. The results showed that compared to the SVR–FRA and M5T models, the MLP–FRA model reduced the MAE error rate by 18% and 8%, respectively. In addition, Scenario 3 was selected as the best scenario. Principal component analysis was used to select the influential inputs. Analysis of the uncertainties of the models used in this study showed that the MLP–FRA model had a better performance than the other models with an uncertainty bandwidth index of 0.10 and 95% of the data replacement within the range of confidence band in the test phase. Scenario 3 had a better performance than other models. Precipitation maps obtained with the MLP–FRA model were highly consistent with the observed values in all the months studied and indicated significant spring and winter precipitation. Given the presence of the Aidoghmush reservoir dam downstream of the basin, precipitation forecast is very important, because this information is useful for making decisions around water rationing in low-precipitation seasons. Future studies can investigate precipitation fluctuations under climate conditions using the MLP–FRA model.

Author Contributions: Formal analysis, F.B.B. and W.M.R.; Methodology, F.B.B., M.E., and A.E.-S.; Validation, C.M.F.; Writing—original draft, F.B.B. and A.N.A.; Writing—review & editing, H.A.A., W.M.R., and A.S.

Funding: This research was funded by the University of Malaya from research grant coded UMRG RP025A-18SUS and GPF082A-2018.

Acknowledgments: The authors appreciate so much the facilities support by the Civil Engineering Department, Faculty of Engineering, University of Malaya, Malaysia and Universiti Tenaga Nasional.

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

References

1. Ehteram, M.; Binti Koting, S.; Afan, H.A.; Mohd, N.S.; Malek, M.A.; Ahmed, A.N.; El-shafie, A.H.; Onn, C.C.; Lai, S.H.; El-Shafie, A. New Evolutionary Algorithm for Optimizing Hydropower Generation Considering Multireservoir Systems. *Appl. Sci.* **2019**, *9*, 2280. [[CrossRef](#)]
2. Ehteram, M.; Singh, V.P.; Ferdowsi, A.; Mousavi, S.F.; Farzin, S.; Karami, H.; Mohd, N.S.; Afan, H.A.; Lai, S.H.; Kisi, O.; et al. An improved model based on the support vector machine and cuckoo algorithm for simulating reference evapotranspiration. *PLoS ONE* **2019**, *14*, e0217499. [[CrossRef](#)] [[PubMed](#)]
3. Pradhan, A.M.S.; Lee, S.-R.; Kim, Y.-T. A shallow slide prediction model combining rainfall threshold warnings and shallow slide susceptibility in Busan, Korea. *Landslides* **2018**, *16*, 647–659. [[CrossRef](#)]
4. Lian, C.Y.; Huang, Y.F.; Ng, J.L.; Mirzaei, M.; Koo, C.H.; Tan, K.W. A proposed hybrid rainfall simulation model: Bootstrap aggregated classification tree–artificial neural network (BACT-ANN) for the Langat River Basin, Malaysia. *J. Water Clim. Chang.* **2019**, *jwc2019*, 294. [[CrossRef](#)]
5. Ghimire, S.; Deo, R.C.; Downs, N.J.; Raj, N. Global solar radiation prediction by ANN integrated with European Centre for medium range weather forecast fields in solar rich cities of Queensland Australia. *J. Clean. Prod.* **2019**, *216*, 288–310. [[CrossRef](#)]

6. Kisi, O.; Cimen, M. Precipitation forecasting by using wavelet-support vector machine conjunction model. *Eng. Appl. Artif. Intell.* **2012**, *25*, 783–792. [[CrossRef](#)]
7. Venkata Ramana, R.; Krishna, B.; Kumar, S.R.; Pandey, N.G. Monthly Rainfall Prediction Using Wavelet Neural Network Analysis. *Water Resour. Manag.* **2013**, *27*, 3697–3711. [[CrossRef](#)]
8. Shamshirband, S.; Gocic, M.; Petkovic, D.; Saboohi, H.; Herawan, T.; Mat Kiah, M.L.; Akib, S. Soft-Computing Methodologies for Precipitation Estimation: A Case Study. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *8*, 1–6. [[CrossRef](#)]
9. Kisi, O.; Sanikhani, H. Prediction of long-term monthly precipitation using several soft computing methods without climatic data. *Int. J. Clim.* **2015**, *35*, 4139–4150. [[CrossRef](#)]
10. Shenify, M.; Danesh, A.S.; Gocić, M.; Taher, R.S.; Abdul Wahab, A.W.; Gani, A.; Shamshirband, S.; Petković, D. Precipitation Estimation Using Support Vector Machine with Discrete Wavelet Transform. *Water Resour. Manag.* **2015**, *30*, 641–652. [[CrossRef](#)]
11. Arab Amiri, M.; Amerian, Y.; Mesgari, M.S. Spatial and temporal monthly precipitation forecasting using wavelet transform and neural networks, Qara-Qum catchment, Iran. *Arab. J. Geosci.* **2016**, *9*. [[CrossRef](#)]
12. Du, J.; Liu, Y.; Yu, Y.; Yan, W. A Prediction of Precipitation Data Based on Support Vector Machine and Particle Swarm Optimization (PSO-SVM) Algorithms. *Algorithms* **2017**, *10*, 57. [[CrossRef](#)]
13. Mirabbasi, R.; Kisi, O.; Sanikhani, H.; Gajbhiye Meshram, S. Monthly long-term rainfall estimation in Central India using M5Tree, MARS, LSSVR, ANN and GEP models. *Neural Comput. Appl.* **2019**, *31*, 6843–6862. [[CrossRef](#)]
14. Danandeh Mehr, A.; Nourani, V.; Karimi Khosrowshahi, V.; Ghorbani, M.A. A hybrid support vector regression–firefly model for monthly rainfall forecasting. *Int. J. Environ. Sci. Technol.* **2018**, *16*, 335–346. [[CrossRef](#)]
15. Azad, A.; Manoochehri, M.; Kashi, H.; Farzin, S.; Karami, H.; Nourani, V.; Shiri, J. Comparative evaluation of intelligent algorithms to improve adaptive neuro-fuzzy inference system performance in precipitation modelling. *J. Hydrol.* **2019**, *571*, 214–224. [[CrossRef](#)]
16. Kumar, D.; Singh, A.; Samui, P.; Jha, R.K. Forecasting monthly precipitation using sequential modelling. *Hydrol. Sci. J.* **2019**, *64*, 690–700. [[CrossRef](#)]
17. Nourani, V.; Tajbakhsh, A.D.; Molajou, A. Data mining based on wavelet and decision tree for rainfall-runoff simulation. *Hydrol. Res.* **2018**, *50*, 75–84. [[CrossRef](#)]
18. Hossain, I.; Rasel, H.M.; Imteaz, M.A.; Mekanik, F. Long-term seasonal rainfall forecasting using linear and non-linear modelling approaches: A case study for Western Australia. *Meteorol. Atmos. Phys.* **2019**, 1–11. [[CrossRef](#)]
19. Ehteram, M.; Karami, H.; Mousavi, S.F.; Farzin, S.; Celeste, A.B.; Shafie, A.-E. Reservoir Operation by a New Evolutionary Algorithm: Kidney Algorithm. *Water Resour. Manag.* **2018**, *32*, 4681–4706. [[CrossRef](#)]
20. Ehteram, M.; Mousavi, S.F.; Karami, H.; Farzin, S.; Deo, R.; Othman, F.B.; Chau, K.-w.; Sarkamaryan, S.; Singh, V.P.; El-Shafie, A. Bat algorithm for dam–reservoir operation. *Environ. Earth Sci.* **2018**, *77*, 510. [[CrossRef](#)]
21. Ehteram, M.; Singh, P.V.; Karami, H.; Hosseini, K.; Dianatikhah, M.; Hossain, M.; Ming Fai, C.; El-Shafie, A. Irrigation Management Based on Reservoir Operation with an Improved Weed Algorithm. *Water* **2018**, *10*, 1267. [[CrossRef](#)]
22. Poornima, S.; Pushpalatha, M. Prediction of Rainfall Using Intensified LSTM Based Recurrent Neural Network with Weighted Linear Units. *Atmosphere (Basel)* **2019**, *10*, 668. [[CrossRef](#)]
23. Min, M.; Bai, C.; Guo, J.; Sun, F.; Liu, C.; Wang, F.; Xu, H.; Tang, S.; Li, B.; Di, D.; et al. Estimating Summertime Precipitation from Himawari-8 and Global Forecast System Based on Machine Learning. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 2557–2570. [[CrossRef](#)]
24. Bhuiyan, M.A.E.; Nikolopoulos, E.I.; Anagnostou, E.N. Machine Learning-based Blending of Satellite and Reanalysis Precipitation Datasets: A Multi-regional Tropical Complex Terrain Evaluation. *J. Hydrometeorol.* **2019**, *20*, 2147–2161. [[CrossRef](#)]
25. Yadav, P.; Sagar, A. Rainfall Prediction Using Artificial Neural Network (ANN) for tarai Region of Uttarakhand. *Curr. J. Appl. Sci. Technol.* **2019**, *33*, 1–7. [[CrossRef](#)]
26. Bhuiyan, M.A.E.; Nikolopoulos, E.I.; Anagnostou, E.N.; Quintana-Seguí, P.; Barella-Ortiz, A. A nonparametric statistical technique for combining global precipitation datasets: Development and hydrological evaluation over the Iberian Peninsula. *Hydrol. Earth Syst. Sci.* **2018**, *22*, 1371–1389. [[CrossRef](#)]

27. Chen, H.; Chandrasekar, V.; Tan, H.; Cifelli, R. Rainfall Estimation From Ground Radar and TRMM Precipitation Radar Using Hybrid Deep Neural Networks. *Geophys. Res. Lett.* **2019**, *46*, 10669–10678. [[CrossRef](#)]
28. Pham, Q.; Yang, T.-C.; Kuo, C.-M.; Tseng, H.-W.; Yu, P.-S. Combining Random Forest and Least Square Support Vector Regression for Improving Extreme Rainfall Downscaling. *Water* **2019**, *11*, 451. [[CrossRef](#)]
29. Moosavi, V. Prediction of Rainfall as One of the Main Variables in Several Natural Disasters. *Adv. Nat. Technol. Hazards Res.* **2018**, *48*, 165–203.
30. Tahani, M.; Babayan, N. Flow Regime Algorithm (FRA): A physics-based meta-heuristics algorithm. *Knowl. Inf. Syst.* **2018**, *60*, 1001–1038. [[CrossRef](#)]



© 2019 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 (<http://creativecommons.org/licenses/by/4.0/>).