

RESEARCH

Open Access



Rainfall modeling using two different neural networks improved by metaheuristic algorithms

Saad Sh. Sammen¹, Ozgur Kisi^{2,3*}, Mohammad Ehteram⁴, Ahmed El-Shafie⁵, Nadhir Al-Ansari⁶,
Mohammad Ali Ghorbani⁷, Shakeel Ahmad Bhat⁸, Ali Najah Ahmed⁹ and Shamsuddin Shahid¹⁰

Abstract

Rainfall is crucial for the development and management of water resources. Six hybrid soft computing models, including multilayer perceptron (MLP)–Henry gas solubility optimization (HGSO), MLP–bat algorithm (MLP–BA), MLP–particle swarm optimization (MLP–PSO), radial basis neural network function (RBFNN)–HGSO, RBFNN–PSO, and RBFNN–BA, were used in this study to forecast monthly rainfall at two stations in Malaysia (Sara and Banding). Different statistical measures (mean absolute error (MAE) and Nash–Sutcliffe efficiency (NSE) and percentage of BIAS (PBIAS)) and a Taylor diagram were used to assess the models' performance. The results indicated that the MLP–HGSO performed better than the other models in forecasting rainfall at both stations. In addition, transition matrices were computed for each station and year based on the conditional probability of rainfall or absence of rainfall on a given month. The values of MAE for testing processes for the MLP–HGSO, MLP–PSO, MLP–BA, RBFNN–HGSO, RBFNN–BA, and RBFNN–PSO at the first station were 0.712, 0.755, 0.765, 0.717, 0.865, and 0.891, while the corresponding NSE and PBIAS values were 0.90–0.23, 0.83–0.29, 0.85–0.25, 0.87–0.27, 0.81–0.31, and 0.80–0.35, respectively. For the second station, the values of MAE were found 0.711, 0.743, 0.742, 0.719, 0.863 and 0.890 for the MLP–HGSO, MLP–PSO, MLP–BA, RBFNN–HGSO, RBFNN–BA, and RBFNN–PSO during testing processes and the corresponding NSE–PBIAS values were 0.92–0.22, 0.85–0.28, 0.89–0.26, 0.91–0.25, 0.83–0.31, 0.82–0.32, respectively. Based on the outputs of the MLP–HGSO, the highest rainfall was recorded in 2012 with a probability of 0.72, while the lowest rainfall was recorded in 2006 with a probability of 0.52 at the Sara Station. In addition, the results indicated that the MLP–HGSO performed better than the other models within the Banding Station. According to the findings, the hybrid MLP–HGSO was selected as an effective rainfall prediction model.

Keywords MLP, RBFNN, Probability matrix, Markov chain, Rainfall modelling

*Correspondence:

Ozgur Kisi

ozgur.kisi@th-luebeck.de; ozgur.kisi@iliauni.edu.ge

¹ Department of Civil Engineering, College of Engineering, University of Diyala, Diyala Governorate, Iraq

² Department of Civil Engineering, Technical University of Lübeck, 23562 Lübeck, Germany

³ Department of Civil Engineering, Ilia State University, 0162 Tbilisi, Georgia

⁴ Department of Water Engineering and Hydraulic Structures, Faculty of Civil Engineering, Semnan, Iran

⁵ Civil Engineering Department, Faculty of Engineering, University of Malaya, Kuala Lumpur, Malaysia

⁶ Civil, Environmental and Natural Resources Engineering, Lulea University of Technology, 971 87 Lulea, Sweden

⁷ Department of Civil Engineering, Istanbul Technical University, Ayazaga Campus, 34469 Maslak, Istanbul, Turkey

⁸ College of Agricultural Engineering and Technology, SKUAST-Kashmir, Srinagar 190025, India

⁹ Institute of Energy Infrastructure (IEI), Universiti Tenaga Nasional (UNITEN), 43000 Selangor, Malaysia

¹⁰ School of Civil Engineering, Faculty of Engineering, Universiti Teknologi Malaysia (UTM), 81310 Johor, Malaysia

Introduction

Water is one of the planet's most critical and life-sustaining resources [1, 2]. Water management is a significant concern due to rising water consumption and pollution [3, 4]. Accurate and timely rainfall forecasting is crucial for water resource management, irrigation planning, and reservoir operation [5]. Forecasting rainfall is crucial, because heavy rain frequently poses hydrological risks [6]. The nature of the atmospheric circulation system that generates rain makes rainfall forecasting difficult [7]. Physically based models commonly used for rainfall forecasting perform poorly when predicting rainfall 1 month in advance. However, rainfall forecasting 1 month in advance is crucial for water resource management. Consequently, hydrological modelers continuously attempt to develop a perfect model for rainfall prediction [8].

Soft computing techniques have been extensively used for parameter prediction and modeling in various fields of water resources, such as hydraulics [9], hydrology [10–12], and water quality [13, 14]. In addition, rainfall prediction methods based on soft computing have been frequently investigated in recent years [15, 16]. Soft computing models have been found to be quite effective at identifying nonlinear systems in a variety of engineering domains. Using various forms of soft computing, numerous studies have been conducted on rainfall forecasting. Rainfall was modeled by Mandal and Jothiprakash [17] utilizing a multilayer perceptron (MLP) neural network, a radial basis neural network (RBFNN), and a time-lagged recurrent network (TLRN). Regarding accuracy, the ANN models performed better than the regression models.

Hipni et al. [18] modeled rainfall using ANFIS and support vector machine (SVM) models. They observed that the ANFIS model could provide accurate estimates of rainfall. Nourani and Komasi [19] examined the performance of integrated geomorphological adaptive ANFIS (IGAANFIS) in modeling rainfall and reported that it outperformed the other models. Akrami et al. [20] compared the ANFIS and ANN rainfall prediction models. According to their findings, the ANFIS provided greater accuracy than the ANN model. They utilized various time delays between time (t) and time ($t-4$) (scale: month).

Mehdizadeh et al. [21] estimated monthly rainfall time series using gene express programming autoregressive conditional heteroscedasticity. The outcomes demonstrated that the new method was more precise than gene expression programming. Mehr and Nourani [22] compared multi-gene-genetic programming (MGGP) and genetic programming (GP) for estimating seasonal rainfall and demonstrated that the MGGP is more accurate than the GP. Ouyang and Lu [23] predicted monthly rainfall using MGGP, echo state network, and SVM models.

In terms of predicting monthly rainfall, the results indicated that the MGGP performed poorly compared to other models.

Recent research has demonstrated that combining soft computing models with preprocessing and optimization algorithms can boost their performance. Adiwjaya (2015) improved rainfall forecasting by combining ANN with evolutionary algorithms. Feng et al. [24] combined the wavelet technique with SVM and ANN and demonstrated that the wavelet SVM (WSVM) model outperformed others. Yaseen et al. [25] investigated the application of integrated ANFIS model with the firefly optimization algorithm (FFA) and revealed that the FFA-optimized ANFIS is more accurate in stream-flow forecasting than standalone ANFIS. Mehr et al. [26] investigated the combination of SVM and FFA for rainfall forecasting. According to the results, the new model generated accurate precipitation forecasts.

Pham et al. [27] developed ANN–particle optimization (PSO) and SVM–PSO models for predicting rainfall. The results indicated that the SVM–PSO was the most accurate model. Basha et al. [28] predicted rainfall using autoregressive integrated moving average (ARIMA), ANN, and SVM. The outcomes revealed that the ANN model outperformed the ARIMA model. Akner (2021) developed an ANN model for rainfall forecasting. The Levenberg–Marquardt algorithm was employed to adjust ANN parameters. The study reported that the ANN provided accurate results. Ahmed et al. (2021) developed different hybrid models including MLP–HGSO, MLP–NRO and MLP–EO for estimation of stream flow for High Aswan dam. They used data of 130 years for training and testing of the developed models. They concluded that the NRO was the best optimization algorithm that improves the capability of MLP model for hydrological modeling. Ruma et al. [29] used PSO to enhance the performance of ANN and LSTM for water level prediction in one of Bangladesh rivers. They concluded that the PSO could enhance the performance of the LSTM better than the ANN.

Wang et al. [30] combined wavelet packet decomposition (WPD) and an extreme learning machine (ELM) model. They stated that the WPD enhanced the ELM model's effectiveness. Kumar et al. [31] coupled biogeography-based optimization (BBO) models with ELM and deep neural network models. They selected the inputs using partial autocorrelation function (PACF). They reported that the ELM–BBO was a reliable tool for risk mitigation. Li et al. [32] designed the pruned ELM (OP–ELM) to forecast rainfall. Different activation functions of the OP–ELM model were utilized. The OP–ELM was highly capable of predicting rainfall.

The review above indicates that the soft computing model integrated with an optimized algorithm performs better than the standalone model in terms of rainfall modeling. However, the performance of such a hybrid model varies greatly depending on the optimization algorithm chosen. Consequently, it is essential to evaluate the performance of various optimization algorithms after their incorporation into soft computing models. In this study, new and conventional optimization algorithms were compared in terms of their performance in enhancing the accuracy of soft computing models for rainfall forecasting.

The RBFNN has numerous benefits. While MLP models can have multiple hidden layers, the RBFNN models have one. Therefore, it is faster and can converge sooner. In addition, RBFNN models offer simplicity of implementation and high precision. There are various machine learning models for predicting rainfall, such as optimized ANFIS and ELM models. However, adjusting these models' unknown parameters may be time-consuming and challenging. These unknown parameters may increase the values of model uncertainty. Consequently, the current study employs optimized MLP and RBFNN models for rainfall forecasting.

This study evaluates the effectiveness of new optimization algorithms in training ANNs and analyzes the uncertainty of new RBF and MLP models. Monthly rainfall with varying time lags was used as input parameters to predict rainfall 1 month in advance. Several models were developed using different input combinations that include time lags ranging from months ($t-1$) to ($t-9$), where ($t-1$) was used as input for the first model and ($t-1$) and ($t-2$) for the second model and so on. The main objectives and novelties of the current paper are as follows:

- (1) Developing a new hybrid models of MLP and RBFNN using meta-heuristic algorithms, Henry gas solubility optimization (HGSO), bat algorithm (BA) and particle swarm optimization (PSO) for predicting long-term rainfall.
- (2) Developing the transient probability matrix (TPM) rainfall for identifying rainy days, where the TPM can be used as an early warning system to identify and motoring of the flood and drought periods.
- (3) The principal component analysis (PCA) was used to preprocess the data, where it is considered as the best tool to decide the best-input data.

Materials and methods

Materials

Study area

In Peninsular Malaysia, the Sara Station (station A) is located at latitude 5° 25' and longitude 101° 25', while the Banding Station (station B) is located at latitude 5°

30' and longitude 101° 30'. Figure 1 depicts the locations of stations on a map of peninsular Malaysia. According to the Koppen classification, both stations have climates characterized by tropical rainforests. The average monthly rainfall was obtained from station A and station B for the period January 2001 to January 2014 to develop the models. The monthly rainfall time series for both stations are depicted in Fig. 2a, b. The models developed in the current study can be applied to other global regions. These models can predict other variables, such as groundwater level, temperature, runoff, and sediment. In addition, these models can be used to predict various variables under climate change conditions. Since the water resource management is important for the basin, these stations are chosen.

Data preprocessing and input selection

Principal component analysis (PCA) was used to reduce a large number of input variables to a manageable number [33, 34]. PCA transforms a collection of correlated

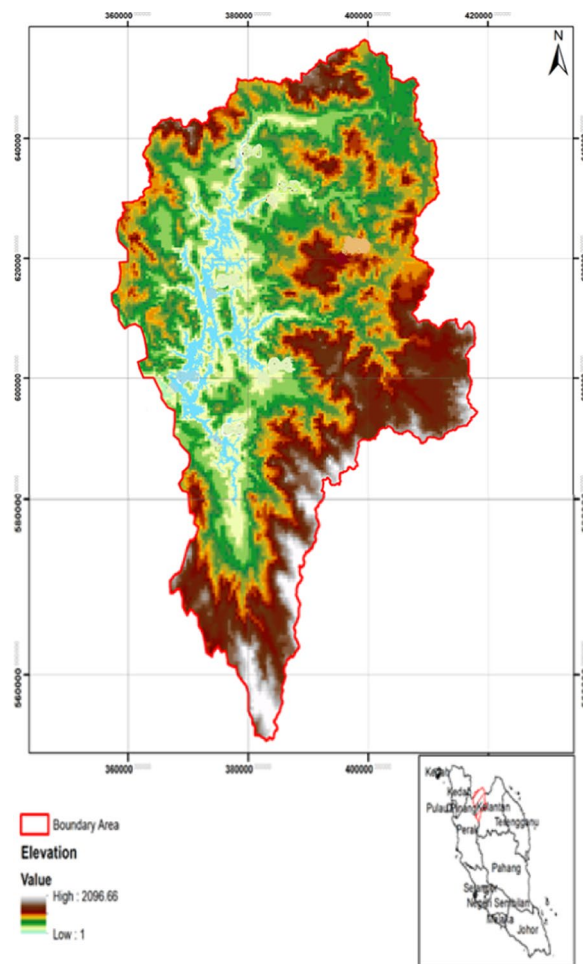
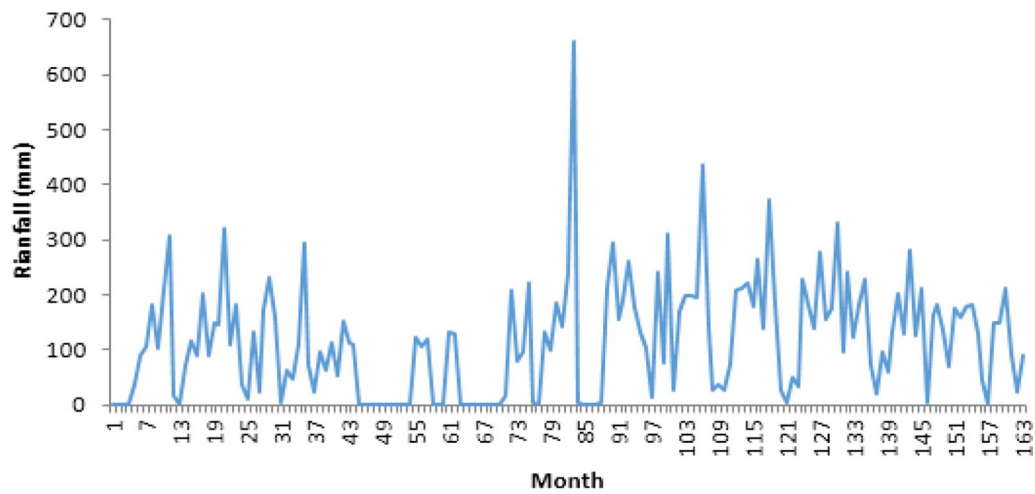
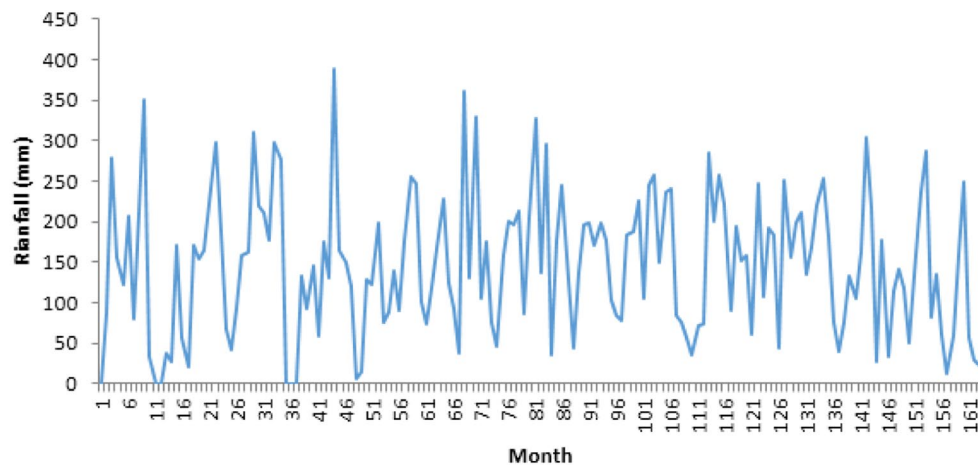


Fig. 1 Location map of rainfall stations



a: Time series for Sara Station



b: Time series for Banding Station

Fig. 2 Time series of rainfall stations

variables into a collection of principal components (uncorrelated variables). The initial PCA accounts for as much information variability as possible. PCA searches for a linear combination of input variables to extract the maximum data variance. This study used PCA to select inputs in various time lags from month ($t-1$) to month ($t-9$).

Methods

Artificial neural network (ANN) models

ANN is a technique based on the workings of the human nervous system [35]. MLP is one of the most significant ANN models for predicting meteorological variables. It is composed of individual processing units

called neurons. The neurons in the MLP structure are arranged in distinct layers. When applied to problems, ANNs do not require prior knowledge of atmospheric conditions [36]. They can circumvent the limitations of empirical models. The neurons of the first layer receive the input variables (x_0). Neurons can learn complicated features via hidden layers. A weight (w_0) associated with each input is considered in the structure of the MLP model. Each neuron also has a bias. Additional inputs to neurons are classed as biases (b). By generating weight values, ANNs build associations between input and output data. The MLP output is obtained by applying the nonlinear activation function to the weighted input sum (Fig. 3).

The weights and biases must be updated to minimize the error predictions. Different algorithms are used to optimize the network parameters. The learning algorithms can be categorized into two groups. The first group employs optimization algorithms, while the second employs conventional numerical optimization techniques. Although numerous optimization algorithms for training ANNs have been studied in the past, the problem of local minima remains unsolved. Consequently, novel MLP and RBFNN methods based on new optimization algorithms are proposed in this article [37].

Henry gas solubility optimization (HGSO)

Hashim et al. [38] introduced a new optimization technique based on Henry’s law, termed HGSO. The algorithm’s performance was evaluated on benchmark functions, a speed reducer design problem, and real-world engineering difficulties. In the experiments, particle swarm optimization (PSO), genetic algorithm (GA), gravitational search algorithm (GSA), whale optimization algorithm (WHA), and cuckoo search algorithm (CSA) performed poorly. HGSO can imitate Henry’s law behavior. Local optimums are omitted through this method. Henry’s law determines how soluble low-solubility gases are in liquids. First, the population (number of gases) and position of gases are initialized using the following equation:

$$X_i(t + 1) = X_{\min} + r(X_{\max} - X_{\min}) \tag{1}$$

where X_i denotes the position of i th gas, N is the population size, r represents a random number, X_{\min} and X_{\max} are bounds of the problem, t is the iteration number, and $X_i(t + 1)$ denotes the new position at the $t + 1$ iteration.

The population gases are divided into equal categories based on their number. Because the gases in each group are comparable, Henry’s constant value for each group is the same. Every gas has its own unique Henry’s law constant.

Henry’s law constant for every gas is proportional to the gas concentration in liquid form and partial pressure in the gas phase. Henry’s law states that the amount of dissolved gas in a given volume of liquid is proportional to its partial pressure above the liquid. Each cluster j is analyzed to determine the optimal gas in its class with the highest equilibrium phase. Different ranks of gases are used to identify the optimal gas for the swarm. Equation (2) is used to update Henry’s coefficient:

$$H_j(t + 1) = H_j(t) \times \exp\left(-C_j \times \left(\frac{1}{T_t} - \frac{1}{T^\theta}\right)\right),$$

$$T_t = \exp\left(\frac{-t}{\text{iter}}\right)H_j(t) = l_1 \times \text{rand}(0, 1),$$

$$P_{ij} = l_2 \times \text{rand}(0, 1), C_j = l_3 \times \text{rand}(0, 1) \tag{2}$$

where H_j denotes Henry’s coefficient of cluster j , T represents temperature, T^θ is constant (298.5), P_{ij} represents partial pressure, C_j denotes enthalpy of dissolution, l_1 is constant (0.05), l_2 is constant (100), and l_3 is constant (0.01). The solubility is updated based on the following equation:

$$S_{ij}(t) = K \times H_j(t + 1) \times P_{ij}(t) \tag{3}$$

where $S_{ij}(t)$ denotes gas i ’s solubility in cluster j ’s, P_{ij} represents gas i ’s partial pressure in cluster j . The position of gases is updated using the following equation:

$$X_{i,j}(t + 1) = X_{i,j}(t) + F \times r \times \gamma \times (X_{i,\text{best}}(t) - X_{i,j}(t))$$

$$+ F \times r \times \alpha \times (S_{i,j}(t) \times X_{i,\text{best}}(t) - X_{i,j}(t))$$

$$\gamma = \beta \times \exp\left(-\frac{F_{\text{best}}(t) + \varepsilon}{F_{i,j}(t) + \varepsilon}\right), \varepsilon = 0.5 \tag{4}$$

where $X_{i,j}$ denotes the position of gas i in cluster j , r is a random constant, t is iteration time, $X_{i,\text{best}}(t)$ denotes the optimal gas, $F_{i,j}$ is the objective function of gas i in

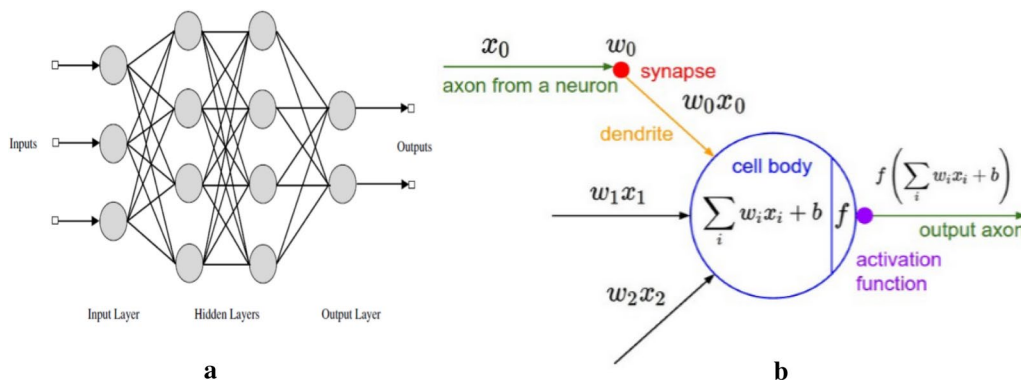


Fig. 3 a Computation process in the MLP and b The details of the structure of the MLP model

cluster j , $X_{i,best}(t)$ denotes the optimal gas in the group, β is constant, F_{best} is the objective function of the optimal gas, $F_{i,j}$ represents the objective function of gas i in cluster j , α is the effect of other gases on gas i in cluster j equal to 1, and F denotes the flag that alters the search gas' direction. Finally, Eq. (9) is used to rank and choose the quantity of the worst gases:

$$N_u = N \times (\text{rand}(c_2 - c_1) + c_1), c_1 = 1, c_2 = 0.2 \quad (5)$$

where N_u denotes the number of worst gases, and N represents the number of gases.

Equation (10) is used to update the worst gases' position:

$$G_{i,j} = G_{\min(i,j)} + r \times (G_{\max(i,j)} - G_{\min(i,j)}) \quad (6)$$

where $G_{i,j}$ denotes gas i 's position in cluster j , r is a random number, and $G_{\max(i,j)}$ and $G_{\min(i,j)}$ are the bounds of the problem.

Particle swarm optimization (PSO)

PSO is utilized in various optimization domains and integrated with other algorithms. This algorithm uses particles to determine the optimal solution. The achieved and optimal swarm positions are utilized to identify each particle. Equations (7, 8) change the positions of the particles as follows [39]:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (xBest_i^t - x_i^t) + c_2 r_2 (gBest_i^t - x_i^t) \quad (7)$$

$$x_i^{t+1} = x_i^t + v_i^t \cdot t \quad (8)$$

where $xBest$ denotes the best particle position, g represents the best group position, ω is inertia weight, c_1 and c_2 are acceleration coefficients, and ω denotes the inertia coefficient. PSO's stochastic features ensure that the solution space is exploited.

Bat algorithm (BA)

The BA is an optimization algorithm based on swarm intelligence introduced by Yang [40]. The algorithm mimics the echolocation behavior of bats.

Bats employ echolocation to locate prey and avoid obstacles. Bats utilize the temporal difference between emission and environmental reflection to locate prey. The bats' volume decreases from a high positive value to a low constant value. Virtual bats are used to produce novel solutions for the bat algorithm:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (9)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i \quad (10)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (11)$$

where v_i^t denotes velocity, x_i^t represents location, t is iteration, x_* denotes the optimal solution, β is a random value, and f_{\min} and f_{\max} are min and max frequency, respectively.

The loudness and pulsation rate must be changed to manage the exploitation and exploration stages. To this end, Eq. 16 is used to calculate the pulsation rate and loudness parameters:

$$\begin{aligned} A_i^{t+1} &= \alpha A_i^t \\ r_i^{t+1} &= r_i^0 \cdot (1 - e^{-\gamma t}) \end{aligned} \quad (12)$$

where A_i^{t+1} denotes the average loudness of all bats, r_i^{t+1} represents the pulsation rate at time $t+1$, α and γ are a constant and initial loudness, respectively, and r_i^0 is the initial pulsation rate.

Locally, the random walk is used to determine a new position for each bat:

$$x_{new} = x_* + \varepsilon A^t \quad (13)$$

Optimization algorithm for training RBFNN and MLP

Each candidate ANN (MLP and RBFNN) is encoded as a dimensional vector for each member (particle, gas, and bat). Vectors consist of a set of weights connecting the first layer to the hidden layer, a set of weights connecting the second layer to the final layer, a set of biases, a set of widths of hidden neurons, and a set of centers of hidden neurons. To assess the objective function value of the candidate RBFNN and MLP networks, the RMSE is computed over all training data for each member. The workflow of the optimization algorithms used in this article to train the MLP and RBFNN is outlined in the steps below:

1. Initialization: A predefined number of particles, bats, and gases show a possible RBFNN and MLP network.
2. An objective function evaluates the equality of generated RBFNN and MLP networks. The provided search agents (weights, biases, widths, and centers) are first assigned to MLP and RBFNN networks, and then the networks are evaluated. The RMSE is a popular objective function in integrated ANNs. Based on the training data, the training algorithm attempts to construct the MLP and RBFNN with the lowest RMSE value possible.
3. The position of each agent is updated.

- Levels 2 to 3 are iterated until the maximum number of iterations is achieved. Finally, the RBFNN and MLP models with the minimum RMSE are tested on the unobserved part of the data.

The current study’s MLP–HGSO model contains one hidden layer. There were four hidden neurons. A process of trial and error was used to determine the number of hidden neurons and layers. Moreover, there were four hidden neurons in RBFNN–HGSO, obtained using a trial-and-error method (Fig. 4).

Evaluation of models performance

To evaluate and compare the results of the developed models in this study, different statistical indices were used, including mean absolute error (MAE), percentage

of BIAS (PBIAS) and Nash–Sutcliffe efficiency (NSE). The equations of these statistical indices are shown as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^n (y_i - y_i^-) \tag{14}$$

$$PBIAS = \sum_{i=1}^n (y_i - y_i^-) * 100 / \sum_{i=1}^n y_i \tag{15}$$

$$NSE = 1 - \left[\frac{\sum_{i=1}^n (y_i - y_i^-)^2}{\sum_{i=1}^n (y_i - y_{mean})^2} \right] \tag{16}$$

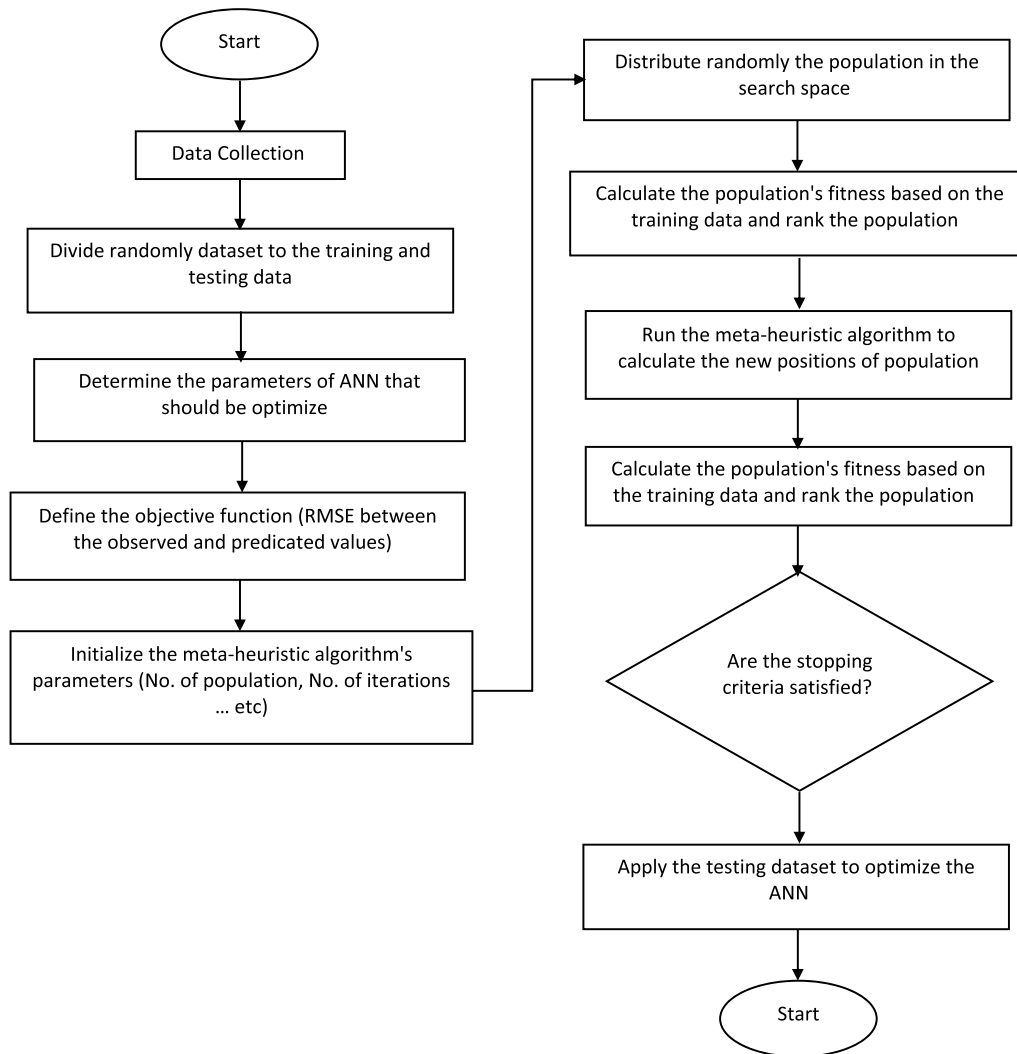


Fig. 4 Flowchart of the developed models

where y_i is the observed value, y_i^- is the predicted value, y_{mean} is the mean value of the data and n is the total number of data [39, 41, 42].

Results and discussion

Taguchi model

The Taguchi model is one of the commonly applied methods for adjusting the parameters of optimization algorithms. In this model, some experiments are considered based on different integrations of some effective parameters to achieve the best integrations of effective parameters. Taguchi’s model uses some special orthogonal arrays (OR). The OR are special cases of the fractional factorial design. In this model, there are two important parameters: noise parameters and controllable parameters. Analyzing the results employs the signal-to-noise ratio. To implement the Taguchi model for adjusting the parameters of optimization algorithms, it is necessary to identify the parameters with the most significant impact on the output. Table 1 outlines the algorithm’s parameters that were determined for this purpose. To conduct the experiments in the Taguchi model, it is necessary to determine the parameter values that result in good fitness.

Consequently, the parameter values are determined using a trial-and-error approach. Next, considering the available degrees of freedom, an appropriate orthogonal array is selected. Since the minimization problem is considered, the signal-to-noise ratio can be calculated through the following equation:

$$\frac{S}{N} = -10 \log (\text{objective (function)}). \tag{17}$$

In this study, 4, 5, and 2 parameters were selected for the PSO, BA, and HGSO, respectively, using the Taguchi model with four levels. Based on L16, L16, and L16 orthogonal arrays, 16 experiments were performed for the PSO, BA, and HGSO parameters, respectively. The outputs of tuning parameters using the Taguchi model are presented in Table 1.

PCA results

Inputs to the various soft computing models were selected based on the past monthly average rainfall.

Table 1 Parameters of the algorithm implemented in this study

PSO	Population size: 100, c_1 : 2, c_2 and w : 0.9
BA	Population size: 100, maximum frequency: 0.9, minimum frequency: 0.3, maximum loudness: 7 and minimum loudness: 0
HGSO	Population size: 100, number of clusters: 10

As shown in Table 2, the total number of experimental input data is nine, including rainfall at $t-1$ (previous month) to $t-9$ (9 previous months). The PCA was applied to data from 2001 to 2014 to obtain PCs and reduce the dimension of the input data. Table 3 displays the selected important components. Eigenvalues larger than one indicate significant components, where significant components must be retained. The variance described indicates what percentage of its component’s data can describe the variability of raw rainfall inputs. It was determined that the first five PCs accounted for 80% of the total variance, so the input was set to 5. Table 3 displays the loading matrix of the input variables for these components. Greater loading indicates that input data describe more information. The results indicate that the first five components are strongly correlated with the five sets of lagged input data [$R(t-1)$, $R(t-2)$, $R(t-3)$, $R(t-4)$, and $R(t-5)$].

Table 2 Input data and Eigen vectors

Inputs	%	%	
$R(t-1)$, $R(t-2)$, $R(t-3)$, $R(t-4)$, $R(t-5)$, $R(t-6)$, $R(t-7)$, $R(t-8)$ and $R(t-9)$			
*Components	Eigen values	Variance explained	
		Cumulative variance explained	
1	3	0.33	33
2	2	0.22	55
3	1	0.12	67
4	0.75	8.3	75
5	0.75	8.3	83
6	0.5	5.5	88.5
7	0.4	4.4	93
8	0.3	3.3	96.6
9	0.3	3.3	100

Table 3 Loading matrix for input variables

Input	PC1	PC2	PC3	PC4	PC5
$R(t-1)$	0.93	0.90	0.87	0.84	0.82
$R(t-2)$	0.89	0.86	0.83	0.81	0.80
$R(t-3)$	0.87	0.84	0.82	0.78	0.77
$R(t-4)$	0.83	0.82	0.81	0.76	0.75
$R(t-5)$	0.81	0.80	0.78	0.75	0.72
$R(t-6)$	0.72	0.70	0.72	0.71	0.70
$R(t-7)$	0.65	0.64	0.63	0.60	0.58
$R(t-8)$	0.61	0.60	0.59	0.57	0.55
$R(t-9)$	0.60	0.58	0.55	0.52	0.50

Statistical results for soft computing models

The PSO, BA, and HGSO were combined with a standalone MLP model to develop new models for forecasting monthly rainfall at two stations based on antecedent values. Three important indices including the MAE, NSE, and PBIAS were used. They illustrate the disparity between measured and observed data. These indices serve to assess the performance of models. In addition, the calculation of these indices is straightforward.

As shown in Table 4, using the MLP–HGSO model, as an illustration, the MAE value for station 1 during the testing period is 0.712 mm. Table 4 demonstrates that the MLP–HGSO model is more accurate than the other soft computing models. Table 4 clearly demonstrates that the MLP–BA model outperforms the MLP–PSO model. Regarding the obtained results of MLP and RBFNN models for two stations, it could be concluded that the MLP outperforms the RBFNN. The MAE=0.812 mm and PBIAS=0.27 values reflected in the RBFNN–HGSO

outputs for the testing level exceeded the accuracy level of the MLP–HGSO developed with the same inputs.

In terms of the RMSE and PBIAS indices, Table 4 shows that the MLP–HGSO model has the lowest values when compared to other applied soft computing models. According to Table 4, the RBFNN–HGSO model with lower MAE and PBIAS values is more accurate than the other RBFNN models. However, the MLP–HGSO model has the lowest RMSE and PBIAS values among the other models. In addition, Table 4 demonstrates that the RBFNN–HGSO model outperforms the RBFNN–PSO and RBFNN–BA across all performance metrics. Moreover, the HGSO appears to be a more effective optimization tool than the PSO and BA for improving results.

Taylor diagram (TD) for soft computing models

TDs are graphical representations of mathematical models that highlight the most realistic models. TD is used to evaluate the degree of accuracy between simulated and observed data using standard deviation, RMSE, and Pearson’s coefficient of correlation. According to the findings, the MLP–HGSO model is more accurate than the other soft computing models (Fig. 5). The MLP–HGSO and RBFNN–HGSO are found to be closer to the reference point (observed data).

Gradient color curves

Figure 6 depicts the RMSE–observation standard deviation ratio (RSR) values. As can be seen, the maximum MLP–HGSO RSR values at stations 1 and 2 are 0.25 and 0.2, respectively. The HGSO model outperforms the BA and PSO models. According to a previous literature review (Braga et al. 2018), model performance is deemed “satisfactory” if $RSR \leq 0.7$ [43]. Various reports indicate that the model’s performance can be deemed “very good” if $RSR \leq 0.5$. In addition, the R^2 analysis indicates that the MLP–HGSO model is more closely aligned with the observed data compared to other models.

Transient probability matrix rainfall

The Markov chain is a mathematical model that explains stochastic processes and offers probability information on phase transitions. In n time steps, the probability of moving from state i to state j is given as [17]:

$$P[X_{t+1} = i_{t+1} | X_t = i_t, \dots, X_0 = i_0] = \Pr [X_{t+1} = i_{t+1} | X_t = i_t] \tag{18}$$

where X_t denotes a random variable, $i_0, i_1, i_2, \dots, i_{t+1}$ represents the state at each time and $P(x_{t+1} = j | X_t = i) = p_{ij}$ is the transition probability from phase i at time t to phase j at time $t + 1$:

Table 4 Statistical indexes for different soft computing models

Index	MAE, mm	NSE	PBIAS
Station (1) (Train)			
MLP–HGSO	0.687	0.95	0.17
MLP–PSO	0.727	0.89	0.22
MLP–BA	0.825	0.91	0.19
RBFNN–HGSO	0.722	0.93	0.16
RBFNN–BA	0.924	0.87	0.26
RBFNN–PSO	0.941	0.84	0.28
Station (1) (Test)			
MLP–HGSO	0.712	0.90	0.23
MLP–PSO	0.755	0.83	0.29
MLP–BA	0.765	0.85	0.25
RBFNN–HGSO	0.717	0.87	0.27
RBFNN–BA	0.865	0.81	0.31
RBFNN–PSO	0.891	0.80	0.35
Station (2) (Train)			
MLP–HGSO	0.683	0.97	0.15
MLP–PSO	0.725	0.90	0.21
MLP–BA	0.723	0.92	0.20
RBFNN–HGSO	0.691	0.93	0.17
RBFNN–BA	0.914	0.89	0.22
RBFNN–PSO	0.930	0.86	0.23
Station (2) (Test)			
MLP–HGSO	0.711	0.92	0.22
MLP–PSO	0.743	0.85	0.28
MLP–BA	0.742	0.89	0.26
RBFNN–HGSO	0.719	0.91	0.25
RBFNN–BA	0.863	0.83	0.31
RBFNN–PSO	0.890	0.82	0.32

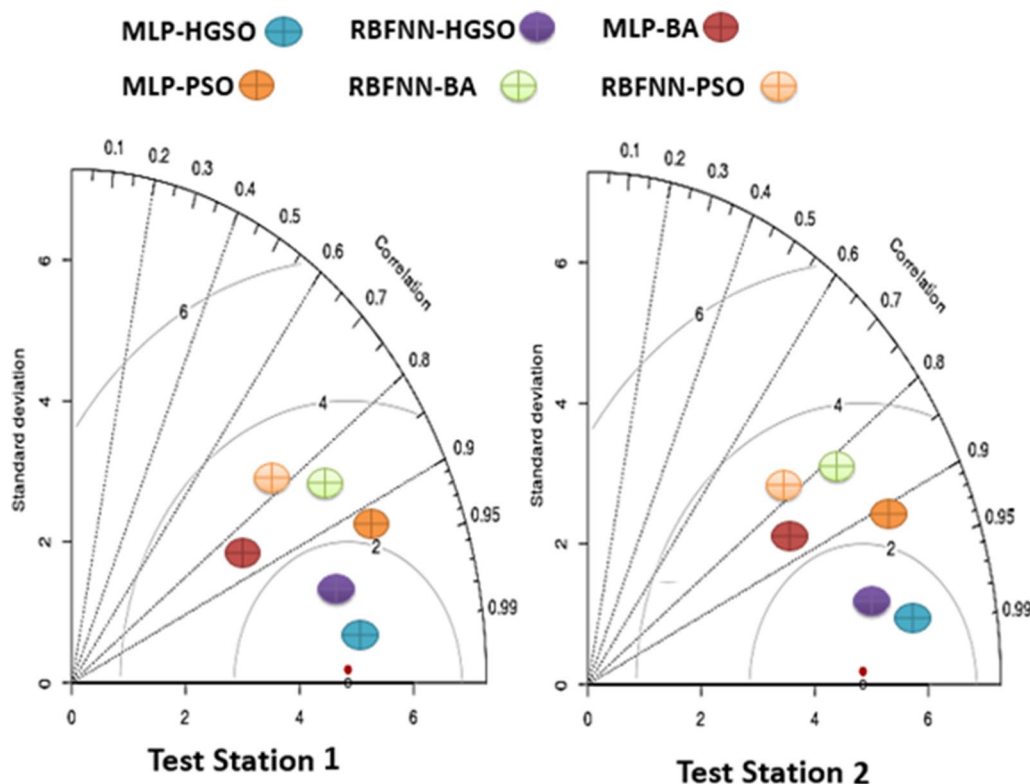


Fig. 5 TD for soft computing models

$$p = [p_{ij}] = \begin{bmatrix} p_{11} & p_{1s} \\ p_{s1} & p_{ss} \end{bmatrix} \tag{19}$$

where $0 \leq p_{ij} \leq 1, \sum_{j=1}^s p_{ij} = 1, s$: the (number) of (state)

Given the strong performance of the newly implemented hybrid MLP-HGSO model, future research could include the hybrid MLP-HGSO model to forecast other hydrological variables in the short and long run. To avoid duplication of discussions, station 1 was selected as a representative example for analyzing rainfalls. A state matrix, which is a row matrix, displays the probabilities for each state. For instance, the steady-state matrix (0.77, 0.23) indicates that station 1’s precipitation follows the probability distribution (Table 5).

In 2001, the probability of precipitation on a given month at station 1 was 0.77, and the probability of dryness on a given month at station 1 was 0.23. The predicted monthly rainfall used in this article was derived from the observed data at stations 1 and 2 for the years 2001–2014, a period of 13 years. Notably, MLP-PSO and RBFNN-PSO were omitted from Table 5 due to their significantly inferior estimates compared to MLP-HGSO, RBFNN-HGSO, MLP-BA, and RBFNN-BA. The results indicate that the rainfall pattern of MLP-HGSO closely resembles those of observed data. For

example, the predicted rainfall values of MLP-HGSO and observed data reveal that the highest rainfall for station 1 occurred in 2001 with a probability of 0.77 and that the highest rainfall for station 2 occurred in 2007 and 2009 with the same probability. Other values also demonstrate that the MLP-HGSO model is superior to other alternatives.

When modelers encounter incomplete, noisy, or small data sets, signal decomposition is a useful performance-enhancing technique. Optimization algorithms, preprocessing techniques, and decomposition methods can improve the performance of ML models. Our data are reliable and complete; thus, optimization and principal component analysis are used to enhance model performance. The models produced the best results without error index-based decomposition techniques. A set of input variables carefully selected can aid forecasting models in capturing the nonlinear characteristics of run-off data. However, ensemble empirical mode decomposition with the MLP model is combined to determine the difference between MLP-EMD and MLP models.

The ensemble empirical mode decomposition (EEMD) is a useful tool for preprocessing data. The EMD generates a finite number of oscillatory modes from the original time series data. Every oscillatory

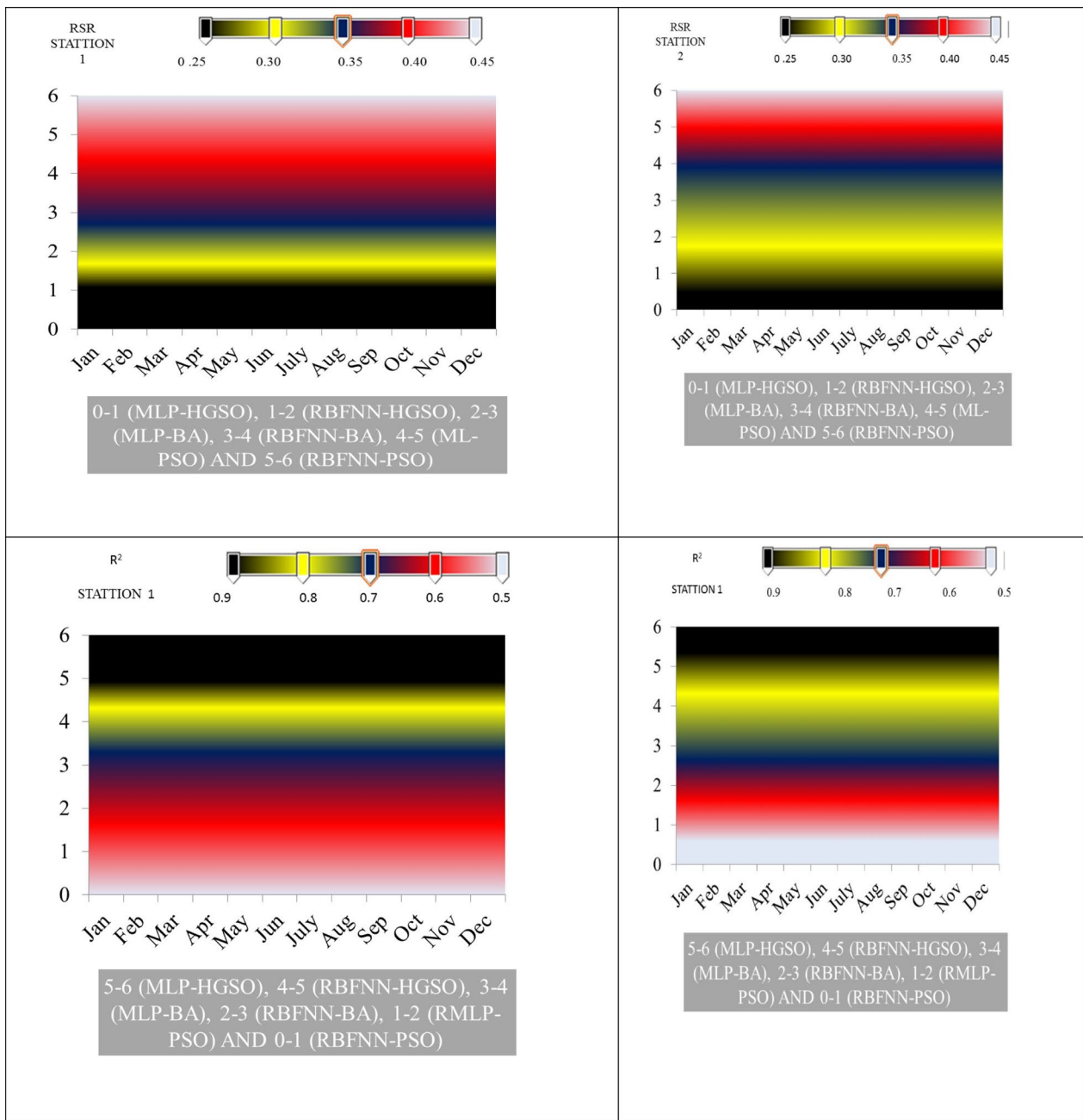


Fig. 6 Gradient color curves for soft computing models

mode is described by an intrinsic mode function (IMF). Depending on the data-driven mechanism, the IMF can be derived directly from the data. An EMD algorithm converts a time series into IMF modes through a shifting process. More details about the EMD can be found in Wang et al. (2015). The results of MLP-HGSO with and without EMD are displayed in Table 3. As shown in the table, MLP-HGSO-EMD does not significantly improve MLP-HGSO performance.

Since robust optimization algorithms and PCA were used to improve the performance of MLP models, the MLP-HGSO results are reliable without an EMD model. At the first station, the MLP-HGSO-EMD and MLP-HGSO models had training MAEs of 0.685 and 0.687 mm, respectively. At the testing level, the MLP-HGSO-EMD and MLP-HGSO models had MAE values of 0.710 and 0.712 Table 6.

Table 5 Stable probability matrix for stations

	Years	Observed Value		MLP-HGSO		RBFNN-HGSO		MLP-BA		RBFNN-BA	
Station 1	2001	0.77	0.23	0.77	0.23	0.75	0.25	0.76	0.24	0.78	0.22
	2002	0.65	0.35	0.65	0.35	0.64	0.36	0.63	0.37	0.65	0.35
	2003	0.55	0.45	0.55	0.45	0.57	0.43	0.58	0.42	0.59	0.41
	2004	0.52	0.48	0.52	0.48	0.54	0.46	0.59	0.40	0.60	0.40
	2005	0.57	0.43	0.56	0.44	0.59	0.41	0.60	0.40	0.61	0.39
	2006	0.56	0.44	0.57	0.43	0.57	0.43	0.57	0.43	0.57	0.43
	2007	0.59	0.41	0.58	0.42	0.60	0.40	0.60	0.40	0.60	0.40
	2008	0.61	0.39	0.60	0.40	0.64	0.36	0.68	0.32	0.68	0.32
	2009	0.64	0.36	0.62	0.38	0.65	0.35	0.67	0.33	0.67	0.33
	2010	0.64	0.36	0.64	0.36	0.67	0.33	0.69	0.31	0.69	0.31
	2011	0.65	0.35	0.65	0.35	0.69	0.31	0.69	0.31	0.69	0.31
	2012	0.72	0.28	0.72	0.28	0.71	0.29	0.71	0.29	0.71	0.29
	2013	0.65	0.35	0.65	0.35	0.65	0.35	0.65	0.35	0.65	0.35
	2014	0.62	0.38	0.62	0.38	0.62	0.38	0.62	0.38	0.62	0.38
Station 2	2001	0.65	0.35	0.65	0.35	0.64	0.36	0.63	0.37	0.65	0.35
	2002	0.66	0.34	0.66	0.34	0.65	0.35	0.65	0.35	0.65	0.35
	2003	0.57	0.57	0.57	0.43	0.57	0.43	0.58	0.42	0.58	0.42
	2004	0.64	0.36	0.62	0.38	0.65	0.35	0.67	0.33	0.67	0.33
	2005	0.66	0.34	0.66	0.34	0.65	0.35	0.60	0.40	0.61	0.39
	2006	0.56	0.44	0.57	0.43	0.57	0.43	0.57	0.43	0.57	0.43
	2007	0.72	0.28	0.72	0.28	0.71	0.29	0.71	0.29	0.71	0.29
	2008	0.61	0.39	0.60	0.40	0.64	0.36	0.68	0.32	0.68	0.32
	2009	0.72	0.28	0.72	0.28	0.71	0.29	0.71	0.29	0.71	0.29
	2010	0.62	0.38	0.62	0.38	0.62	0.38	0.62	0.38	0.62	0.38
	2011	0.65	0.35	0.65	0.35	0.65	0.35	0.65	0.35	0.65	0.35
	2012	0.62	0.38	0.62	0.38	0.62	0.38	0.62	0.38	0.62	0.38
	2013	0.65	0.35	0.65	0.35	0.65	0.35	0.65	0.35	0.65	0.35
	2014	0.64	0.36	0.64	0.36	0.67	0.33	0.69	0.31	0.69	0.31

Table 6 Comparison of MLP-HGSO-EMD with MLP-HGSO

Model	MAE	NSE	PBIAS
Training (Station 1)			
MLP-HGSO	0.687	0.95	0.17
MLP-HGSO-EMD	0.685	0.96	0.14
Testing			
MLP-HGSO	0.712	0.90	0.23
MLP-HGSO-EMD	0.710	0.92	0.18
Training (Station 1)			
MLP-HGSO	0.683	0.97	0.15
MLP-HGSO-EMD	0.681	0.98	0.12
Testing			
MLP-HGSO	0.711	0.92	0.22
MLP-HGSO-EMD	0.709	0.94	0.20

This research found that optimization algorithms play a crucial role in modeling. They can improve the output prediction accuracy of machine learning models. To select the optimal input scenarios, developers should employ methods, such as PCA. However, there are a variety of ways to define optimal features. The gamma test is a reliable method for selecting inputs. Since the optimization algorithms use different operators, the MLP and RBFNN models have different accuracy. Moreover, these models are highly adaptable.

The modelers can integrate these models with climate scenarios to predict targets under climate change conditions. The models of the current paper can be used for predicting spatial and temporal variation of rainfall in different regions. Figure 7 shows the boxplot models at the testing level. At the Sara Station, the median of the observed data, MLP-HGSO, RBFNN-HGSO,

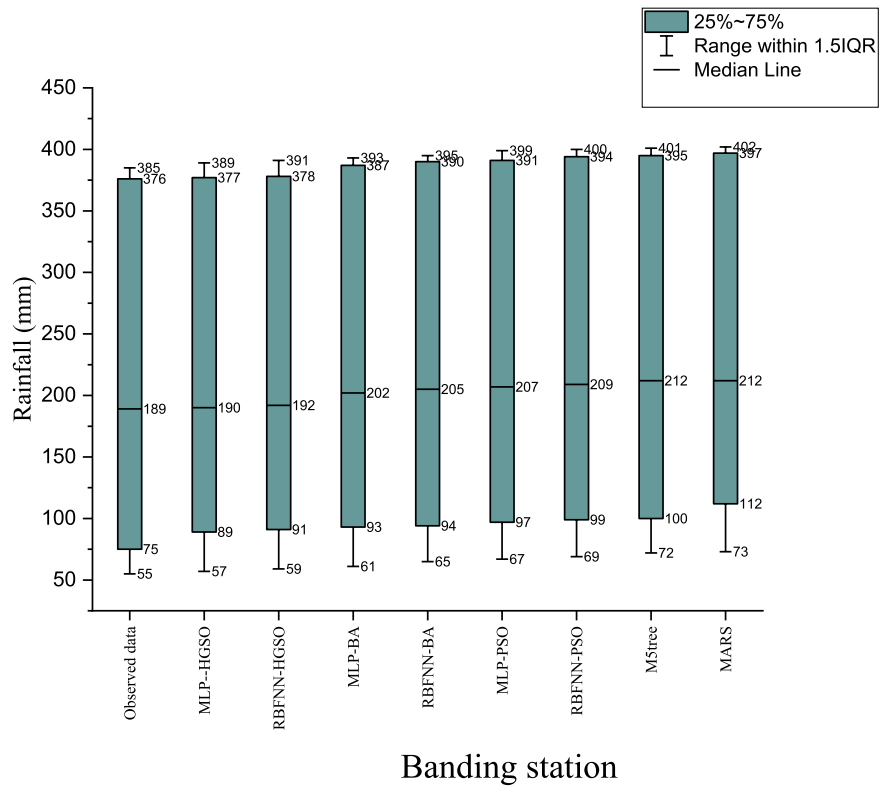
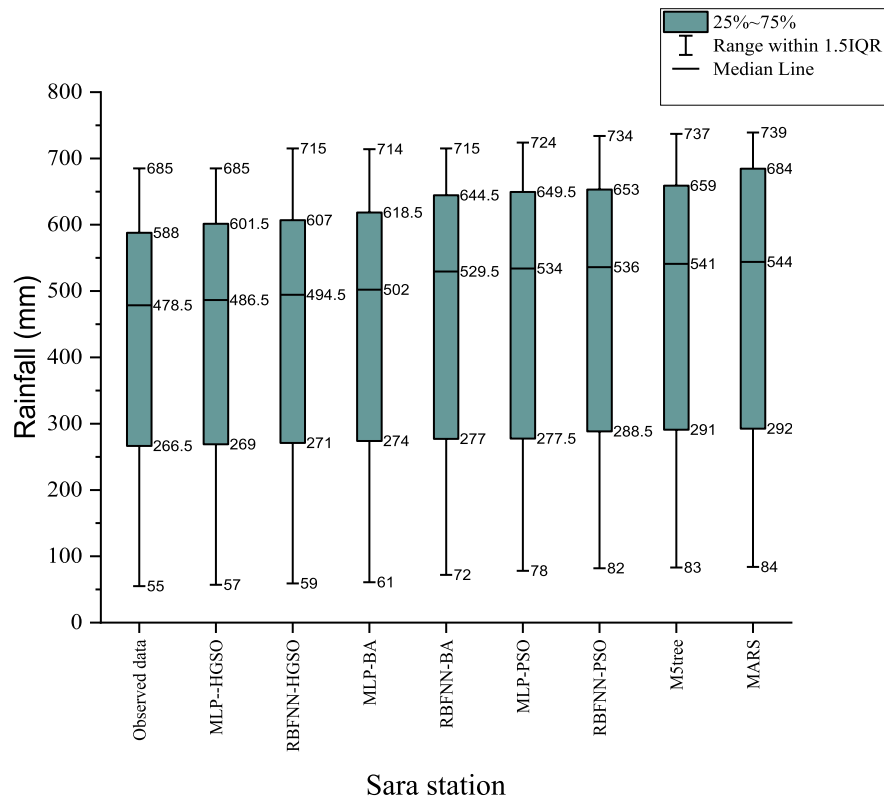


Fig. 7 Boxplots of models for comparing models

MLP–BA, RBFNN–BA, MLP–PSO, and RBFNN–PSO, M5Tree and multivariate adaptive regression splines (MARS) was 478 mm, 486.5 mm, 494.5 mm, 502 mm, 529.4 mm, 534 mm, 536 mm, 541 mm, and 544 mm, respectively. At the Banding Station, the median of the observed data, MLP–HGSO, RBFNN–HGSO, MLP–BA, RBFNN–BA, MLP–PSO, and RBFNN–PSO, M5Tree and multivariate adaptive regression splines (MARS) was 189 mm, 190 mm, 192 mm, 202 mm, 205.4 mm, 207 mm, 209 mm, 212 mm, and 221 mm, respectively.

Acharya et al. [44] used multi-model ensemble using ELM for monsoon rainfall forecasting over south peninsular India and the correlation coefficient (R) of simple arithmetic mean (EM), singular value decomposition-based multiple linear regression (SVD–MLR) and ELM in the test stage were found to be -0.34 , 0.47 and 0.63 , respectively. Beheshti et al. [45] used three different algorithms, centripetal accelerated PSO (CAPSO), a gravitational search algorithm and an imperialist competitive algorithm in training ANN for forecasting monthly rainfall in Johor State, Malaysia. Among the available methods, the MLP–CAPSO provided the best accuracy in the test stage with R value of 0.9103 . Helali et al. [46] forecasted rainfall using generalized regression ANN (GRNN), ANN, least square SVM and multi-linear regression (MLR) for 1–6-month lead times. The best GRNN, MLP, LSSVM, and MLR models gave the R values of 0.65 , 0.74 , 0.48 , and 0.74 in the test stage, respectively. In the presented study, the best model MLP–HGSO accurately forecasted the monthly rainfalls with an efficiency of 0.90 and 0.92 for the first and second stations, respectively.

Concerning water resources management, the outcomes offer valuable insights for policymakers aiming to ensure fair water resource distribution, especially in densely populated, water-scarce regions in Malaysia. With the use of rainfall forecasts, water managers can make well-informed decisions regarding water resource management. This may include adjusting water releases from reservoirs to enhance storage capacity based on anticipated inflow or implementing water conservation measures to safeguard water supplies during drought conditions [47]. In addition, rainfalls forecasts can aid in establishing early warning systems for floods and droughts, guiding water management strategies to alleviate the impact of weather-related disasters [46].

Conclusion

Rainfall is critical to irrigation and hydrological projects. This study explores the potential of hybrid soft computing models (MLP–PSO, MLP–BA, MLP–HGSO,

RBFNN–BA, RBENN–PSO, and RBFNN–HSO). Monthly rainfall data from two Malaysian stations were used as model inputs. In terms of efficiency, the hybrid MLP–HGSO model outperformed the MLP–BA, MLP–PSO, RBFNN–BA, RBFNN–PSO, and RBFNN–HGSO models. The RBFNN–HGSO outputs for the testing level reflected a value of $MAE=0.812$ mm and $PBIAS=0.27$, which was also more accurate than the MLP–HGSO developed with the same inputs.

The MLP–HGSO predicted and observed rainfall values revealed that the highest rainfall was observed in 2012, with a probability of 0.72 . Given the newly implemented hybrid MLP–HGSO model's strong performance, further research could include using the hybrid MLP–HGSO model to forecast other hydrological variables in the short and long run.

The current paper's challenges and limitations are data collection, adjusting computer systems and models, and determining the best input scenarios. It is recommended that the current models be tested in different climate zones and compared with other soft computing models could be adopted to find more precision results in rainfall modeling. In addition, another limitation of the present study is that conducted using rainfall data of two stations, it can be further augmented to use data of more stations (greater number of observations) to obtain better prediction results. In addition, the models are currently being researched for use as early warning systems in various regions. Some of the ML models are heavy and hence require huge computational resources and large computing time, and therefore, the proposed method can be compared with other ML methods which has simpler structure so as to see if the solution of the investigated problem is possible with simpler ML method. On the other hand, with the advanced technology (parallel computing), it is possible to use more complex ML models in real-time applications.

Acknowledgements

The authors are grateful to Malaysia's Department of Irrigation and Drainage for providing the necessary data.

Author contributions

All authors (SSS, OK, ME, AE, NA, MAG, SAB, ANA, SS) contributed to the manuscript. All authors read and approved the final manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Availability of data and materials

Not applicable.

Code availability

Not applicable.

Declarations

Competing interests

The authors declare that they have no conflicts of interest to report regarding the present study. Ozgur Kisi is one of the guest editors of the special issue

'Advanced computational methodologies for environmental modeling and sustainable water management' of *Environmental Sciences Europe*.

Received: 1 September 2023 Accepted: 23 November 2023

Published online: 14 December 2023

References

- Dar MUD, Shah AI, Bhat SA, Kumar R, Huisings D, Kaur R (2021) Blue Green infrastructure as a tool for sustainable urban development. *J Clean Product* 318:128474. <https://doi.org/10.1016/j.jclepro.2021.128474>
- Elbeltagi A, Nagy A, Mohammed S, Pande CB, Kumar M, Bhat SA, Zsembeli J, Huzsvai L, Tamás J, Kovács E, Harsányi E (2022) Combination of limited meteorological data for predicting reference crop evapotranspiration using artificial neural network method. *Agronomy* 12(2):516–537
- Elbeltagi A, Azad N, Arshad A, Mohammed S, Mokhtar A, Pande C, Etedali HR, Bhat SA, Islam ARMT, Deng J (2021) Applications of Gaussian process regression for predicting blue water footprint: case study in Ad Daqahliyah. *Egypt Agric Water Manag* 255:107052
- Shah AI, Dar MUD, Bhat RA, Singh JP, Singh K, Bhat SA (2020) Prospectives and challenges of wastewater treatment technologies to combat contaminants of emerging concerns. *Ecol Eng* 152:105882
- Yahya A, Saeed A, Ahmed AN, Binti Othman F, Ibrahim RK, Afan HA, Elshafie A (2019) Water quality prediction model based support vector machine model for ungauged river catchment under dual scenarios. *Water* 11(6):1231
- Danladi A, Stephen M, Aliyu B, Gaya G, Silikwa N, Machael Y (2018) Assessing the influence of weather parameters on rainfall to forecast river discharge based on short-term. *Alexandria Eng J* 57:1157–1162
- Silva VD, Maciel GF, Braga CC, Silva JL, Souza EP, Almeida RS, Silva MT, Holanda RM (2017) Calibration and validation of the AquaCrop model for the soybean crop grown under different levels of irrigation in the Motopiba region. *Brazil Ciência Rural*. <https://doi.org/10.1590/0103-8478r20161118>
- Yousif AA, Sulaiman SO, Diop L, Ehteram M, Shahid S, Al-Ansari N, Yaseen ZM (2019) Open channel sluice gate scouring parameters prediction: different scenarios of dimensional and non-dimensional input parameters. *Water* 11(2):353
- Sihag P, Dursun OF, Sammen SS, Malik A, Chauhan A (2021) Prediction of aeration efficiency of parshall and modified venturi flumes: application of soft computing versus regression models. *Water Supply* 21(8):4068–4085. <https://doi.org/10.2166/ws.2021.161>
- Sihag P, Kumar M, Sammen SS (2021) Predicting the infiltration characteristics for semi-arid regions using regression trees. *Water Supply* 21(6):2583–2595. <https://doi.org/10.2166/ws.2021.047>
- Ebtehaj I, Sammen SS, Sidek LM, Malik A, Sihag P, Al-Janabi AMS, Chau K-W, Bonakdari H (2021) Prediction of daily water level using new hybridized GS-GMDH and ANFIS FCM models. *Eng Appl Comput Fluid Mechan* 15(1):1343–1361. <https://doi.org/10.1080/19942060.2021.1966837>
- Sammen SS, Ehteram M, Abba SI et al (2021) A new soft computing model for daily streamflow forecasting. *Stoch Environ Res Risk Assess* 35:2479–2491. <https://doi.org/10.1007/s00477-021-02012-1>
- Abba S, Abdulkadir R, Gaya M, Sammen SS, Ghali U, Nawaila M, Oğuz G, Malik A, Al-Ansari (2021) Effluents quality prediction by using nonlinear dynamic block-oriented models: A system identification approach. *Desalin Water Treat* 218:52–62
- Pham QB, Mohammadpour R, Linh NTT et al (2021) Application of soft computing to predict water quality in wetland. *Environ Sci Pollut Res* 28:185–200. <https://doi.org/10.1007/s11356-020-10344-8>
- Kumar A, Sridevi C, Durai VR, Singh KK, Mukhopadhyay P, Chattopadhyay N (2019) MOS guidance using a neural network for the rainfall forecast over India. *J Earth Syst Sci* 128(5):130
- Yaseen ZM, Ehteram M, Hossain MS, Fai CM, Binti Koting S, Mohd NS, Binti Jaafar WZ, Afan HA, Hin LS, Zaini N, Ahmed AN (2019) A novel hybrid evolutionary data-intelligence algorithm for irrigation and power production management: Application to multipurpose reservoir systems. *Sustainability* 11(7):1953
- Mandal T, Jothiprakash V (2012) Short-term rainfall prediction using ANN and MT techniques. *ISH J Hydraulic Eng* 18(1):20–26
- Hipni A, El-shafie A, Najah A, Karim OA, Hussain A, Mukhlisin M (2013) Daily forecasting of dam water levels: comparing a support vector machine (SVM) model with adaptive neuro fuzzy inference system (ANFIS). *Water Resour Manage* 27(10):3803–3823. <https://doi.org/10.1007/s11269-013-0382-4>
- Nourani V, Komasi M (2013) A geomorphology-based ANFIS model for multi-station modeling of rainfall–runoff process. *J Hydrol* 490:41–55
- Akrami SA, Nourani V, Hakim SJS (2014) Development of nonlinear model based on wavelet-ANFIS for rainfall forecasting at Klang Gates Dam. *Water Resour Manage* 28(10):2999–3018
- Mehdizadeh S, Behmanesh J, Khalili K (2018) New approaches for estimation of monthly rainfall based on GEP-ARCH and ANN-ARCH hybrid models. *Water Resour Manage* 32(2):527–545
- Mehr AD, Nourani V (2018) Season algorithm-multigene genetic programming: a new approach for rainfall-runoff modelling. *Water Resour Manage* 32(8):2665–2679
- Ouyang Q, Lu W (2018) Monthly rainfall forecasting using echo state networks coupled with data preprocessing methods. *Water Resour Manage* 32(2):659–674
- Feng Q, Wen X, Li J (2015) Wavelet analysis-support vector machine coupled models for monthly rainfall forecasting in arid regions. *Water Resour Manage* 29(4):1049–1065. 540
- Yaseen ZM, Jaafar O, Deo RC, Kisi O, Adamowski J, Quilty J, El-Shafie A (2016) Stream-flow forecasting using extreme learning machines: a case study in a semi-arid region in Iraq. *J Hydrol* 542:603–614
- Mehr AD, Nourani V, Khosrowshahi VK, Ghorbani MA (2019) A hybrid support vector regression–firefly model for monthly rainfall forecasting. *Int J Environ Sci Technol* 16(1):335–346
- Pham BT, Le LM, Le TT, Bui KTT, Le VM, Ly HB, Prakash I (2020) Development of advanced artificial intelligence models for daily rainfall prediction. *Atmos Res* 237:104845
- Basha CZ, Bhavana N, Bhavya P, Sowmya V (2020) Rainfall prediction using machine learning & deep learning techniques. In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 92–97). IEEE.
- Ruma JF, Adnan MSG, Dewan A, Rahman RM (2023) Particle swarm optimization based LSTM networks for water level forecasting: A case study on Bangladesh river network. *Result Eng* 17:100951. <https://doi.org/10.1016/j.rineng.2023.100951>
- Wang H, Wang W, Du Y, Xu D (2021) Examining the applicability of wavelet packet decomposition on different forecasting models in annual rainfall prediction. *Water* 13(15):1997
- Kumar R, Singh MP, Roy B, Shahid AH (2021) A comparative assessment of metaheuristic optimized extreme learning machine and deep neural network in multi-step-ahead long-term rainfall prediction for all-Indian regions. *Water Resour Manage* 35(6):1927–1960
- Li H, He Y, Yang H, Wei Y, Li S, Xu J (2021) Rainfall prediction using optimally pruned extreme learning machines. *Nat Hazards* 108(1):799–817
- Riahi-Madvar H, Gharabaghi B (2022) Pre-processing and Input Vector Selection Techniques in Computational Soft Computing Models of Water Engineering. In: Bozorg-Haddad, O., Zolghadr-Asli, B. (eds) *Computational Intelligence for Water and Environmental Sciences. Studies in Computational Intelligence, Vol 1043*. Springer, Singapore. https://doi.org/10.1007/978-981-19-2519-1_20
- Wang X, Liu Z, Zhou W, Jia Z, You Q (2019) A forecast-based operation (FBO) mode for reservoir flood control using forecast cumulative net rainfall. *Water Resour Manage* 33(7):2417–2437
- Hammid AT, Sulaiman MH, Abdalla AN (2018) Prediction of small hydropower plant power production in Himreen Lake dam (HLD) using artificial neural network. *Alexandria Eng. J.* 57(1):211–221
- Graham A, Sahu JK, Sahu YK, Yadu A (2019) Forecast future rainfall & temperature for the study area using seasonal auto-regressive integrated moving averages (SARIMA) model. *IJCS* 7(1):894–897
- Yahya BM, Seker DZ (2019) Designing weather forecasting model using computational intelligence tools. *Appl Artif Intell* 33(2):137–151
- Hashim R, Roy C, Motamedi S, Shamshirband S, Petković D, Gocic M, Lee SC (2016) Selection of meteorological parameters affecting rainfall estimation using neuro-fuzzy computing methodology. *Atmos Res* 171:21–30
- Ehteram M, Sammen SS, Panahi F et al (2021) A hybrid novel SVM model for predicting CO2 emissions using Multiobjective Seagull Optimization.

Environ Sci Pollut Res 28:66171–66192. <https://doi.org/10.1007/s11356-021-15223-4>

40. Yang X-S (2010) A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, pp 65–74
41. Almohammed F, Sihag P, Sammen SS, Ostrowski KA, Singh K, Prasad CVSR, Zajdel P (2022) Assessment of soft computing techniques for the prediction of compressive strength of bacterial concrete. *Materials* 15:489. <https://doi.org/10.3390/ma15020489>
42. Mokhtar A, El-Ssawy W, He H, Al-Anasari N, Sammen SS, Gyasi-Agyei Y, Abuarab M (2022) Using machine learning models to predict hydroponically grown lettuce yield. *Front Plant Sci* 13:706042. <https://doi.org/10.3389/fpls.2022.706042>
43. Malekian A, Choubin B, Liu J, Sajedi-Hosseini F (2019) Development of a new integrated framework for improved rainfall-runoff modeling under climate variability and human activities. *Water Resour Manage* 33(7):2501–2515
44. Acharya N, Shrivastava NA, Panigrahi BK et al (2014) Development of an artificial neural network based multi-model ensemble to estimate the northeast monsoon rainfall over south peninsular India: an application of extreme learning machine. *Clim Dyn* 43:1303–1310. <https://doi.org/10.1007/s00382-013-1942-2>
45. Beheshti Z, Firouzi M, Shamsuddin SM et al (2016) A new rainfall forecasting model using the CAPSO algorithm and an artificial neural network. *Neural Comput Applic* 27:2551–2565. <https://doi.org/10.1007/s00521-015-2024-7>
46. Helali J, Nouri M, Mohammadi Ghaleni M et al (2023) Forecasting precipitation based on teleconnections using machine learning approaches across different precipitation regimes. *Environ Earth Sci* 82:495. <https://doi.org/10.1007/s12665-023-11191-9>
47. Pattanaik DR, Das AK (2015) Prospect of application of extended range forecast in water resource management: a case study over the Mahanadi River basin. *Nat Hazards* 77:575–595. <https://doi.org/10.1007/s11069-015-1610-4>
48. Nhita F, Annisa S, Kinasih S (2015) Comparative study of grammatical evolution and adaptive neuro-fuzzy inference system on rainfall forecasting in Bandung. In 2015 3rd International Conference on Information and Communication Technology (ICoICT) (pp. 6–10). IEEE.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
