

RESEARCH

Open Access



# Gaussian mutation–orca predation algorithm–deep residual shrinkage network (DRSN)–temporal convolutional network (TCN)–random forest model: an advanced machine learning model for predicting monthly rainfall and filtering irrelevant data

Mohammad Ehteram<sup>1\*</sup>, Mahdie Afshari Nia<sup>2</sup>, Fatemeh Panahi<sup>2</sup> and Hanieh Shabanian<sup>3</sup>

## Abstract

Monitoring water resources requires accurate predictions of rainfall data. Our study introduces a novel deep learning model named the deep residual shrinkage network (DRSN)—temporal convolutional network (TCN) to remove redundant features and extract temporal features from rainfall data. The TCN model extracts temporal features, and the DRSN enhances the quality of the extracted features. Then, the DRSN–TCN is coupled with a random forest (RF) model to model rainfall data. Since the RF model may be unable to classify and predict complex patterns and data, our study develops the RF model to model outputs with high accuracy. Since the DRSN–TCN model uses advanced operators to extract temporal features and remove irrelevant features, it can improve the performance of the RF model for predicting rainfall. We use a new optimizer named the Gaussian mutation (GM)–orca predation algorithm (OPA) to set the DRSN–TCN–RF (DTR) parameters and determine the best input scenario. This paper introduces a new machine learning model for rainfall prediction, improves the accuracy of the original TCN, and develops a new optimization method for input selection. The models used the lagged rainfall data to predict monthly data. GM–OPA improved the accuracy of the orca predation algorithm (OPA) for feature selection. The GM–OPA reduced the root mean square error (RMSE) values of OPA and particle swarm optimization (PSO) by 1.4%–3.4% and 6.14–9.54%, respectively. The GM–OPA can simplify the modeling process because it can determine the most important input parameters. Moreover, the GM–OPA can automatically determine the optimal input scenario. The DTR reduced the testing mean absolute error values of the TCN–RAF, DRSN–TCN, TCN, and RAF models by 5.3%, 21%, 40%, and 46%, respectively. Our study indicates that the proposed model is a reliable model for rainfall prediction.

**Keywords** Deep learning models, Hybrid models, Feature selection, Optimizers

\*Correspondence:

Mohammad Ehteram  
mohammdehteram@semnan.ac.ir

Full list of author information is available at the end of the article



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Introduction

Food production and crop growth depend on rainfall [30]. Excessive rainfall can lead to floods, landslides, and other natural disasters. Accurate rainfall prediction is essential for monitoring water resource sources [12, 34]. Urban planners use rainfall predictions to design drainage systems [14, 15]. Rainfall predictions can be used to study the effects of climate change on different regions of the world [10, 44]. Predicting and mitigating floods requires timely and accurate rainfall predictions. We need accurate rainfall predictions to fully understand climate patterns [41]. To assess the impact of changing weather patterns on ecosystems and communities, scientists use rainfall prediction data.

Rainfall prediction is complex because numerous factors influence rainfall patterns [5, 6, 8]. Local factors such as urbanization and land use changes can influence microclimates and rainfall patterns [1]. In order to accurately predict rainfall, these factors must be considered.

Machine learning (ML) model is a valuable tool for predicting rainfall. Machine learning models can update their predictions as new data becomes available [1, 16]. Machine learning models can process large and diverse data sets, including historical weather data, radar data, and various atmospheric parameters [3, 24]. A random forest (RF) model is a simple and accurate machine learning model [2, 28]. The RF has a simple structure and high accuracy. Scholars have used the RAF model to classify and predict data.

Yu et al. [39] developed the support vector machine (SVM) and RF models to predict rainfall. The accuracy of the RF model decreased as the time horizon increased. Ali et al. [2] combined complete ensemble empirical mode decomposition (CEEMD) with the RF model to predict monthly rainfall. Time series data were decomposed into intrinsic mode functions (IMFs). The study results indicated that the CEEMD–RF successfully predicted rainfall values. Shijun et al. [33] compared the RF and the support vector machine model (SVM) for predicting rainfall. The study concluded that the RF model was more accurate than the SVM model. Singh et al. [32] developed the RAF and SVM model to predict runoff. They reported that the RF was successful in predicting runoff. Qiao et al. [29] coupled the RF model with the temporal convolutional network (TCN) to identify the most important predictors for rainfall prediction. The study results indicated that the RAF–TCN provided high accuracy for rainfall predictions. Vergni and Todisco [35] developed the RF model to predict soil loss. They used the runoff coefficient and the period of occurrence to predict soil loss. The study results showed that the RF model could successfully estimate soil loss.

Although the RAF model successfully predicts rainfall time series, it has some limitations. The RAF model lacks advanced operators to capture temporal dependencies [44]. Time series data may have irregular intervals or missing values, but the RF model may not process them. This paper addresses the shortcomings of the RF and predicts monthly rainfall data. The main innovations of the study are the development of a new model for spatial and temporal rainfall predictions and the creation of a new hybrid optimization algorithm for choosing inputs. The new optimization algorithm can reduce the complexity of the feature selection process because it selects the most important features.

Recent studies have indicated that deep learning models are reliable tools for extracting important information from data and improving the performance of classical machine learning models such as random forest models and regression models [1, 18]. Temporal convolutional network (TCN) is a deep learning model that processes sequential data. TCNs use convolutional layers to extract features from sequential data [18, 19]. These convolutional layers can capture local patterns from input sequences [37, 38]. A dilated convolution can capture long-term dependencies.

As deep learning models use advanced operators and different computational layers for processing data, they can enhance the performance of classical machine learning models [1, 37, 38]. A TCN model includes multiple layers that extract features from data and process a large number of data. These features make TCN ideal for coupling with classical machine learning models and improving their performance. Our study uses the capabilities of the TCN model to enhance the performance of the RF model. The TCN–RF model predicts data and extracts features. The TCN model can be used as a feature extractor to improve the performance of the RAF model for predicting time series. The TCN model utilizes one-dimensional convolutional layers to process sequential data [43]. Rainfall patterns often exhibit non-linear dependencies over time. Linear models may not be able to capture these complex relationships. The TCN model can effectively capture the temporal dependencies of the rainfall time series. TCN can act as a feature extractor. It can automatically learn relevant features from the rainfall time-series data.

In this study, we develop and introduce a novel TCN model that is combined with the RF model and predicts rainfall data. The original TCN model may not be able to identify and remove unnecessary or redundant features from time-series data. The irrelevant and redundant features can influence the accuracy of the TCN model. Irrelevant features may contain misleading information. If the TCN model uses unnecessary and redundant features, it

may learn incorrect patterns. Our study addresses this drawback of the TCN model.

Our study introduces a novel technique called a deep residual shrinkage network (DRSN) to address the limitations of the TCN model. The DRSN model is inserted into the residual connections of the TCN model. Since the DRSN model removes redundant and irrelevant features, it enhances the quality of the extracted features [11]. The DRSN uses a soft thresholding operator to remove irrelevant features. [4]. The TCN model can improve its prediction accuracy as it focuses on the key features and reduces irrelevant features. The soft thresholding operator deletes features whose absolute values are below the threshold  $T$  and shrinks features whose absolute values are above the threshold  $T$ . Since selecting input features and predicting rainfall and non-linear patterns is complex and difficult, we need advanced tools to accurately predict rainfall patterns. This article introduces an effective mechanism for removing irrelevant data and improving the performance of machine learning models. The DRSN was introduced to address this issue. Also, the hybrid TCN–RF model can capture spatiotemporal patterns and make accurate rainfall predictions. Thees a new robust optimizer for feature selection and reduce the complexity of the problem.

The current paper contains the following innovations:

- The paper introduces the TCN–RF model to enhance the efficiency of the RF model for monthly rainfall prediction. The TCN extracts features and sends them to the RF model for rainfall prediction. No study has developed the TCN–RF model to predict rainfall data.
- Our study develops the TCN model for identifying important features. A DRSN–TCN–RF (DTR) predicts rainfall data and removes redundant and irrelevant features. The DRSN–TCN is a novel deep learning model for processing data.
- The choice of input selection plays a key role in rainfall prediction. Binary optimizers are robust tools for choosing inputs. Moreover, we need a robust optimizer to set the model parameters. Scholars have developed different optimizers to solve optimization problems in recent years. Jiang et al. [13] introduced a novel optimization algorithm named the orca predation algorithm (OPA) to solve complex problems. It was inspired by the hunting behavior of orcas. Compared to other algorithms, the OPA produced better results. The OPA may get stuck in local optima when an optimization problem involves a large number of decision variables and constraints. Our study develops the OPA to address this issue. Jiang et al. [13] introduced the continuous version of the OPA to

solve complex problems. Since we need binary versions of optimization algorithms for feature selection, we convert the continuous version of the OPA into a binary version of the OPA. The process of feature selection involves binary decision variables. We use a new binary algorithm to identify the best input scenarios.

Details of the models are presented in the following section. In addition, the fourth and fifth sections will discuss the results. Finally, the sixth section concludes and presents suggestions for the next paper.

## Materials and methods

### Random forest model

A random forest model consists of several decision trees. The prediction process begins with collecting input and output data [22]. The dataset is cleaned to handle missing values, outliers, and inconsistencies [20]. Multiple decision trees are trained on different subsets of the training data. Decision trees are created recursively. The algorithm selects the best features and divides the nodes of each decision tree based on a criterion such as Gini impurity (for classification) or mean squared error (for prediction) [42]. The root node of a decision tree is the first node. The root node is split into child nodes based on particular features and splitting criteria [42]. The model divides nodes until a stopping criterion is met. The stop creation can be determined by a maximum depth, a minimum number of samples, or other factors. After we create all the individual decision trees, we have an ensemble structure. The ensemble structure combines individual predictions to make a final prediction for each data point.

### TCN model

TCNs can capture both short- and long-term dependencies. TCNs use residual connections to propagate gradients and improve model convergence. The TCN model consists of convolutional layers [43]. The convolutional layer processes sequential data. Each convolutional layer has a set of filters that slide (convolve) over the input sequence to extract local patterns and features [37, 38]. Causal convolution is the next component of TCNs that capture temporal dependencies of sequential data [18, 19]. Causal convolutions prevent information leakage from future time steps. Causal convolutions use dilated convolutions to capture temporal dependencies and efficiently cover distant parts of input sequences. A convolutional filter can capture both short-term and long-term dependencies when the dilation rate is higher than 1. The receptive field of a convolutional filter refers to the region of the input sequence that affects the output of that

particular filter. A residual connection is the next component of the TCN model.

Residual or skip connections are commonly employed to tackle the vanishing gradient problem and train deep TCN architectures [18, 19]. The TCN models can use max-pooling or average-pooling layers to down sample the outputs. The pooling layer can reduce the spatial dimensionality of feature maps [17]. TCNs typically have an output layer that produces predictions after the input sequence is processed by convolutional layers.

### Hybrid DRSN–TCN model

The DRSN–TCN (DRT) is a deep learning model designed for predicting time-series data. DRSN inserts a soft thresholding mechanism into the residual connections of the traditional TCN model. A soft thresholding technique reduces irrelevant information. This method eliminates or shrinks redundant features. We drop features whose absolute values are below a threshold and shrink features whose absolute values are above the threshold. DRSN uses subnetworks for threshold estimation. These subnetworks receive the absolute value of each element of the output matrix and apply global average pooling to obtain mean values for each channel. Based on a sigmoid activation function, threshold values are scaled between 0 and 1. The hybrid model consists of multiple components. An inflated causal convolution layer is the first component of the hybrid model. Causal convolutions ensure that outputs depend only on past and present inputs. Dilated convolutions increase the receptive field gradually. At the next level, a threshold estimation building unit is employed to estimate thresholds for feature shrinkage. Feature shrinkage reduces the dimension of the feature space and preserves relevant information. The threshold estimation building unit computes the absolute value of each element of the output matrix of the causal convolution layer [11]. The DTR uses this strategy to ensure that each threshold is positive [4]. Once we get the absolute values, we apply one-dimensional global average pooling to the output matrix. Global average pooling computes the mean value of the features to downsize the feature map. The one-dimensional vector of the mean values of each feature map is used as input for later thresholding. Each feature map will be shrunk using these thresholds. We set feature values below the threshold to zero to reduce redundant features. Soft thresholding is computed as follows:

$$z = \begin{bmatrix} x - \varphi \leftarrow x > \varphi \\ 0 \leftarrow -\varphi \leq x \leq \varphi \\ x + \varphi \leftarrow x < -\varphi \end{bmatrix}, \tag{1}$$

where  $x$  is the input feature,  $z$  is the output, and  $\varphi$  is the threshold value.

### Optimization algorithms for feature selection

#### Orca optimization algorithm (OPA)

Orcas are intelligent and social mammals. They are the largest members of the dolphin family [13]. Typically, an orca group consists of three members: calves (20%), adult males (20%), and females (60%). A male may leave a social group if it becomes too large. Orcas emit sounds such as clicks, whistles, and pulsed calls, into the water to precisely determine the location and distance of potential prey [13]. To locate prey and communicate with each other, orcas use sonar. Orcas confine their prey to the surface of the water. Orcas employ a variety of tactics to stun or incapacitate fish. They hit the fish with their powerful tails. First, we initialize a group of orcas as follows:

$$OR = (or_1, \dots, or_N), \tag{2}$$

where  $OR$ : a group of orcas,  $or_1$ : The location of the first orca,  $or_N$ : The location of the Nth orca. Orcas use echolocation to locate their prey and communicate with group members. They emit different types of vocalizations, including clicks, whistles, and pulsed calls, to transfer information to each other [13]. Orcas use vocalizations to coordinate their actions during hunting and to maintain social bonds. The orcas employ a variety of strategies, including echolocation (sonar), coordinated swimming, and communication, to drive the school of fish to the surface of the water. Once the school of fish approaches the surface, the orcas create a controlled enclosure around the fish [13]. A group of orcas surrounds the fish. Orcas employ strategies such as driving and encircling prey to catch their prey.

*Driving of prey* The small groups of orcas locate and hunt the school of fish efficiently (First scenario). Large groups of orcas may reach the desired position (Second scenario). When the orca group is large, the spatial dimension of swimming is high. Orcas change their location based on the size of their population [13]

$$VE_{chase,1,i} = \mu \times (\beta \times OR_{best}^t - F \times (\varphi \times J^t + b \times OR_i^t)), \tag{3}$$

$$VE_{chase,2,i} = e \times OR_{best}^t - OR_i^t, \tag{4}$$

$$J = \frac{\sum_{i=1}^N OR_i^t}{N}, \tag{5}$$

$$b = 1 - \beta, \tag{6}$$



$$\left[ \begin{array}{l} OR_{chase,1,i} = OR_i^t + VE_{chase,1,i}^t \leftarrow \text{if}(\text{rand}) > u \\ OR_{chase,1,i} = OR_i^t + VE_{chase,1,i}^t \leftarrow \text{if}(\text{rand}) \leq u \end{array} \right], \quad (7)$$

where  $OR_{chase,1,i}$  is the location of orca based on the first scenario,  $OR_{chase,1,i}$  is the location of orca based on the second scenario,  $u$  is the random number,  $N$  is the number of orcas,  $VE_{chase,1,i}$  is the velocity of orca based on the first scenario,  $\beta$  is the random number,  $VE_{chase,2,i}$  is the velocity of orca based on the second scenario,  $F$  is the controller parameter ( $F=2$ ),  $\mu$  is the random parameter,  $e$  is the random number,  $OR_{best}^t$  is the location of best orca, and  $J$  is the average location of one group.

**Encircling of prey** Fish schools cannot move in different directions because orcas surround them. Orcas often use vocalizations to coordinate their movements and determine their next position. They change their location based on the locations of three randomly selected orcas [13].

$$RO_{chase,3,i,k}^t = RO_{j1,k}^t + \chi \times (RO_{j2,k}^t - RO_{j3,k}^t), \quad (8)$$

$$\chi = 2 \times \left( \text{rand} - \frac{1}{2} \right) \times \frac{\max(it) - t}{\max(it)}, \quad (9)$$

where  $RO_{chase,3,i,k}^t$  is the new location of orcas,  $RO_{j1,k}^t$ ,  $RO_{j2,k}^t$ , and  $RO_{j3,k}^t$  is the random location of orcas,,  $\text{rand}$  is the random number,  $\max(it)$  is the maximum number of iterations, and  $t$  is the iteration number. The orcas will update their location if they achieve a better objective function value at a new location.

$$\left[ \begin{array}{l} OR_{chase,i}^t = OR_{chase,i}^t \text{ if } \rightarrow f(OR_{chase,i}^t) < f(OR_i^t) \\ OR_{chase,i}^t = OR_i^t \text{ if } \rightarrow f(OR_{chase,i}^t) \geq f(OR_i^t) \end{array} \right], \quad (10)$$

where  $f(OR_{chase,i}^t)$  is the objective function of the new location of the orca,  $f(OR_i^t)$  is the objective function value of the current location of the orca.

**Attacking preys** A group of orcas forms an enclosure or barrier to capture prey. After the orca enters the enclosure to attack prey, it whips or stuns the fish with its tail. The OPA hypothesizes that four orcas have the best locations in the enclosure [13]. The other orcas update their direction based on the locations of the four orcas that enter the enclosure. After eating their prey, orcas return to their original position and replace another orca [13]. At this level, orcas adjust their direction of movement based on the positions of nearby orcas, which are randomly placed in the search area. The final location and speed of the orcas are determined using the following equations:

$$VE_{attack,1,i} = (OR_{first}^t + OR_{second}^t + OR_{third}^t + OR_{fourth}^t), \quad (11)$$

$$VE_{attack,2,i} = (OR_{chase,j1}^t + OR_{chase,j2}^t + OR_{chase,j3}^t), \quad (12)$$

$$OR_{attack,i}^t = OR_{chase,i} + \alpha_1 \times VE_{attack,1,i} + \alpha_2 \times VE_{attack,2,i}, \quad (13)$$

where  $VE_{attack,1,i}$  is the speed of the  $i$ th orca;  $VE_{attack,2,i}$  is the speed of the  $i$ th orca at the enclosure;  $OR_{first}^t$ ,  $OR_{second}^t$ ,  $OR_{third}^t$ ,  $OR_{fourth}^t$  are the location of the first, second, third, fourth orca;  $OR_{chase,j1}^t$ ,  $OR_{chase,j2}^t$ , and  $OR_{chase,j3}^t$  are the random locations of orcas;  $OR_{attack,i}^t$  is the final location of orca;  $\alpha$  and  $\alpha_2$  are the random numbers. The optimization process begins with creating an initial population of orcas. Then, we use Eqs. 3–10 to update the position and speed of orcas. Finally, we update the location of the orcas based on the attacking prey.

### Improved OPA

Our study introduces a novel technique to tackle the problem of premature convergence and local optima of OPA. A feature selection problem may contain a large number of decision variables, so the OPA may get stuck in local optima. Premature convergence is also another disadvantage of the OPA. A novel approach is used to address the shortcomings of the OPA. To increase the diversity of solutions and avoid local optima, we used the Gaussian mutation method. The Gaussian mutation (GM) produces new solutions that follow the characteristics of the normal distribution [40]. The proposed solutions are placed near the original solutions [40]. Thus, the Gaussian mutation improves the exploitation ability of the OPA. As the Gaussian mutation produces new solutions, it can improve the ability of the OPA to avoid trapping in local optima. The GM also helps the OPA identify promising regions.

$$f_G = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(z - \mu)^2}{2\sigma^2}\right), \quad (14)$$

$$OR_G(t + 1) = G(OR_\mu, OR_{\sigma^2}), \quad (15)$$

$$OR_\mu = \frac{OR_{best}(t) + OR(t)}{2}, \quad (16)$$

$$OR_{\sigma^2} = |OR_{best} - OR_t|, \quad (17)$$

where  $G(OR_\mu, OR_{\sigma^2})$  is a random number,  $OR_\mu$  is the mean of the Gaussian mutation,  $OR_{\sigma^2}$  is the variance of the Gaussian mutation,  $x$  is the random variable,  $\sigma$  is the

variance of random variable,  $\mu$  is the mean of random variable,  $OR_{best}(t)$  is the best position of orca, and  $OR(t)$  is the current location of orca. The new algorithm is named GM-OPA. Our study compares new optimization algorithms with multiple optimization algorithms.

**Particle swarm optimization algorithm (PSO).**

PSO is one of the practical optimization algorithms for solving complex problems. The best particle guides the swarm particle toward the best location [26]. Particles are characterized by their velocity and position [27]. The particles change their components as follows:

$$ve_i(t + 1) = \varepsilon ve_i(t) + \tau_1 r_1 [p_i - s_i] + \tau_2 r_2 [p_g - s_i], \tag{18}$$

$$s_i(t + 1) = s_i(t) + ve_i(t + 1), \tag{19}$$

where  $ve_i(t + 1)$  is the  $i$ th velocity at iteration  $(t + 1)$ ,  $s_i(t)$  is the  $i$ th solution,  $r_1$  and  $r_2$  are the random numbers,  $p_i$  is the local best solution, and  $p_g$  is the global best solution,  $s_i(t)$  is the  $i$ th solution, and  $s_i(t + 1)$  is the new solution,  $\tau_1$  and  $\tau_2$  are the learning factors, and  $r_1$  and  $r_2$  are the random numbers, and  $\varepsilon$  is the weight coefficient.

**Crow optimization algorithm (COA)**

The COA was inspired by the life of crows [31]. Crows hide their food in a safe place. Each crow may pursue other crows to steal their food. First, we initialize the location of the crows as follows: [21]:

$$CR_j^{it} = [CR_1^i, CR_2^i, \dots, CR_d^i], \tag{20}$$

where  $CR_j^{it}$  is the  $i$ th location of crow in the  $j$ th dimension and  $d$  is the number of dimensions. The COA uses a memory to save the best solutions:

$$M_j^i = [m_1^i, m_2^i, \dots, m_d^i], \tag{21}$$

where  $M_j^i$  is the  $i$ th memory in the  $j$ th dimension. The crow search algorithm (COA) uses two scenarios to find optimal solutions. The first scenario assumes that the first crow knows that the second crow is following it. The first crow deceives the second crow. The first crow updates its location based on the following equation.

$$X_j^{t+1} = \begin{cases} X_j^t + r_i \times fl_j^t \times (M_i^t - X_j^t) & \leftarrow r_j \geq AP_j^t \\ \text{Select}(a)(\text{random})(\text{location}) & \end{cases} \tag{22}$$

where  $X_j^{t+1}$  is the  $j$ th location at  $t + 1$  iteration,  $M_i^t$  is the  $i$ th memory at iteration  $t$ ,  $AP_j^t$  is the awareness probability of the crow  $j$ ,  $r_i$  is the random number,  $Y_j^t$  is the current location of crow. The second scenario assumes that

the first crow does not know that the second crow is following it. The second crow update its locations based on the following equation [9]:

$$X_j^{t+1} = X_j^t + r_j \times fl_j^t \times (M_i^t - X_j^t), \tag{23}$$

where  $X_j^{t+1}$  is the new location of the  $j$ th crow. The COA updates the memory using Eq. 24:

$$M^{t+1} = \begin{cases} X_i^{t+1} & \leftarrow \text{fitness}(X_i^{t+1}) > X_i^t \\ M_i^t & \end{cases} \tag{24}$$

The first step is to define the initial positions of the crows. The objective function value is used to determine the best crow. Then, we use one of the scenarios to update the crow location. The memory is updated at each level. The process continues until the convergence criteria are met.

**Sine cosine optimization algorithm (SCOA)**

SCA uses cosine and sine functions to search for optimal solutions. First, we define the initial solutions for the optimization process [25]. At the next level, the objective function value is calculated to evaluate the quality of the solutions. Equations 25 and 26 are used to update the values of solutions [7].

$$SO_i^{t+1} = \begin{cases} SO_i^t + v_1 \times \sin(v_2) \times |v_3 SO_{ig}^t - SO_i^t|, & v_4 \leq 0.50 \\ SO_i^t + v_1 \times \cos(v_2) \times |v_3 SO_{ig}^t - SO_i^t|, & v_4 > 0.50 \end{cases} \tag{25}$$

where  $SO_i^{t+1}$  is the new solution,  $v_1, v_2, v_3$ , and  $v_4$  are the random values, and  $SO_{ig}^t$  is the best solution. The  $v_1$  is computed using the following equation:

$$v_1 = \kappa - \frac{t}{T} \kappa, \tag{26}$$

where  $\kappa = 2$ ,  $t$  is the iteration number, and  $T$  is the maximum number of iterations.

**Binary optimization algorithms**

We use a binary optimization algorithm for feature selection. To convert continuous versions of OPA, SCOA, PSO and COA into binary versions, the transfer functions are used. Our research uses an S-shaped transfer function to convert a continuous version into a binary version. It was reported that the S-shaped transfer function had the highest accuracy for solving complex problems.

$$S(SO_{m,j}^t) = \frac{1}{1 + e^{-x_{m,j}^t}}, \tag{27}$$

$$SO_{m,j}^t = \begin{bmatrix} 1 \leftarrow rand < S(SO_{m,j}^t) \\ 0 \leftarrow rand \geq S(SO_{m,j}^t) \end{bmatrix}, \quad (28)$$

where  $SO_{m,j}^t$  is the  $j$ th solution and  $S(SO_{m,j}^t)$  is the transfer function.

### Hybrid GE-POA-DTR model

Our study uses the GE-POA-DTR to monthly rainfall. One of the main advantages of the new model is that it can capture spatiotemporal patterns. The new model can be used in different climates. The classical machine learning models are sensitive to irrelevant features. These features may reduce the accuracy of machine learning models. The DRSN-TCN-RF (DTR) model is a robust model for handling complex data sets as it discards irrelevant data. We follow the following levels to create the new model:

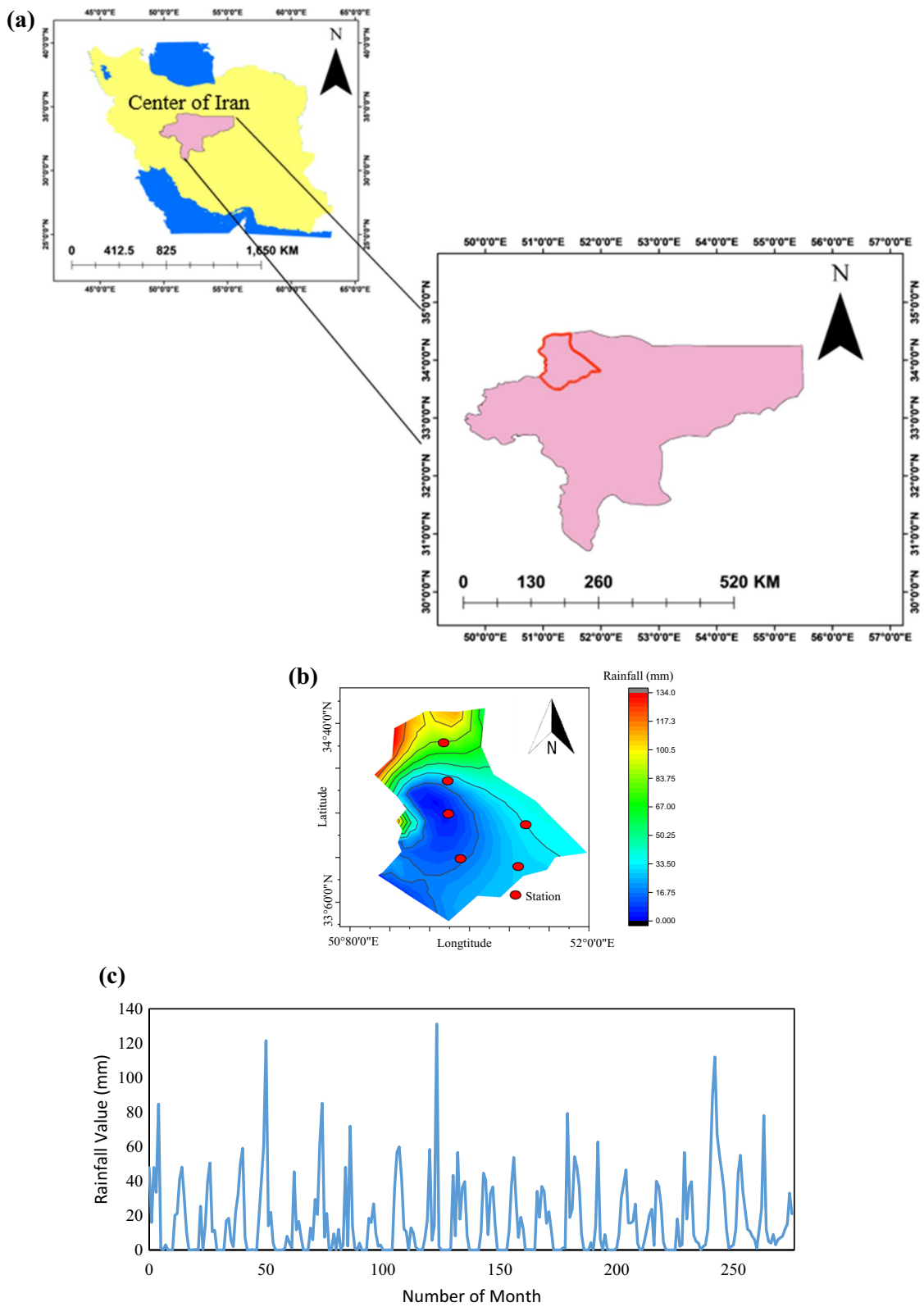
- Determining the data size is essential for running models at the training and testing levels. The K-fold cross-validation method (KSV) is a suitable tool for classifying data into training and testing data. Data points are divided into distinct groups using K folds [36]. Then, predictive models are executed k times. The error function values are calculated for each run [23]. To evaluate the performance of models in predicting output, the average error function value is computed. K-fold cross-validation is a valuable technique for preventing overfitting. As the KSV uses different subsets of data to train and test models, it can reduce the risk of overfitting.
- The decision variables are defined. The input names and model parameters are decision variables. The initial algorithm population consists of these decision variables.
- The DRSN-TCN model is executed and an error function is calculated to evaluate its performance. Based on threshold values, the DRSN model determines important features. The TCN then uses the key features to predict rainfall patterns.
- If the stop criterion is met, the model will enter the testing level; otherwise, we link the DRSN-TCN model with the optimizers. Operators of optimizers are used to update the input combination and model parameters. The process continues until the convergence criterion is achieved.
- The testing data are used to run the DRSN-TCN model.
- The outputs of the DRSN-TCN are used as inputs to the RF model. The RF model is used to predict monthly rainfall data at training and testing levels.

Appendix A: Table 3 shows list of variables.

### Case study

The Kashan Plain is a plain in central Iran. The Kashan Plain is characterized by a dry climate and desert and semi-desert landscapes. The summers are hot and dry, while the winters are cool. Traditional techniques such as qanats (underground water channels) are used to meet water needs. In the Kashan plain, irrigation and crop production depend on rainfall. Rainfall is an important source of water in the Kashan Plain. The data were collected from 1992 to 2015. April, May, and June are the wettest months, while July, August, and September are the driest months. Kashan Plain experiences high temperatures during summer. The average daily temperature ranges from 30 to 40 degrees Celsius. Monthly rainfall varies from 0 to 134 mm. Figure 1a shows the plain location. Figure 1b displays the location of the stations. Figure 1c displays time-series data. We used lagged rainfall data to predict one-month ahead rainfall. The lag times of (t-1) to (t-12) were used to predict outputs. Rainfall often exhibits temporal patterns and dependencies. Lagged rainfall data capture these temporal relationships. Historical data sets are required for training predictive models. Machine learning models require past data to learn the underlying patterns and relationships. A predictive model can gain a better understanding of historical patterns by considering lagged rainfall data. Seven stations were used because the new model could achieve the lowest values of the error function. The Thiessen polygon method was used to determine the average monthly rainfall values for the entire catchment area. The basin was divided into several polygons. Weight coefficients were assigned to the polygons to obtain average precipitation values (Table 1).

We collect monthly rainfall data from monitoring stations. We identify the basin boundary, which is used for creating Thiessen polygons. A Thiessen Polygon (Voronoi Polygon) is created around each monitoring station. We calculate the area of each Thiessen polygon. The area of each polygon represents the influence zone of the corresponding monitoring station. The weight of each polygon is calculated by estimating its area. These weights are multiplied by observed rainfall values at monitoring stations. To estimate monthly rainfall for the entire basin area, we sum the weighted rainfall values of all monitoring stations for each month. To assess the efficiency of models, the following error functions are used:



**Fig. 1** . a Location of plain on Iran map, b location of stations, and c time-series data



**Table 1** The details of data points

Level	Maximum	Minimum	Average
Train	134	0	45.67
Test	128	0	48.89

**Willmott index**

$$WI = 1 - \frac{\sum_{j=1}^n (RI_{iob} - RI_{ies})^2}{\sum_{i=1}^n (|RI_{iob} - \bar{RI}| + |RI_{ies} - \bar{RI}|)^2}. \tag{29}$$

**Root mean square error (RMSE)**

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (RI_{ob} - RI_{es})^2}. \tag{30}$$

**Mean absolute error (MAE)**

$$MAE = \frac{1}{n} \sum_{i=1}^n |OC_{es} - OC_{ob}|. \tag{31}$$

**Percent bias (PBIAS)**

$$PBIAS = \frac{\sum_{i=1}^N (RI_{ob} - RI_{es})}{\sum_{i=1}^N RI_{ob}}. \tag{32}$$

**U95**

$$U_{95} = 1.96 \left( SD^2 + RMSE \right)^{\frac{1}{2}}, \tag{33}$$

where  $RI_{ob}$  is the observed rainfall,  $RI_{es}$  is the estimated rainfall,  $SD$  is the standard deviation,  $\bar{RI}$  is the average observed rainfall. Our study also uses Eq. 34 to evaluate the ability of optimization algorithms in choosing inputs:

$$Average(selection(size)) = \frac{1}{M} \sum_{i=1}^M \frac{size(so^i)}{NU}, \tag{34}$$

where  $M$  is the number of runs,  $NU$  is the number of total inputs, and  $size(so^i)$  is the size of chosen inputs at each run. The best algorithm has the lowest value of average selection size (ASS).

**Results**

**Determination of number of K**

Calculating  $K$  is necessary for the modeling process and splitting the data into training and testing data points.

Figure 2 shows the RMSE values for different  $K$  numbers. The DTB model was used based on different sizes of  $K$ .  $K=10$ ,  $K=5$ , and  $K=2$  provided RMSE values of 0.187–0.235, 0.187–0.270, and 0.189–0.276, respectively. Thus, we chose  $K=10$  to run models.

**Computation of the values of random parameters**

Adjusting the random parameters is necessary to achieve the highest accuracy of optimizers. Optimal parameter values minimize the RMSE of optimizers. The maximum number of iterations (MNI) is one of the most important random parameters. Population size (PS) also changes the accuracy of optimizers. The GM-OPA and OPA had the best performance (lowest RMSE) based on  $PS=120$  and  $MNI=300$ .  $PS=180$  and  $MNI=300$  provided the highest precision for the SCOA (Fig. 3).

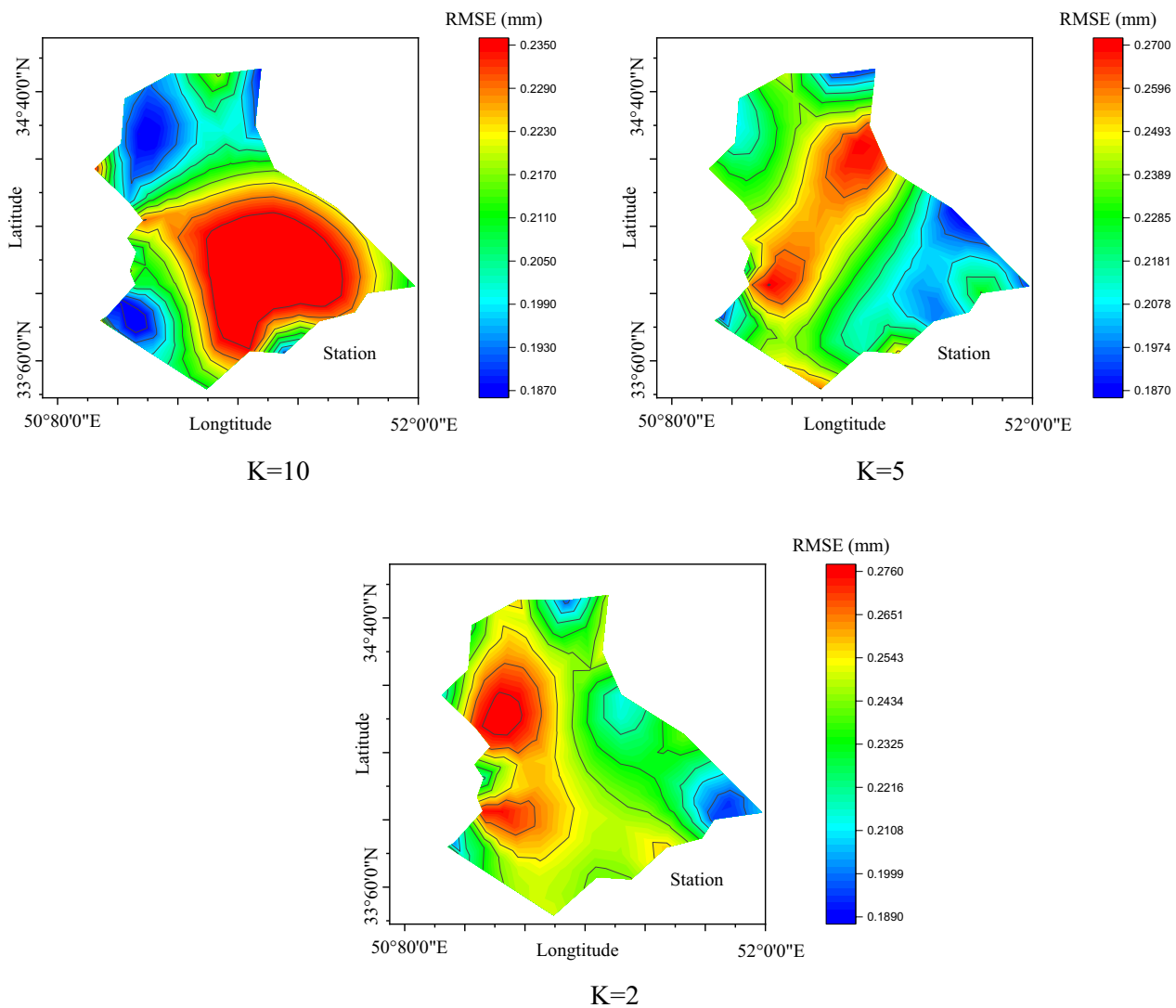
**Choice of the best optimizers and input scenarios**

Figure 4 shows accuracy optimizers for choosing inputs. The RMSE of the GM-OPA varied from 0.180 to 0.200 mm. The GM-OPA reduced the RMSE values of OPA and PSO by 1.4%–3.4% and 6.14–9.54%, respectively. Thus, GM-OPA improved the performance of the ROA.

Figure 5 shows the percentage of features selected. Also, this figure shows ASS values of different algorithms. The GM-OPA, OPA, SCOA, COA, and PSO chose 33%, 41%, 41%, 50%, and 50% of the total number of inputs. The ASS of the GM-ROA was lower than that of other algorithms. Thus, the GM-ROA achieved the highest accuracy with fewer input variables. The GM-OPA selected rainfall (t-1), rainfall (t-2), and rainfall (t-3) as predictors for rainfall prediction. Table 2 shows the optimal values of hybrid model parameters. The optimization algorithm systematically searches the problem space to find the optimal values of the model parameters. First, we initialize the initial values of the model parameters. The GM-OPA then updates the values of the parameters at each iteration. When the GM-OPA converges to the lowest value of the objective function value, the process is completed. The optimal values of model parameters are reported and used to train and test models. Although there are  $2^{12}-1$  input scenarios, our model chose the optimal input scenario with three input variables. Thus, the optimizers played a key role in reducing the computational cost of the modeling process.

**Evaluation of the performance of models**

Figure 6 displays the error function values of different models. The DTR reduced the testing MAE values of the TCN-RE, DRSN-TCN, TCN, and RF models by 5.3%, 21%, 40%, and 46%, respectively. The RF provided the highest MAE values. The testing MAE of the RF was 46%, 41%, 32%, and 6.4% was higher than that of the TCN-RE,



**Fig. 2** The determination of K

DTR–TCN, and TCN models. The DRSN model improve the accuracy of the TCN model. The testing MAE of the DRSN–TCN was 28% lower than that of the TCN model. The training NSE of the DTR, TCN–RF, DRSN–TCN, TCN, and RF models was 0.98, 0.95, 0.92, 0.89, and 0.87, respectively. The testing NSE of the DTR and TCN–RF was 0.96 and 0.94. Thus, the DTR and TCN–RF outperformed the other models. The TCN model outperformed the RF model as it used advanced layers and operators for analyzing data. The testing NSE of the TCN model was 3.5% higher than that of the RF model. The DTR model had the best performance as it used the DRSN mechanism to remove irrelevant data.

The DTR reduced the testing WI values of the TCN–RF, DRSN–TCN, TCN, and RF models by 4.16%, 7.29%,

14%, and 16%, respectively. The training PBIAS of the DTR, TCN–RF, DRSN–TCN, TCN, and RF models was 3, 7, 11, 14, and 18, respectively. The testing U95 of the DTR, TCN–RAF, DRSN–TCN, TCN, and RAF models was 8, 12, 16, 18, and 22, respectively. U95 is an index that can evaluate the accuracy of models based on RMSE and standard devalues. The U95 of the RF model was higher than that of the other models as irrelevant models influence the reliability of the RF model. The DTR model outperformed all models because it leveraged the capabilities of three models. The DRSN component of the DRSN–TCN model employs soft thresholding to identify and eliminate irrelevant features of the rainfall data. The model can control the impact of less relevant features by dynamically setting thresholds. The DRSN–TCN uses

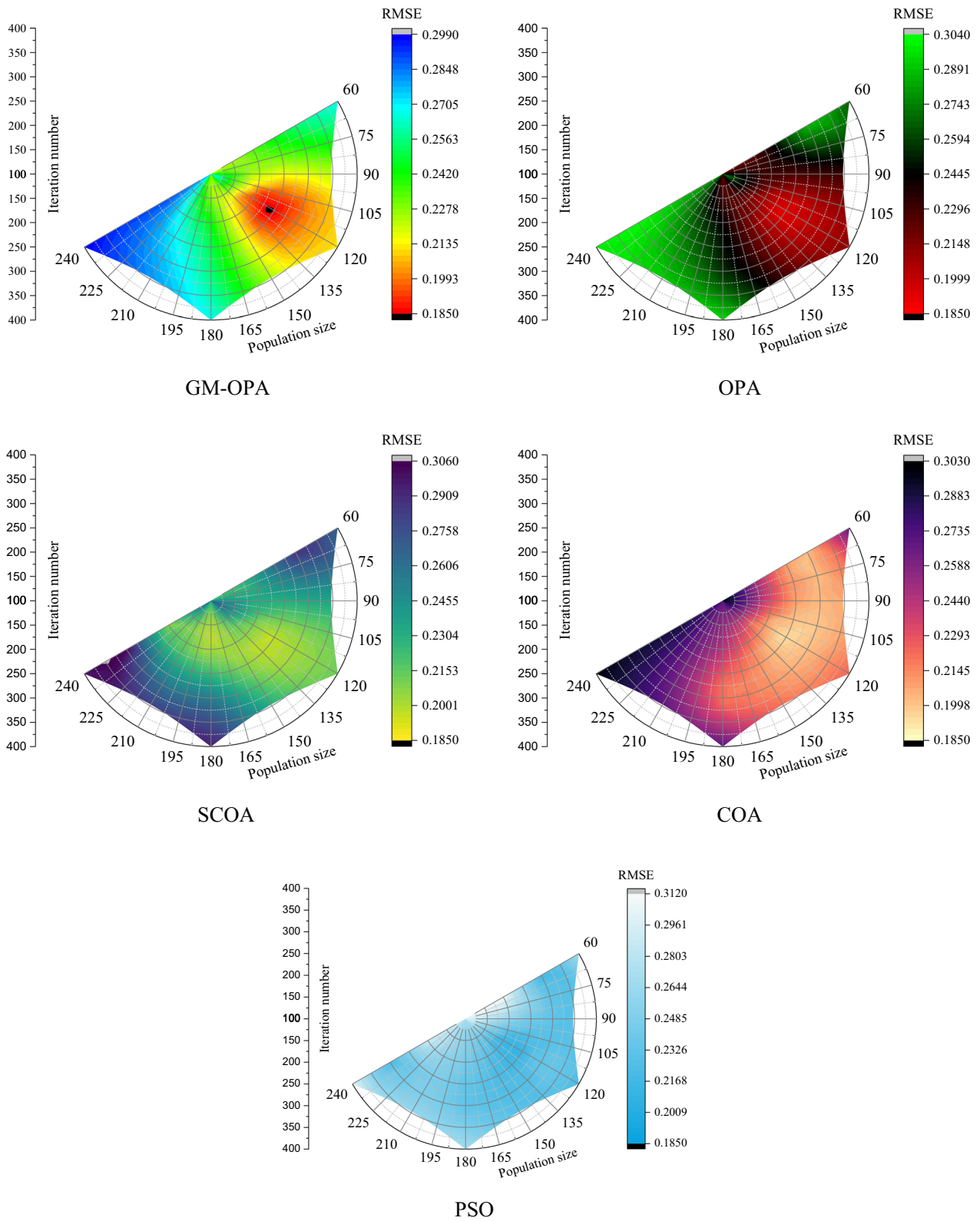
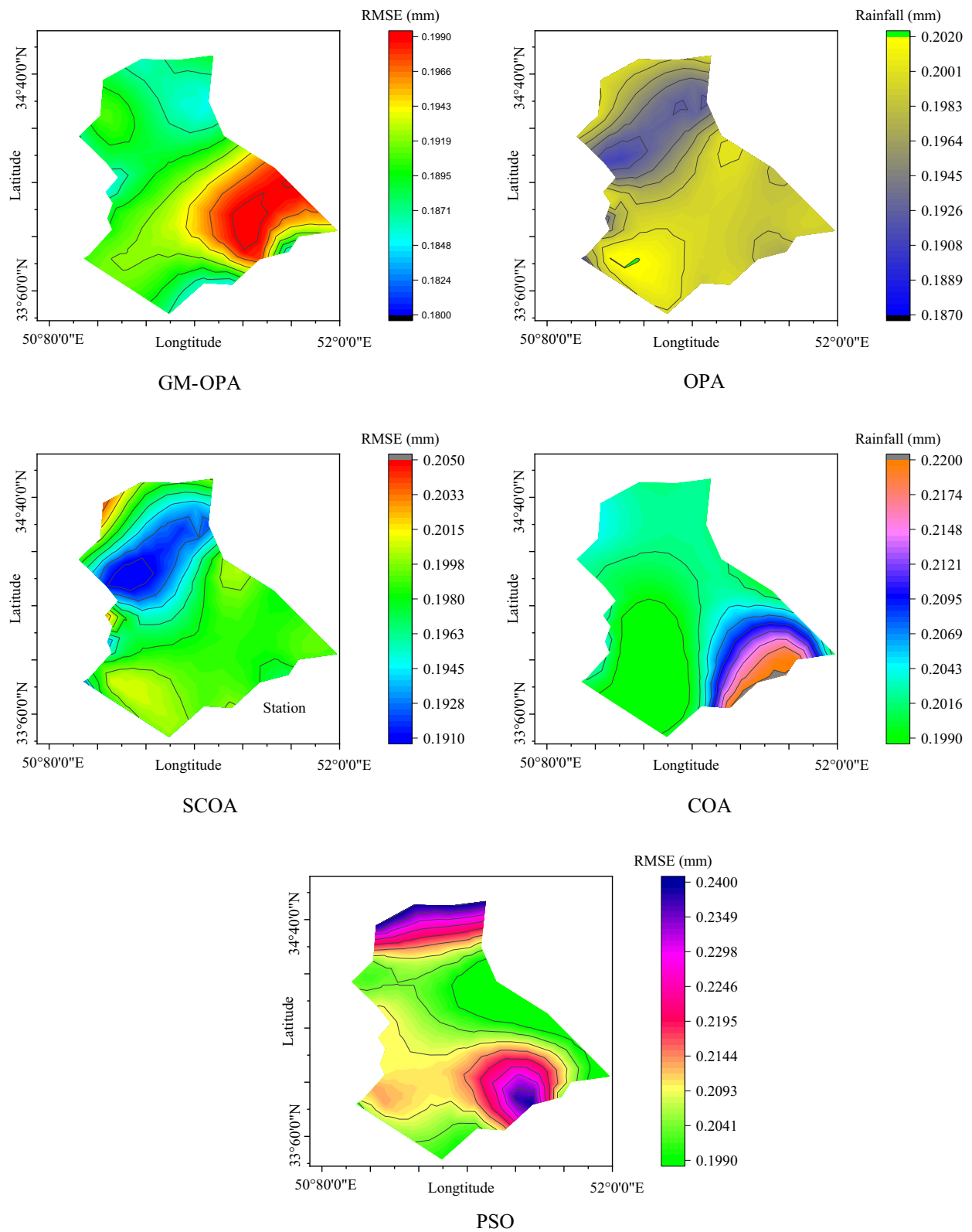
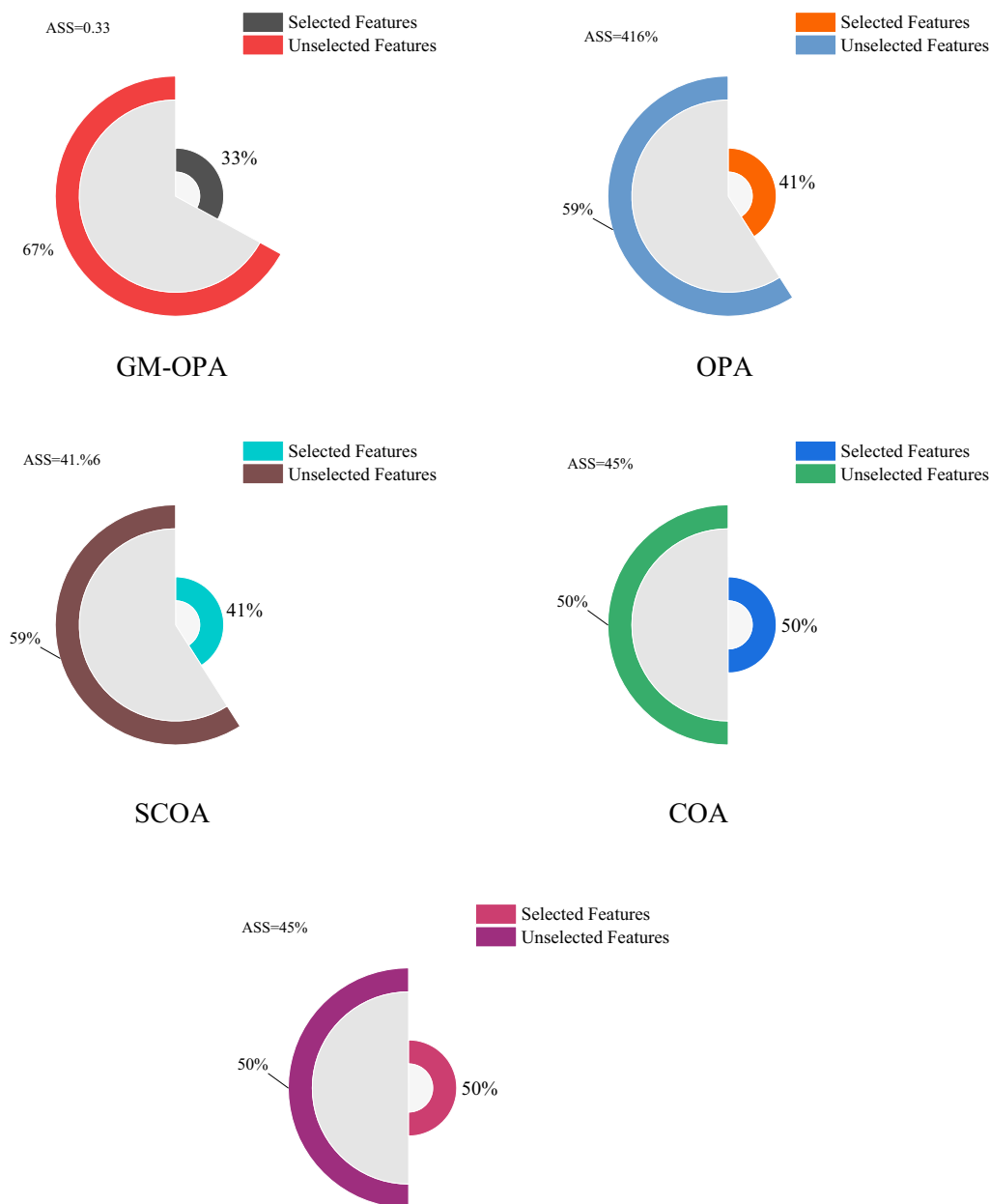


Fig. 3 Determination of random parameters



**Fig. 4** The spatial values of RMSE of different models



**Fig. 5** The Percentage of feature selected numbers

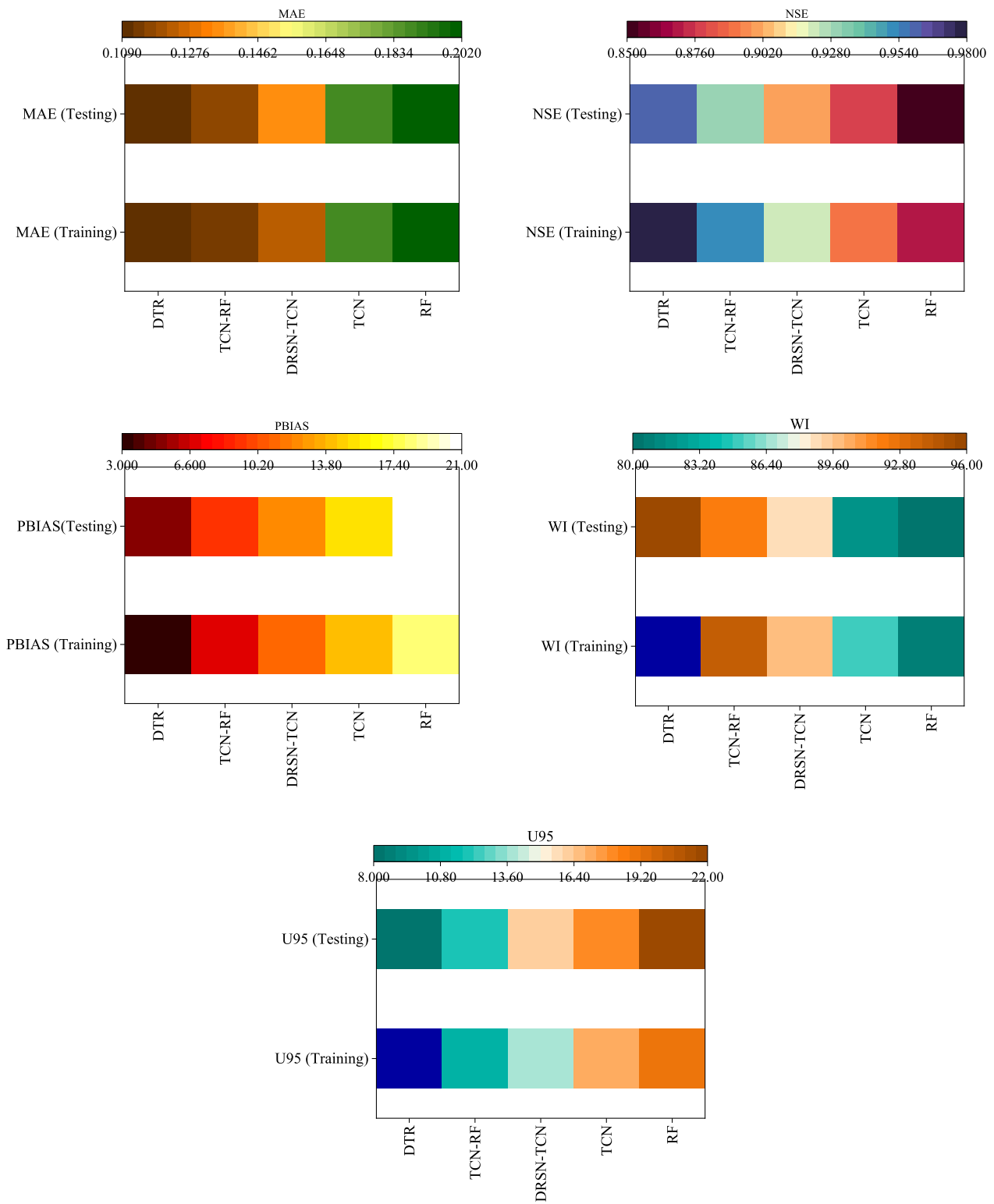
**Table 2** Optimal values of TCN parameters

Parameter	Value
Learning rate	0.001
dropout	0.05
Number of filters	3
Batch size (BS)	BS (4, 20)

soft thresholding to reduce the influence of irrelevant features of a dataset. By effectively reducing irrelevant features, the DRSN–TCN model can improve the generalization ability of the model.

Figure 7 displays heat scatterplots of models. The  $R^2$  value of the DTR, TCN–RF, DRSN–TCN, TCN, and RAF models was 0.9916, 0.9848, 0.9791, 0.9691, and 0.9400, respectively.





**Fig. 6** Error function values of different models

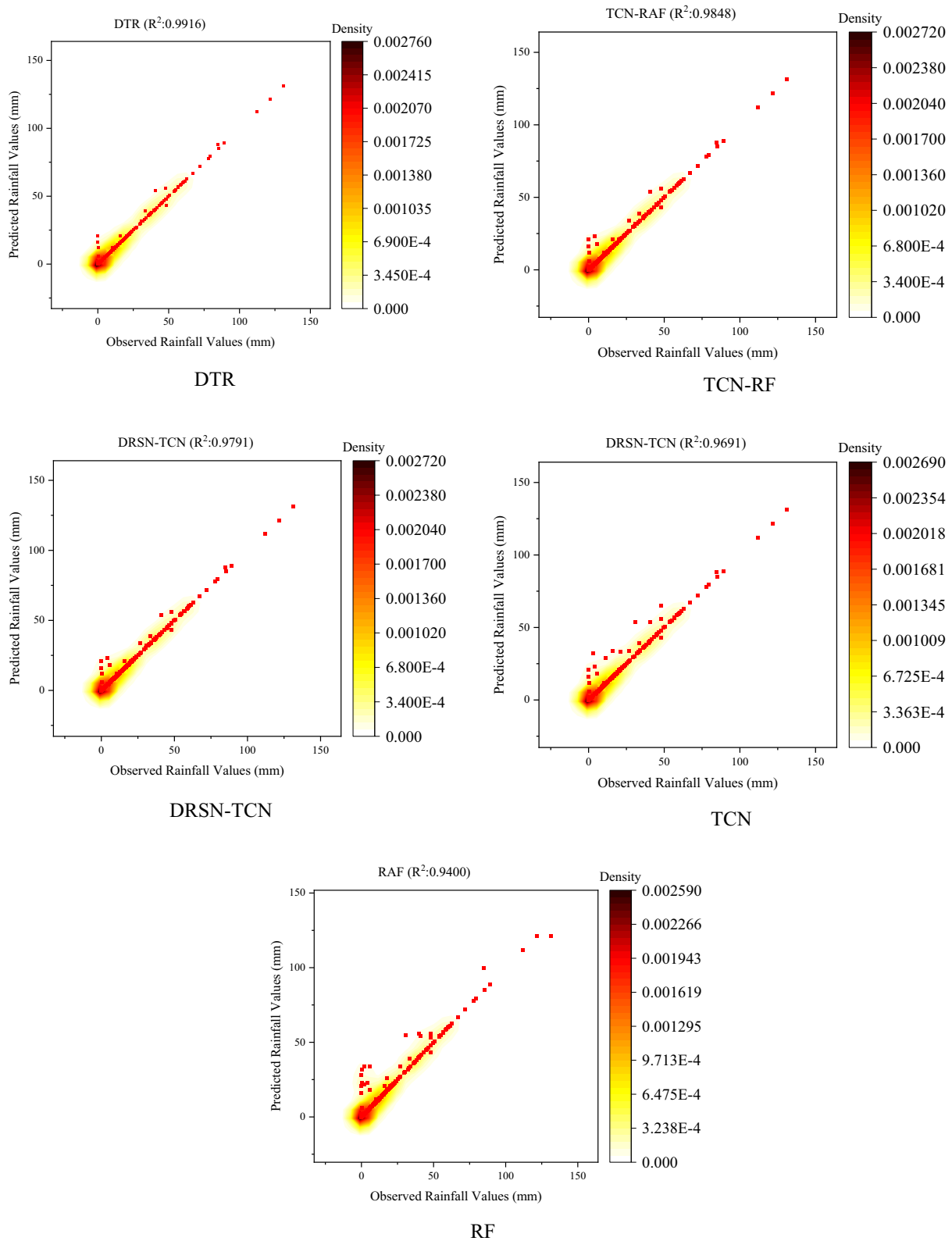


Fig. 7 Heat scatterplots of models

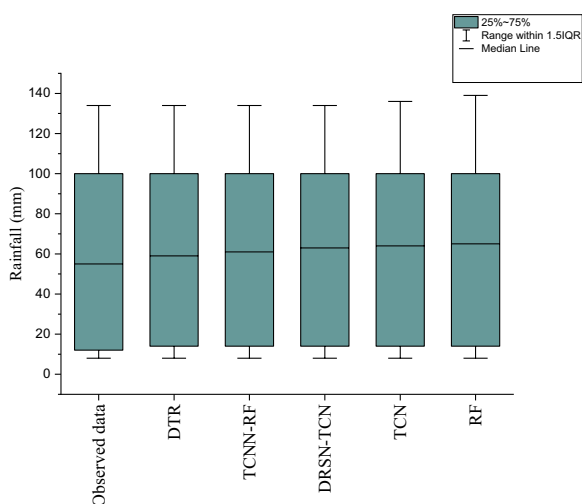


Fig. 8 Boxplots of models for comparing models

The study results indicated that DTR and TCN–RF performed better than the other models. The DRSN mechanism could improve the performance of the TCN model. The  $R^2$  value of the DRSN–TCN model was 1.02% higher than that of the TCN model. However, the irrelevant features reduce the accuracy of the RF model. The RF model had the lowest  $R^2$  values among other models.

Figure 8 displays boxplots of different models. The median of the observed data, DTR, TCN–RF, DRSN–TCN, TCN, and RF models was 55 mm, 59 mm, 61 mm, 63 mm, 64 mm, and 65 mm, respectively. The maximum value of the observed data, DTR, TCN–RF, DRSN–TCN, TCN, and RF model was 134 mm, 134 mm, 134 mm, 134 mm, 135 mm, and 136 mm, respectively.

The study results revealed that the DTR had the best performance among the other models. The DRSN was introduced to address the limitations of the TCN model and improve the quality of the extracted features. The DRSN ensures that the hybrid model concentrates on the most relevant information, which results in more accurate predictions. In summary, the DRSN–TCN–random forest model outperforms the TCN and RAF models because it enhances the quality of extracted features, focuses on relevant information, and enhances efficiency. The TCN model captures non-linear relationships.

Our study developed the GM–OPA to set parameter values and select the best input scenario. As the GM improved the ability of the POA to locally and globally search the problem space, it outperformed other optimizers. The GM–OPA chose the best input scenario based on the fewest number of input variables.

Taylor diagram is a graphical tool for evaluating the performance of the models. At the center of the diagram,

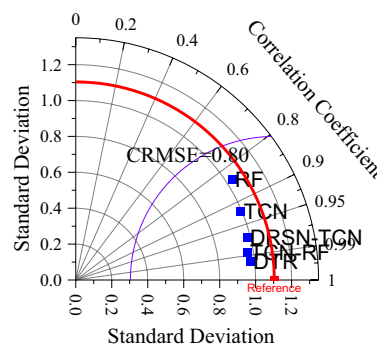
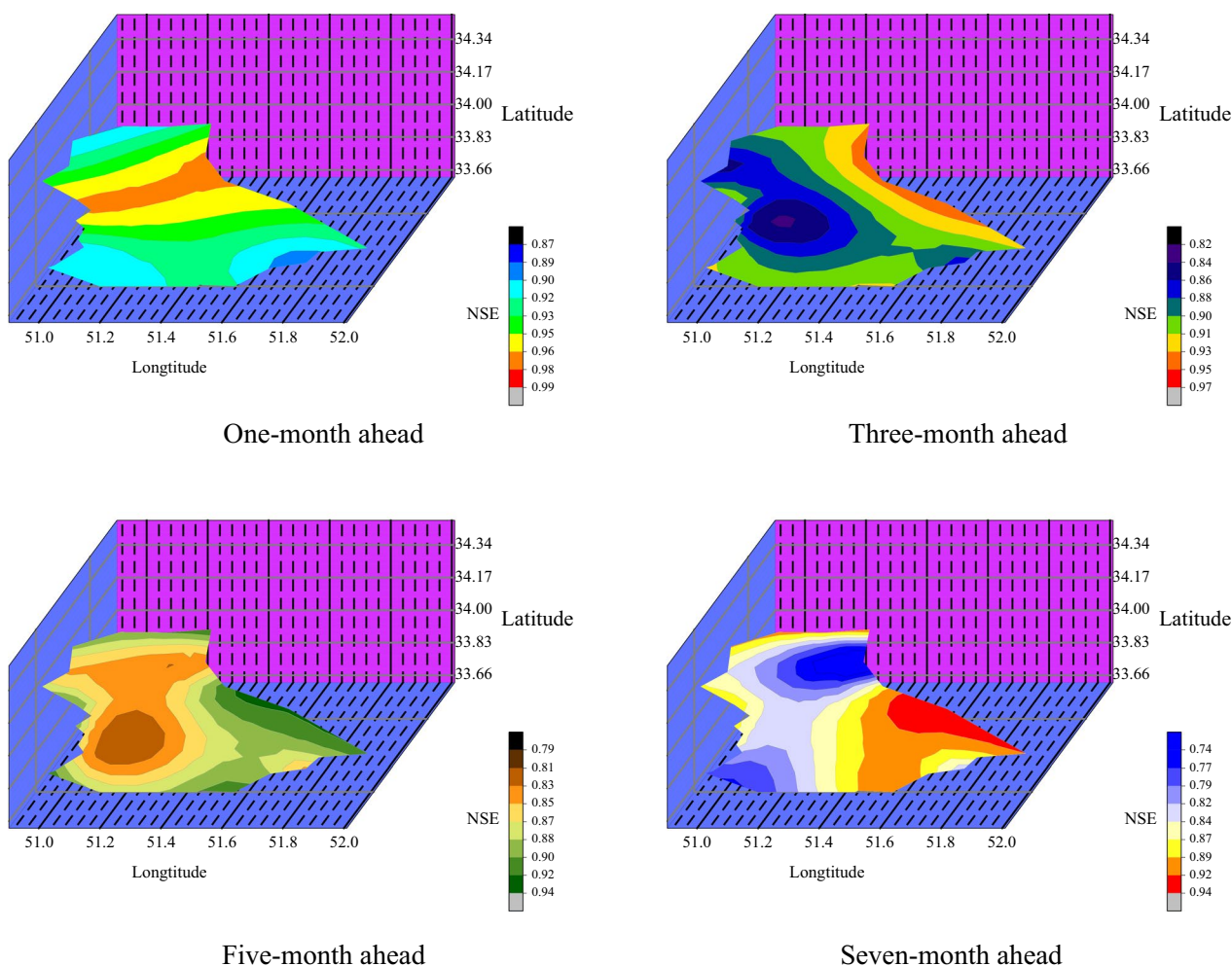


Fig. 9 Taylor diagram for comparing models

a point is represented as the reference dataset. The radial axis represents the standard deviation. The tangential axis represents the correlation coefficient. Contour lines represent centralized root mean square (CRMSE) values. Figure 9 shows a Taylor diagram to compare models at the testing level. The CRMSE of DTR, TCN–RF, DRSN–TCN, TCN, and RF was 0.15, 0.19, 0.26, 0.39, and 0.56, respectively. The correlation coefficients of DTR, TCN–RF, DRSN–TCN, TCN, and RF were 0.99, 0.98, 0.97, 0.92, and 0.84, respectively. The CRMSE and correlation values indicated that the DRSN mechanism improved the performance of the TCN model. The DRSN–TCN improved the CRMSE and correlation values of the TCN model by 19% and 5%, respectively. The RF model provided the highest CRMSE model. The DTR model improved the CRMSE of the RF model by 73%.

### Discussion

While we use predictive models to predict one-month ahead rainfall, it is required to examine their accuracy for predicting other time horizons. We use the DTR as the best model to predict one-, three-, five-, and seven-month ahead rainfall. Figure 10 shows the spatial accuracy of the DTR model for multi-step-ahead rainfall prediction. The DTR model provided NSE values of 0.87–0.99, 0.82–0.97, 0.79–0.94, and 0.70–0.94 for 1, 3, 5, and 7 months ahead. As the time horizon increases, the efficiency of models decreases. Climate phenomena such as large climate indices can impact long-term patterns. Predicting these large-scale patterns can be difficult and complex. Human activities such as deforestation, urbanization and greenhouse gas emissions can change rainfall patterns over long periods. These factors may increase the uncertainty of long-term rainfall predictions and affect the accuracy of models.



**Fig. 10** NSE values of different models for predicting different horizon times

The DTR model is a useful model as we can use its outputs for water resource management planning and other hydrological simulations. To demonstrate the practical applications of the DT model, we use it to forecast monthly rainfall values for the period 2023–2024. Figure 11 shows the monthly rainfall values for the period 2023–2024. The month of January had the highest rainfall values. In January 2023, 8–40 mm of rainfall fell. The lowest rainfall values were observed in the summer months. In the autumn and winter months, the average rainfall was 21.12 and 18.78 mm, respectively.

Our models also can be used to predict rainfall at different time scales. We use DTR, TCN–RF, DRSN–TCN, TCN, and RF models to predict daily data. Figure 12 displays the absolute relative error (ABR) of daily predictions of different models. The DTR model gave an ABR of 1–5% for 4% of the data points. The RF model gave the highest values of ABR. The MLR model provided an ABR of 19–23% for 10% of the data points. The training NSE

of the DTR, TCN–RF, DRSN–TCN, TCN, and RF models was 0.98, 0.95, 0.92, 0.89, and 0.87, respectively.

The DTR model effectively removes irrelevant data and extracts temporal features from rainfall data. The DTR model is required for improving prediction accuracy and assessing climate change impact. Our study creates a superior rainfall prediction model, develops innovative optimization techniques, and demonstrates the superior performance of the proposed model. The study findings have practical implications for climate research, water resource management, and disaster preparedness. The new model can be used for understanding water availability, managing reservoirs, and planning for potential floods or droughts. Water resource managers can use the new model to plan for regional variations of rainfall. The model contributes to climate resilience planning as it provides insights into future rainfall trends.

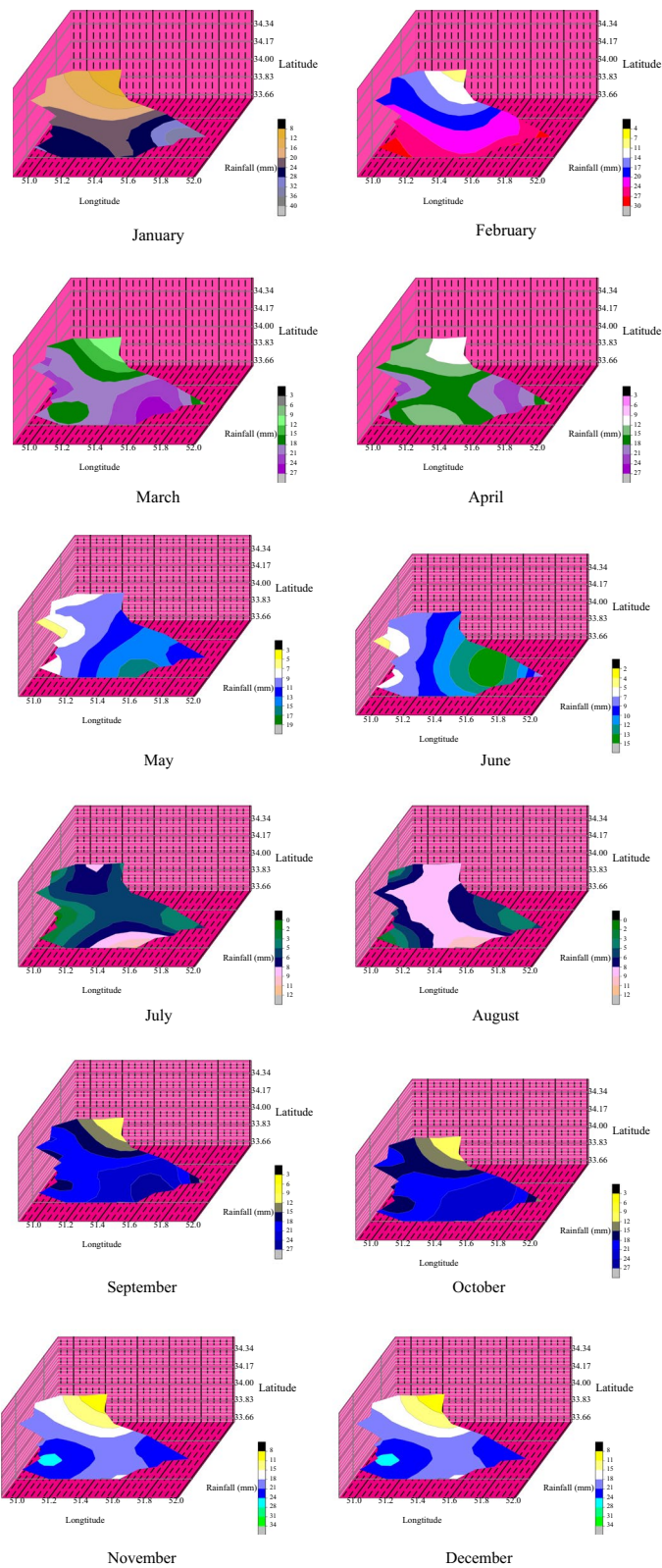
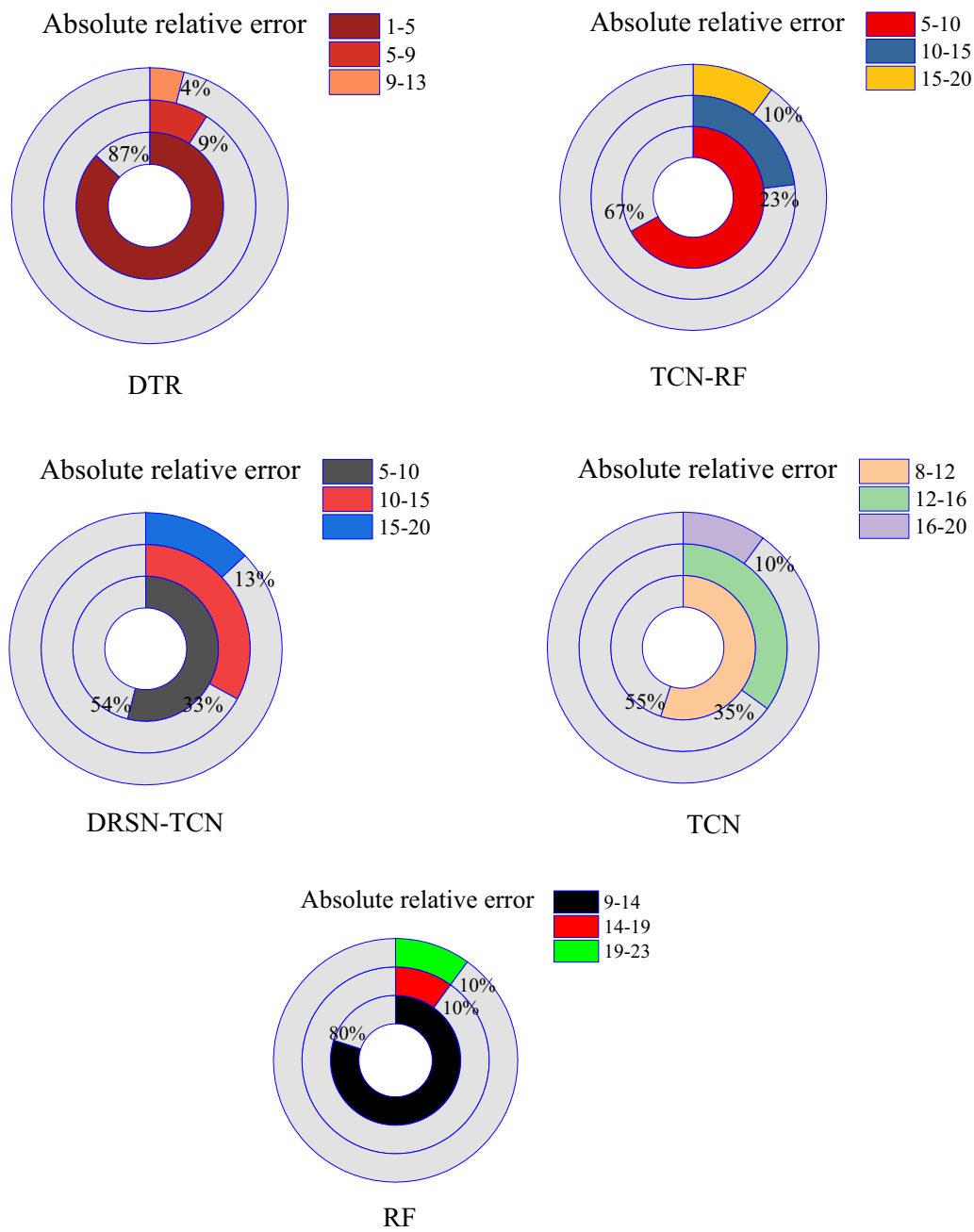


Fig. 11 Monthly rainfall values for the period 2023–2024





**Fig. 12** The ABR of different models for predicting daily data

Based on the analysis of the results, the following points should be considered:

- The GM-OPA can intelligently select a subset of features from the original input space. The algorithm can dynamically adjust the complexity of the model

based on the evolving characteristics of the data. The GM-OP can identify and retain only the most relevant features for the target variable. The new algorithm can reduce the dimensionality of the input space as it chooses the most important inputs.

- Irrelevant features increase the complexity of the model without providing meaningful information. When irrelevant features are removed, the model becomes simpler and focuses on the important features of the data. The DRSN method can reduce the risk of overfitting as it can remove irrelevant features.
- The DTR–TCN–RF model is a robust model for spatiotemporal predictions. As the TCN model uses convolutional layers and dilated convolutions, it can capture spatiotemporal patterns.
- The GM–OPA had two main advantages. The algorithm achieved the highest accuracy. Also, the new algorithm chose fewer inputs for predicting rainfall.
- The study results showed that the RF model could not achieve high accuracy because it could not analyze non-linear patterns.

### Conclusion

Accurate rainfall predictions are required for water resources management, including reservoirs, dams, and groundwater recharge. Our study introduced a new model named the DTR to predict 1 month ahead. The DTR consisted of three models. The DRSN was used to remove redundant data. The RAF model received temporal features from the TCN model. Finally, The RF model made rainfall predictions. We developed a novel optimizer called GM–ROA to select inputs. Lagged rainfall values were applied to predict outputs. The study results indicated that GM–ROA improved the ability of the ROA for feature selection. The RMSE of the GM–ROA varied from 0.180 to 200 mm. The GM–ROA reduced the RMSE values of ROA and PSO by 1.4%–3.4% and 6.14–9.54%, respectively. The study results revealed that the DTR outperformed the other models. The DRSN is specifically designed to enhance the quality of the extracted features. The DTR model also could predict daily data with high accuracy. The new model also accurately predicted monthly data for the period 2023–2024. Our study uses the TCN model which processes information in one direction. The execution of these models may have limitations. To run these models. we need advanced computer systems. Data collection may be another limitation of the study. The next studies can develop the TCN model to process information in two directions. In addition, we can use different methods such as the analysis of variance method (ANOVA) to decompose uncertainty values into parameter and input uncertainties.

### Appendix A

See Table 3.

**Table 3** List of variables

Variables	Definition
$AP_j^t$	The awareness probability of the crow
$CR_j^{it}$	The $i$ th location of crow in the $j$ th dimension
$e$	random number
$F$	The controller parameter
$f(OR_{chase,i}^t)$	The objective function of the new location of the orca
$f(OR_i^t)$	The objective function value of the current location of the orca
$J$	The average location of one group
$M_j^i$	The $i$ th memory in the $j$ th dimension
$\max(it)$	maximum number of iterations
$NU$	number of total inputs
$or_N$	The location of the $N$ th orca
$or_1$	The location of the first orca
$OR$	A group of orcas
$OR_{chase,1,i}$	The location of orca based on the first scenario
$OR_{chase,2,i}$	The location of orca based on the second scenario
$OR_{best}^t$	The location of best orca
$p_i$	The local best solution
$p_g$	The global best solution
$r_1$	Random numbers,
$R_{lob}$	Observed rainfall
$R_{les}$	Estimated rainfall
$s_i(t)$	The $i$ th solution
$SO_j^{t+1}$	The new solution
$SO_{mj}^t$	The $j$ th- solution
$t$	iteration number
$T$	maximum number of iterations
$u$	Random number
$ve_i(t + 1)$	The $i$ th velocity at iteration $(t + 1)$
$VE_{chase,1,i}$	The velocity of orca based on the first scenario
$VE_{chase,2,i}$	The velocity of orca based on the second scenario
$x$	Input feature
$X_j^{t+1}$	The $j$ th location at $t + 1$ iteration
$Y_j^t$	The current location of crow
$z$	Output
$\mu$	random parameter
$\beta$	Random number
$\varphi$	Threshold value
$\alpha$	Random number

**Author contributions**

MAN: formal analysis and writing, FP: formal analysis and writing and ME, writing, review, and editing, and forma analysis, HS: writing, review, and editing.

**Funding**

No funding.

**Data availability**

The data presented in this research are available upon request from the corresponding author.

**Declarations****Ethics approval and consent to participate**

We consent to participate in the "Ethical responsibilities of Authors of Enviro Monitoring & Assessment journal.

**Consent for publication**

Not applicable.

**Competing interests**

The authors state no competing interests.

**Author details**

<sup>1</sup>Department of Water Engineering, Semnan University, Semnan, Iran. <sup>2</sup>Faculty of Natural Resources and Earth Sciences, University of Kashan, Kashan, Iran.

<sup>3</sup>Department of Computer Science, Western New England University, Springfield, MA, USA.

Received: 23 November 2023 Accepted: 5 January 2024

Published online: 12 January 2024

**References**

- Afshari Nia M, Panahi F, Ehteram M (2023) Convolutional Neural Network-ANN- E (Tanh): a new deep learning model for predicting rainfall. *Water Resour Manage*. <https://doi.org/10.1007/s11269-023-03454-8>
- Ali M, Prasad R, Xiang Y, Yaseen ZM (2020) Complete ensemble empirical mode decomposition hybridized with random forest and kernel ridge regression model for monthly rainfall forecasts. *J Hydrol*. <https://doi.org/10.1016/j.jhydrol.2020.124647>
- Azad A, Manoochehri M, Kashi H, Farzin S, Karami H, Nourani V, Shiri J (2019) Comparative evaluation of intelligent algorithms to improve adaptive neuro-fuzzy inference system performance in precipitation modelling. *J Hydrol*. <https://doi.org/10.1016/j.jhydrol.2019.01.062>
- Chen J, Lin W, Cai S, Yin Y, Chen H, Towsy D (2023) BITCN\_DRSN: An effective software vulnerability detection model based on an improved temporal convolutional network. *J Syst Softw*. <https://doi.org/10.1016/j.jss.2023.111772>
- Choubin B, Malekian A, Samadi S, Khalighi-Sigaroodi S, Sajedi-Hosseini F (2017) An ensemble forecast of semi-arid rainfall using large-scale climate predictors. *Meteorol Appl*. <https://doi.org/10.1002/met.1635>
- Choubin B, Zehetabian G, Azareh A, Rafiei-Sardooi E, Sajedi-Hosseini F, Kişi Ö (2018) Precipitation forecasting using classification and regression trees (CART) model: a comparative study of different approaches. *Environ Earth Sci*. <https://doi.org/10.1007/s12665-018-7498-z>
- Devarapalli R, Venkateswara Rao B, Dey B, Vinod Kumar K, Malik H, Garcia Marquez FP (2022) An approach to solve OPF problems using a novel hybrid whale and sine cosine optimization algorithm. *J Intell Fuzzy Syst*. <https://doi.org/10.3233/JIFS-189763>
- Elbeltagi A, Zerouali B, Bailek N, Bouchouicha K, Pande C, Santos CAG, Towfiqul Islam ARM, Al-Ansari N, El-kenawy E-SM (2022) Optimizing hyperparameters of deep hybrid learning for rainfall prediction: a case study of a Mediterranean basin. *Arab J Geosci*. <https://doi.org/10.1007/s12517-022-10098-2>
- Gracia-Velásquez DG, Morales-Rodríguez AS, Montoya OD (2022) Application of the crow search algorithm to the problem of the parametric estimation in transformers considering voltage and current measures. *Computers*. <https://doi.org/10.3390/computers11010009>
- Gu J, Liu S, Zhou Z, Chalov SR, Zhuang Q (2022) A stacking ensemble learning model for monthly rainfall prediction in the Taihu basin. *China Water* (Switzerland). <https://doi.org/10.3390/w14030492>
- Han T, Zhang Z, Ren M, Dong C, Jiang X, Zhuang Q (2023) Speech emotion recognition based on deep residual shrinkage network. *Electronics* (Switzerland). <https://doi.org/10.3390/electronics12112512>
- Hussein EA, Ghaziasgar M, Thron C, Vaccari M, Jafta Y (2022) Rainfall prediction using machine learning models: literature survey. *Stud Comput Intell*. [https://doi.org/10.1007/978-3-030-92245-0\\_4](https://doi.org/10.1007/978-3-030-92245-0_4)
- Jiang Y, Wu Q, Zhu S, Zhang L (2022) Orca predation algorithm: a novel bio-inspired algorithm for global optimization problems. *Expert Syst Appl*. <https://doi.org/10.1016/j.eswa.2021.116026>
- Johny K, Pai ML, S, A. (2022) A multivariate EMD-LSTM model aided with time dependent intrinsic cross-correlation for monthly rainfall prediction. *Appl Soft Comput*. <https://doi.org/10.1016/j.asoc.2022.108941>
- Kisi O, Mohsenzadeh Karimi S, Shiri J, Keshavarzi A (2019) Modelling long term monthly rainfall using geographical inputs: assessing heuristic and geostatistical models. *Meteorol Appl*. <https://doi.org/10.1002/met.1797>
- Kisi O, Shiri J (2014) Prediction of long-term monthly air temperature using geographical inputs. *Int J Climatol*. <https://doi.org/10.1002/joc.3676>
- Kumar Sharma D, Brahmachari S, Singhal K, Gupta D (2022) Data driven predictive maintenance applications for industrial systems with temporal convolutional networks. *Comput Ind Eng*. <https://doi.org/10.1016/j.cie.2022.108213>
- Li W, Wei Y, An D, Jiao Y, Wei Q (2022) LSTM-TCN: dissolved oxygen prediction in aquaculture, based on combined model of long short-term memory network and temporal convolutional network. *Environ Sci Pollut Res*. <https://doi.org/10.1007/s11356-022-18914-8>
- Li D, Jiang F, Chen M, Qian T (2022) Multi-step-ahead wind speed forecasting based on a hybrid decomposition method and temporal convolutional networks. *Energy*. <https://doi.org/10.1016/j.energy.2021.121981>
- Lotfirad M, Esmaeili-Gisavandani H, Adib A (2022) Drought monitoring and prediction using SPI, SPEI, and random forest model in various climates of Iran. *J Water Clim Change*. <https://doi.org/10.2166/wcc.2021.287>
- Lu X, Kanghong D, Guo L, Wang P, Yildizbasi A (2020) Optimal estimation of the Proton Exchange Membrane Fuel Cell model parameters based on extended version of Crow Search Algorithm. *J Clean Prod*. <https://doi.org/10.1016/j.jclepro.2020.122640>
- Luo Y, Zhou L, qing, Yang, F., Chen, J. cai, Chen, J. jun, & Wang, Y. jun. (2023) Construction and analysis of a conjunctive diagnostic model of HNCCC with random forest and artificial neural network. *Sci Rep*. <https://doi.org/10.1038/s41598-023-32620-6>
- Lyu Z, Yu Y, Samali B, Rashidi M, Mohammadi M, Nguyen TN, Nguyen A (2022) Back-propagation neural network optimized by k-fold cross-validation for prediction of torsional strength of reinforced concrete beam. *Materials*. <https://doi.org/10.3390/ma15041477>
- Masrur Ahmed AA, Deo RC, Feng Q, Ghahramani A, Raj N, Yin Z, Yang L (2021) Deep learning hybrid model with Boruta-Random forest optimiser algorithm for streamflow forecasting with climate mode indices, rainfall, and periodicity. *J Hydrol*. <https://doi.org/10.1016/j.jhydrol.2021.126350>
- Mohamadi S, Sheikh Khozani Z, Ehteram M, Ahmed AN, El-Shafie A (2022) Rainfall prediction using multiple inclusive models and large climate indices. *Environ Sci Pollut Res*. <https://doi.org/10.1007/s11356-022-21727-4>
- Mohar SS, Goyal S, Kaur R (2023) Exploration of different topologies for optimal sensor nodes deployment in wireless sensor networks using jaya-sine cosine optimization algorithm. *J Supercomp*. <https://doi.org/10.1007/s11227-023-05147-w>
- Nayak J, Swapnarekha H, Naik B, Dhiman G, Vimal S (2023) 25 Years of particle swarm optimization: flourishing voyage of two decades. In *Arch Comput Methods Eng*. <https://doi.org/10.1007/s11831-022-09849-x>
- Pan H, Gong J (2023) Application of Particle Swarm Optimization (PSO) algorithm in determining thermodynamics of solid combustibles. *Energies* 16(14):5302
- Prasad R, Deo RC, Li Y, Maraseni T (2019) Weekly soil moisture forecasting with multivariate sequential, ensemble empirical mode decomposition and Boruta-random forest hybridizer algorithm approach. *CATENA*. <https://doi.org/10.1016/j.catena.2019.02.012>

29. Qiao X, Peng T, Sun N, Zhang C, Liu Q, Zhang Y, Wang Y, Shahzad Nazir M (2023) Metaheuristic evolutionary deep learning model based on temporal convolutional network, improved aquila optimizer and random forest for rainfall-runoff simulation and multi-step runoff prediction. *Expert Syst Appl*. <https://doi.org/10.1016/j.eswa.2023.120616>
30. Rahman AU, Abbas S, Gollapalli M, Ahmed R, Aftab S, Ahmad M, Khan MA, Mosavi A (2022) Rainfall prediction system using machine learning fusion for smart cities. *Sensors*. <https://doi.org/10.3390/s22093504>
31. Sayed GI, Hassanien AE, Azar AT (2019) Feature selection via a novel chaotic crow search algorithm. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-017-2988-6>
32. Singh AK, Kumar P, Ali R, Al-Ansari N, Vishwakarma DK, Kushwaha KS, Panda KC, Sagar A, Mirzania E, Elbeltagi A, Kuriqi A, Heddad S (2022) An integrated statistical-machine learning approach for runoff prediction. *Sustainability (Switzerland)*. <https://doi.org/10.3390/su14138209>
33. Shijun C, Qin W, Yanmei Z, Guangwen M, Xiaoyan H, Liang W (2020) Medium- and long-term runoff forecasting based on a random forest regression model. *Water Sci Technol*. <https://doi.org/10.2166/ws.2020.214>
34. Sulaiman SO, Shiri J, Shiralizadeh H, Kisi O, Yaseen ZM (2018) Precipitation pattern modeling using cross-station perception: regional investigation. *Environ Earth Sci*. <https://doi.org/10.1007/s12665-018-7898-0>
35. Vergni L, Todisco F (2023) A random forest machine learning approach for the identification and quantification of erosive events. *Water*. <https://doi.org/10.3390/w15122225>
36. Vu HL, Ng KTW, Richter A, An C (2022) Analysis of input set characteristics and variances on k-fold cross validation for a Recurrent Neural Network model on waste disposal rate estimation. *J Environ Manage*. <https://doi.org/10.1016/j.jenvman.2022.114869>
37. Wang JJ, Wang C, Fan JS, Mo YL (2022) A deep learning framework for constitutive modeling based on temporal convolutional network. *J Comput Phys*. <https://doi.org/10.1016/j.jcp.2021.110784>
38. Wang Z, Tian J, Fang H, Chen L, Qin J (2022) LightLog: A lightweight temporal convolutional network for log anomaly detection on the edge. *Comput Netw*. <https://doi.org/10.1016/j.comnet.2021.108616>
39. Yu PS, Yang TC, Chen SY, Kuo CM, Tseng HW (2017) Comparison of random forests and support vector machine for real-time radar-derived rainfall forecasting. *J Hydrol*. <https://doi.org/10.1016/j.jhydrol.2017.06.020>
40. Zeng L, Li M, Shi J, Wang S (2023) Spiral aquila optimizer based on dynamic gaussian mutation: applications in global optimization and engineering. *Neural Process Lett* 1:1–47
41. Zhang X, Zhao D, Wang T, Wu X, Duan B (2022) A novel rainfall prediction model based on CEEMDAN-PSO-ELM coupled model. *Water Supply*. <https://doi.org/10.2166/ws.2022.115>
42. Zhao Z, Xiao N, Shen M, Li J (2022) Comparison between optimized MaxEnt and random forest modeling in predicting potential distribution: a case study with *Quasipaa boulengeri* in China. *Sci Total Environ* 842:156867
43. Zhou D, Wang B (2022) Battery health prognosis using improved temporal convolutional network modeling. *J Energy Stor*. <https://doi.org/10.1016/j.est.2022.104480>

## Publisher's Note

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

Submit your manuscript to a SpringerOpen<sup>®</sup> 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)

---