Research Article

Differential evolution based radial basis function neural network model for reference evapotranspiration estimation

Babita Majhi[1] · Diwakar Naidu[1,2]

Abstract

The present study is an effort to examine the capability of a differential evolution based radial basis function neural network (RBFDE) to model weekly reference evapotranspiration (ET_0) as a function of climatic parameters in different agro-climatic zones (ACZs) of a moist sub-humid region in East-Central India. The ET_0 computed using the empirical equation of Penman–Monteith suggested by the Food and Agricultural Organization (FAO56-PM) is considered as a target variable for investigation. The performance of the proposed RBFDE model is compared with particle swarm optimization based radial basis function (RBFPSO), radial basis function neural network (RBFNN), multilayer artificial neural network (MLANN) models and conventional empirical equations of Hargreaves, Turc, Open-Pan, and Blaney-Criddle. Weekly ET_0 estimates that are obtained using RBFDE, RBFPSO, and RBFNN and MLANN are observed to be more consistent than equivalent empirical methods. For a critical analysis of simulation results, mean absolute percentage error (MAPE), root means square error (RMSE), determination coefficient (R^2) and Nash–Sutcliffe efficiency factor (NSE) is computed. Low MAPE and RMSE values along with higher R^2 and NSE close to 1, obtained with soft computing models exhibit that, soft computing models produce better estimates of ET_0 than empirical methods. Among the soft computing models, RBFDE provides improved results as compared to RBFPSO, RBFNN, and MLANN models. This method can be extended for ET_0 estimation in other ACZs.

Keywords ET_0 estimation · FAO56-PM · RBFNN · RBFDE · RBFPSO · MLANN · Empirical methods

1 Introduction

In response to atmospheric demand, soil surface evaporation and transpiration from plant occurs simultaneously in a cropping field and is termed as evapotranspiration (ET) in a combined manner [1]. Approximately two-thirds of the total precipitation is consumed by the atmosphere in the form of ET [2]. Therefore, ET is considered one of the most important water balance components for the determination of crop water requirement, length of the crop growing season, and associated agro-climatic studies. Hence, accurate measurement or estimation of ET is essential for the planning and effective implementation of irrigation and water management practices for practical applications. Accurate measurement of ET by volumetric and gravimetric lysimeter is practically very difficult because various factors affect the ET process, which includes climatic parameters, crop characteristics, soil properties, and management practices. Therefore, consumptive use of water from a uniformly distributed grass reference crop under nonlimiting conditions is estimated for practical purposes and termed as ET_0 [3]. In general, ET_0 is computed employing empirical equations as climatic parameters being the only factor affecting the ET process.

✉ Diwakar Naidu, dnaidu1971@gmail.com | ¹Department of CSIT, Guru Ghasidas Vishwavidyalaya, Central University, Bilaspur 495009, India. ²BRSM College of Agricultural Engineering and Technology and Research Station, Indira Gandhi Krishi Vishwavidyalaya, Mungeli, India.

Solar radiation, temperature, humidity, wind speed, and sunshine are the most influential climatic factors which contribute to the ET process [1]. Precise estimation of ET_0 is vital for the efficient utilization of available water resources for agricultural purposes.

Several physicals, empirical equations based on radiation, temperature, mass transfer, and water budget methods have been derived in the past to determine ET_0 with different input combinations of meteorological parameters. Among these empirical methods, the Penman–Monteith equation is recommended by the Food and Agricultural Organization for ET_0 estimation (FAO56-PM) because of its preciseness [1]. FAO56-PM equation requires meteorological parameters such as maximum and minimum temperatures, relative humidity, sunshine hours, wind speed, and solar radiation to determine ET_0. In developing countries, like India, it is invariably very difficult to obtain long term meteorological parameters to compute ET_0 using the FAO56-PM model [4]. Therefore, other empirical models like Hargreaves [5], Turc [6], Open Pan [7], Blaney-Criddle [7], and Christianson [8], etc., are also in use. These empirical equations involve fewer complex variables as compared to FAO56-PM to compute ET_0. However, ET_0 estimates obtained using these models are not comparable with FAO56-PM as these methods yield more errors and hence their practical applications become limited [9].

To address this issue, in recent decades, researchers have successfully demonstrated the application of a variety of computational intelligence based conventional and hybrid soft computing techniques for modeling extremely complex and non-linear relationship between climatic factors and ET_0 [10–19]. Improved predictions of FAO56-PM ET_0 are obtained by Wen et al. [20] using a support vector machine (SVM) as compared to the artificial neural network (ANN) and empirical methods in extremely arid regions of China. Partal [21] has developed a hybrid model combining wavelet transformation and radial basis function neural network (W-RBF) that outperformed conventional RBF, wavelet-multi-linear regression (W-MLR) and empirical methods of Hargreaves and Turc for daily ET_0 estimation with improved accuracy. Kisi and Demir [22] have evaluated the potential of multi-layer perceptron (MLP) with six different weight update algorithms for modeling ET_0 and found MLP with the Levenberg-Marquard algorithm produced a better estimate of ET_0. In a recent study, Dou and Yang [23] have recommended hybrid extreme learning machine (ELM) and adaptive neuro-fuzzy inference system (ANFIS) based models that are more robust and flexible in comparison to traditional ELM and ANFIS. Adamala [24] has reported improved generalized performance of wavelet neural network (WNN) and ANN model for estimation of ET_0 as compared to linear regression (LR),

wavelet regression (WR) and Hargreaves (HG) methods for the studied locations in different agro-ecological regions of India. Sanikhani et al. [25] have applied several artificial intelligence models including multi-layer perceptron (MLP), generalized regression neural network (GRNN), integrated ANFIS systems with grid partitioning (ANFIS-GP) and subtractive clustering (AFNIS-SC), radial basis neural network (RBNN) and GEP for modeling ET_0 in a cross-station scenario for different locations in Turkey and demonstrated that AI-based models performed better than the empirical equation of Hargreaves-Samani (HS) and its calibrated version (CHS).

It is also observed from the literature review that researchers have successfully implemented various types of hybrid soft computing models combining conventional neural networks along with evolutionary computing algorithms for estimation of ET_0. Application of nature-inspired algorithms such as genetic algorithm (GA), particle swarm optimization (PSO), artificial bee colony (ABC), etc., in combination with conventional neural networks like ANN and RBNN are investigated in some research publications for ET_0 estimation [26–30]. A study conducted by Feng et al. [31] for estimating FAO56-PM ET_0 in a humid region of Southwest China reveals that ELM and ANN optimized by genetic algorithm (GANN) has resulted in better ET_0 estimates than WNN and empirical approaches of Hargreaves, Makkink, Priestley–Taylor and Ritchie models. Gocić et al. [32] have analyzed the potential of genetic programming (GP), support-vector machine-firefly algorithm (SVM-FFA), ANN, and SVM-Wavelet soft computing approaches and found SVM-Wavelet resulted in improved FAO56-PM ET_0 estimates in Serbia. Mehdizadeh et al. [33] have evaluated the performance of gene expression programming (GEP) and MARS along with two SVM based hybrid models, SVM-Polynomial and SVM-RBF for estimation of monthly mean ET_0 and reported SVM-RBF and MARS outperformed GEP and SVM-Poly and also performed better than 16 other empirical equations considered for comparison. However, Mattar and Alazba [34] have confirmed that the GEP model performed better than the conventional multilinear regression (MLR) approach in Egypt. Most of the soft computing models discussed above are developed under a given scenario in terms of study location, the combination of available input climatic parameters, time scale and duration of climatic data, model structure, learning parameters, and an optimization algorithm, etc. Therefore, practically it becomes very difficult to employ these models in a new location without proper calibration and validation of the model parameters.

To examine the potential of an evolutionary optimized soft computing technique, RBFNN in combination with the differential evolution algorithm (RBFDE) is introduced here for the estimation of ET_0 under three different ACZs in the

Chhattisgarh region of East-Central India. Differential evolution (DE) is considered because it is a simple algorithm in comparison to GA which requires intensive calculations. Due to its simplicity, DE is used in various applications [35–37]. Technical analysis of DE parameters, hybridization of DE with other soft computing techniques, and its practical applications have been discussed by Das et al. [38]. Different variants over state-of-the-art DE have also been presented in the literature. Among these, Hui and Suguntham [39] suggested ensemble and arithmetic recombination-based speciation DE for multimodal optimization of common benchmark problems. Ramdas et al. [40] developed a reconstructed mutation strategy for DE and applied the same with multilevel image thresholding for improved weather radar image segmentation [41]. A DE variant with multi-donor mutation strategy and annealing-base local search has been developed by Ghosh et al. [42] for optimization of Lennard-Jones potential function-based molecular clustering. The effect of DE-based constraint handling techniques has been evaluated by Biswas et al. [43] for the optimization of power flow systems. One of the authors of this investigation has also been engaged in DE based training of adaptive autoregressive moving average (ARMA) model for exchange rate forecasting [44] and development of a hybrid system using functional link artificial neural network (FLANN) and DE for Odia handwritten numeral recognition [45]. The proposed evolutionary optimized hybrid structure of RBFDE is developed and used for the first time to model FAO56-PM ET_0, and therefore it may be considered as a novel scientific approach for such application. Conventional soft computing techniques like MLANN, RBFNN along with empirical methods of Hargreaves, Turc, Open Pan, and Blaney-Criddle are considered for comparison purposes. Results obtained with RBFDE is also compared with RBFPSO under similar condition. This paper is organized into different sections. Section 1 introduces the problem formulation, literature reviews, and motivation behind the investigation. The detailed description of the data sets, soft computing techniques, and empirical methods are described in the Materials and methods of Sect. 2. Simulation results and comparative performance evaluation of different models are outlined in the results and discussion of Sect. 3. The salient findings of the study are summarized in the conclusion section.

2 Materials and methods

2.1 Study area and dataset

This investigation is carried out to model weekly ET_0 using soft computing techniques. Long term weekly meteorological data (2001 to 2019) of maximum temperature (T_{max}), minimum temperature (T_{min}), bright sunshine hours (BSS), wind speed (WS), morning relative humidity during (RH_1), afternoon relative humidity (RH_2) and weekly cumulative pan evaporation (EP) are collected from Raipur, Jagdalpur and Ambikapur stations located in three distinct ACZs of Chhattisgarh region in central India (Fig. 1). The climate of Chhattisgarh is moist sub-humid in general with an average annual rainfall of 1200–1400 mm and annual ET_0 losses between 1400 and 1600 mm in different ACZs. Data sets are collected from the India Meteorological Department (IMD) (https://mausam.imd.gov.in/) certified observatories located in these stations. These surface meteorological observatories follow the World Meteorological Organization (WMO) guidelines for data collection [46]. WMO guidelines for the observational procedure and quality control are adopted uniformly in these surface meteorological observatories while data acquisition, tabulation, and computation. The online data entry system, itself has an inbuilt quality control mechanism to test the errors like data format, duplicate records, and incorrect units of measurement, impossible values, extremes, and outliers.

Descriptive statistics of different meteorological parameters in terms of mean, high, low, range, standard deviation (SD), and coefficient of variation (CV) are also computed to understand data patterns and to ensure the quality check of data (Table 1). To measure the strength and direction of a linear relationship between two variables, correlation coefficient (R) between meteorological parameters (T_{max}, T_{min}, BSS, WS, RH_1, RH_2, and EP) with FAO56-PM ET_0 are also computed (Table 1). Weekly totals of ET_0 are computed using the FAO56-PM equation which is considered as the target output for model development [1].

The pattern of different meteorological parameters considered as input variables for model development along with target variable FAO56-PM ET_0 in selected stations is represented as box plot arrangements in Fig. 2. The middle line of the box plot signifies the median value while the upper and lower edges signify 75% and 25% of the data set respectively. The highest and lowest limits of the upper and lower vertical lines indicate the highest and lowest values respectively. The square depicts the simulated mean, and the straight-line shows the observed mean.

2.2 Design of soft computing models

2.2.1 Radial basis function neural network (RBFNN) based estimator

RBFNN is a category of feed-forward neural network with a single hidden layer and an output layer formulated by Broomhead and Lowe [47]. Pictorial representation of the

Fig. 1 Location map of the study area
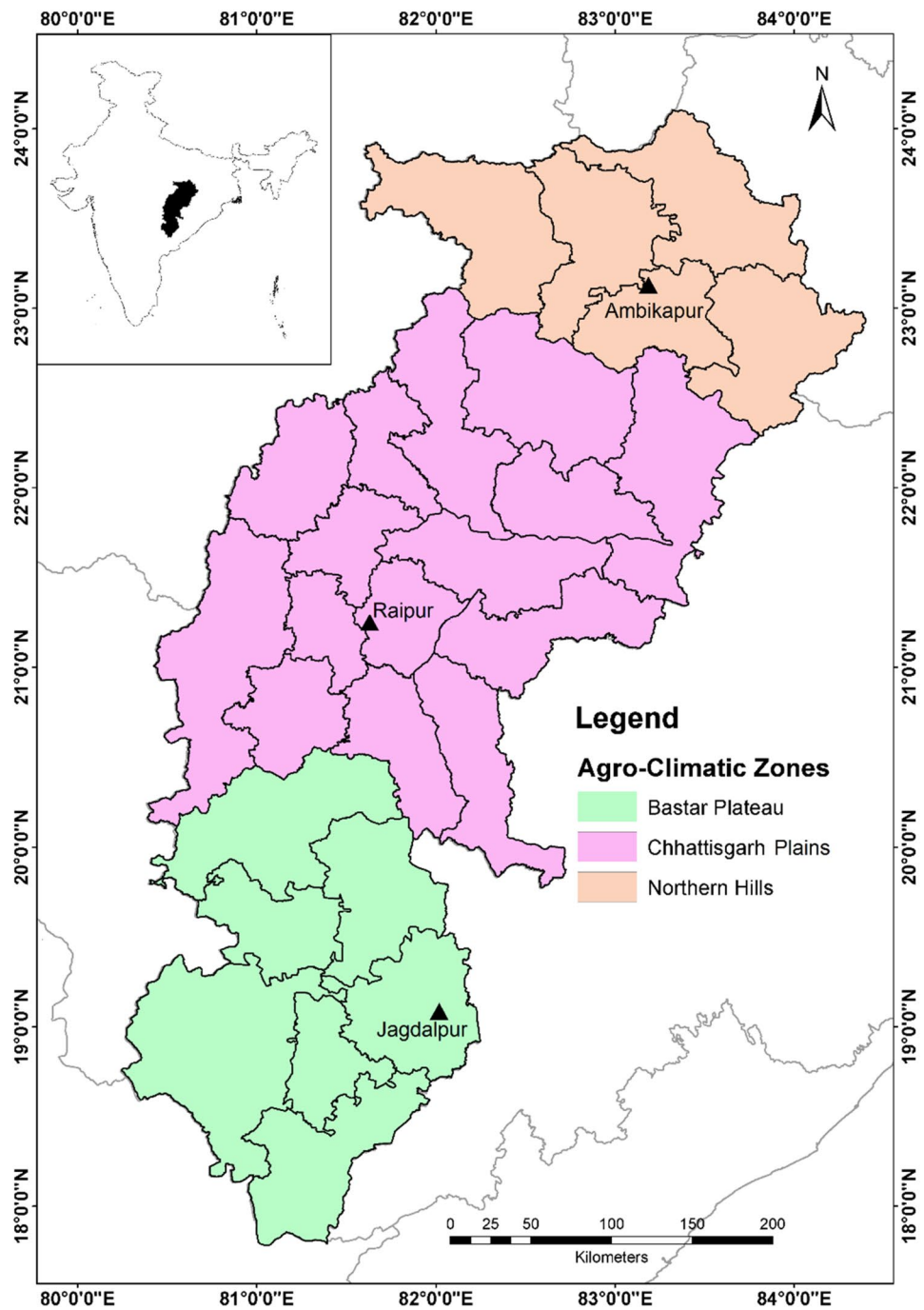
RBFNN is given in Fig. 3. The processing units termed as neurons in the hidden layer are associated with centers, $c = c_1, c_2, c_3, ., ., ., c_h,$ and their width $\sigma = \sigma_1, \sigma_2, \sigma_3, ., ., .\sigma_h,$ where h is the number of neurons in the hidden layer. Each neuron in the hidden layer receives the same set of input data $(X = x_1, x_2, x_3, ., ., ., x_n)$. The centers of every hidden neuron have the same dimension as that of the input data, i.e. $c_i \in R^n, X \in R^n$. The output of hidden layer neurons $(\varphi_1, \varphi_2, \varphi_3, ., ., ., \varphi_h)$ are associated with synaptic weights

$(w_1, w_2, w_3, ., ., ., w_h)$. Output, $\emptyset_i$ of ith hidden layer neuron is basically a Gaussian function and is represented by:

$$\emptyset_i(z) = e^{\frac{-z^2}{2\sigma_i^2}} \tag{1}$$

where $z = ||x - c_i||$, represents the Euclidian distance between input data and the corresponding centers and $\emptyset_i = \emptyset(||X - c_i||)$. The Gaussian function used in each hidden layer neuron is a category of radial basis function.

Table 1 Descriptive statistics of weekly meteorological parameters (2001–2019) at different locations

Parameters	Mean	High	Low	Range	SD	CV	R
Raipur (21.14°N, 81.38°E, 289 m)							
T_{max} (°C)	32.9	46.0	22.0	24.0	4.9	14.8	0.95
T_{min} (°C)	20.4	31.5	6.6	24.9	5.8	28.6	0.62
BSS (hours)	6.7	11.1	0.0	11.1	2.6	39.5	0.33
WS (Kmph)	4.9	14.9	0.5	14.4	3.1	62.7	0.44
RH_1 (%)	79.5	96.4	25.9	70.5	16.5	20.7	−0.87
RH_2 (%)	43.8	91.0	6.7	84.3	22.2	50.6	−0.47
EP (mm week^{-1})	38.0	127.2	10.5	116.7	22.0	58.0	0.97
FAO56-PM ET_0 (mm week^{-1})	29.7	68.7	11.0	57.7	12.2	41.1	1.00
Jagdalpur (19.08°N, 82.01°E, 564 m)							
T_{max} (°C)	31.0	42.6	23.3	19.3	4.0	12.8	0.93
T_{min} (°C)	18.1	28.1	4.3	23.8	5.7	31.2	0.47
BSS (hours)	6.2	11.0	0.0	11.0	2.8	44.9	0.43
WS (Kmph)	4.5	11.6	1.0	10.6	2.0	44.3	0.36
RH_1 (%)	86.9	97.7	34.0	63.7	9.4	10.8	−0.83
RH_2 (%)	51.1	95.7	7.4	88.3	20.5	40.1	−0.56
EP (mm week^{-1})	29.1	93.4	3.8	89.6	14.6	50.0	0.91
FAO56-PM ET_0 (mm week^{-1})	26.8	61.2	13.0	48.2	8.9	33.3	1.00
Ambikapur (23.12°N, 83.20°E, 604 m)							
T_{max} (°C)	30.4	43.6	18.9	24.7	5.12	16.9	0.931
T_{min} (°C)	17.8	28.9	3.1	25.8	6.32	35.5	0.647
BSS (hours)	7.2	11.1	0.2	10.9	2.59	36.1	0.351
WS (Kmph)	3.5	10.4	0.3	10.1	1.79	50.6	0.616
RH_1 (%)	78.8	98.9	25.7	73.2	17.12	21.7	−0.837
RH_2 (%)	46.6	91.6	9.6	82.0	21.62	46.4	−0.452
EP_0 (mm week^{-1})	31.7	92.5	8.8	83.7	16.19	51.0	0.909
FAO56-PM ET_0 (mm week^{-1})	26.7	61.8	11.5	50.4	10.52	39.4	1.000

Finally, the response of the RBFNN at the output layer, for a given set of input data is linear in terms of weights and computed using the following expression.

$$y = \sum_{i=1}^{h} w_i \emptyset_i \qquad (2)$$

Development of the RBFNN for each instant of input data and its corresponding output $\{X, y\}$ is obtained recursively by updating the network parameters $\{w_i, c_i, \sigma_i\}$ to minimize the instantaneous error cost function given as.

$$e = \frac{1}{2}\left(y^d - y\right)^2 \qquad (3)$$

The weight update rules to optimize the network parameters $\{w_i, c_i, \sigma_i\}$ at time t are given by following equations which are derived using gradient descent algorithm [48].

$$w_i(t+1) = w_i(t) + \eta_1\left(y^d - y\right)\emptyset_i \qquad (4)$$

$$c_{ij}(t+1) = c_{ij}(t) + \frac{\eta_2}{\sigma_i^2}\left(y^d - y\right)w_i\emptyset_i\left(x_j - c_{ij}\right) \qquad (5)$$

$$\sigma_i(t+1) = \sigma_i(t) + \frac{\eta_3}{\sigma_i^3}\left(y^d - y\right)w_i\emptyset_i z_i^2 \qquad (6)$$

where y^d desired output or target value, c_{ij} j^{th} element of the ith center, η_1, η_2, η_3 learning rates for network parameters $\{w_i, c_i, \sigma_i\}$ respectively.

2.2.2 Differential evolution based RBF neural network estimator

Differential evolution (DE) [49, 50] is a simple and efficient global optimization technique based on a heuristic method for minimizing a nonlinear function. Using this efficient heuristic approach a hybrid structure, RBFDE is developed in which total *d* number of network parameters, represented by a parameter vector, $\vec{x}_i = \{w_i, c_i, \sigma_i\}$, is optimized by the differential evolution algorithm (DE).
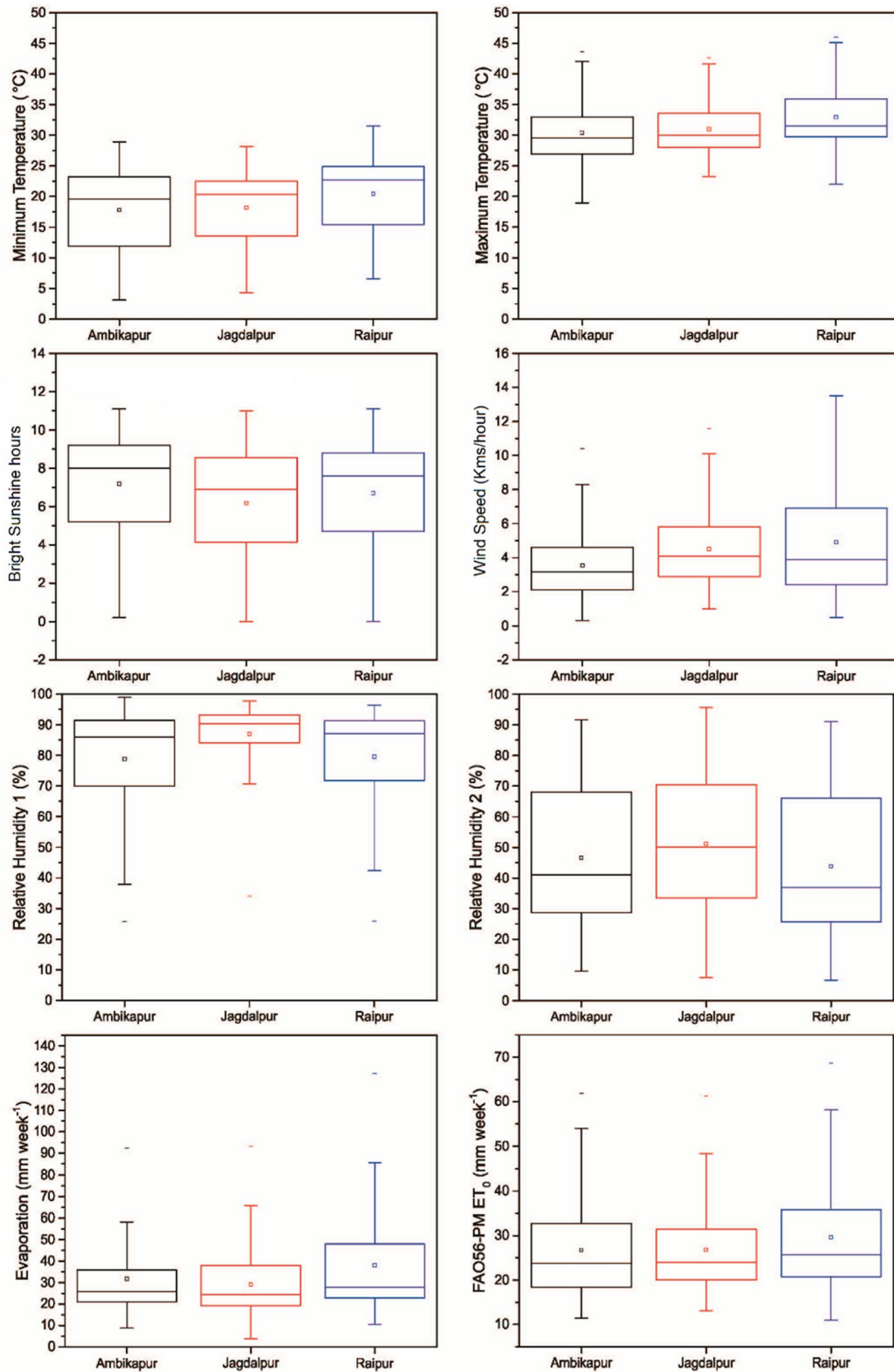
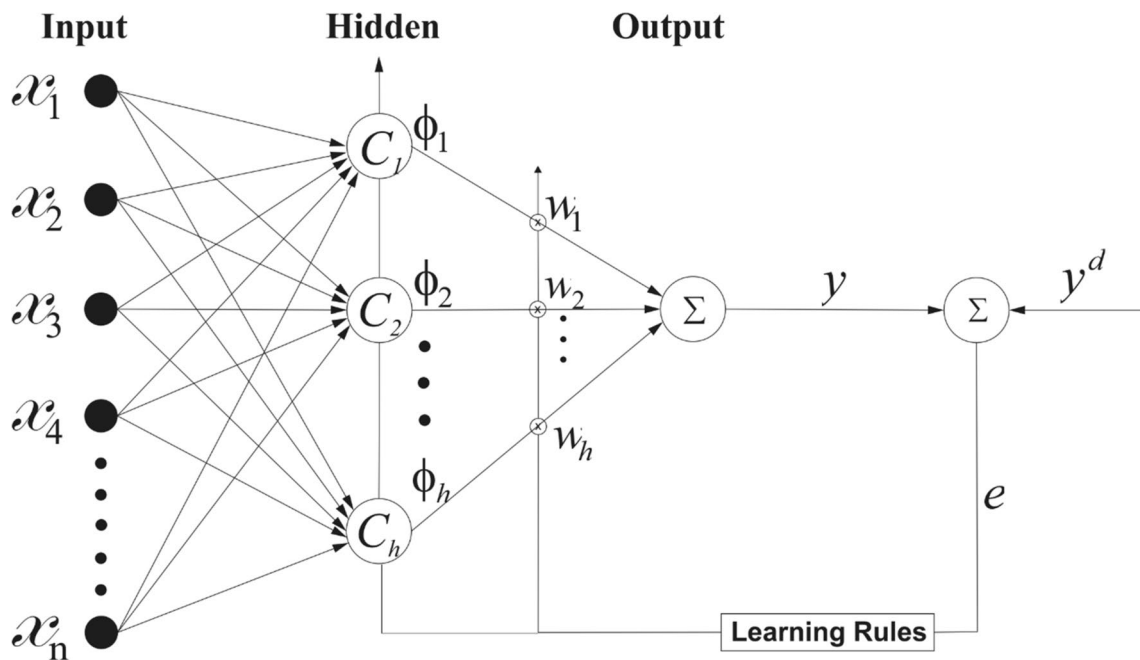**Fig. 2** Box plot of input meteorological parameters and FAO56-PM ET0 in different stations

Fig. 3 Block diagram of RBFNN based estimator

DE algorithm involves three basic operations viz., mutation, recombination, and selection. The step-wise procedure for the development of RBFDE is described below.

Step 1 Randomly initialize $i = 1, 2, 3, ., ., .NP$ number of target or population vectors, $\vec{x}_{i,G}$ between 0 to 1, where each ith individual of the population vector represents parameters of the RBFDE model. The ith target vector of Gth generation, $\vec{x}_{i,G}$ is given as $\vec{x}_{i,G} = \{w_i, c_i, \sigma_i\}$.

Step 2 Repeat step 3 with each target vector $\vec{x}_{i,G}$ for $i = 1, 2, 3, ., ., .NP$.

Step 3 a) Give K numbers of input patterns to the RBF network sequentially with each pattern having dimension n.

b) For each one of the K input patterns, obtain corresponding network output using ith target vector $\vec{x}_{i,G}$ as the parameters of the network and compare it with the corresponding desired output to get an error using (3). For K patterns, the K number of error values will be obtained.

c) Calculate $f(\vec{x}_{i,G})$ using (7), where $f(\vec{x}_{i,G})$ represents the fitness function i.e. mean square error (MSE).

$$f(\vec{x}_{i,G}) = \frac{\sum_{j=1}^{K} e^2}{K} \qquad (7)$$

Step 4 Obtain $f_{min}(\vec{x}_{i,G})$ and represent the corresponding $\vec{x}_{i,G}$ as the $\vec{x}_{best,G}$ for Gth generation.

Step 5 Choose a scaling factor $F \in [o, 1]$ and a cross over ratio $CR, \in [o, 1]$ and repeat step 6 to step 15 until the desired minimum MSE is obtained.

Step 6 Repeat the steps from 7 to 8 for $i = 1, 2, 3, ., ., .NP$ times

Step 7 Randomly choose two indices r_1, r_2 from 1 to NP, such that, $r_1 \neq r_2 \neq i$.

Step 8 Compute the mutant vector $v_{i,G}$ for each target vector $\vec{x}_{i,G}$ for Gth generation as

$$\vec{v}_{i,G} = \vec{x}_{i,G} + F * \left(\vec{x}_{r_{1,G}} - \vec{x}_{r_{2},G}\right) + F * \left(\vec{x}_{best,G} - \vec{x}_{i,G}\right). \qquad (8)$$

Step 9 Repeat the steps from 10 to 11 for $i = 1, 2, 3, ., ., .NP$ times

Step 10 Randomly choose an index r_3 between 1 to d and repeat step 11 for $j = 1$ to d, where d is the dimension of the target or population vector.

Step 11 Generate a random number $rand \in [o, 1]$ and compute the trial vector $\vec{U}_{j,i,G}$ by recombination operation, which replaces the previously successful individuals with mutant vector as

$$\vec{U}_{j,i,G} = \begin{cases} \vec{v}_{j,i,G} & if (rand \leq CR) \; or \; j = r_3 \\ else & \\ \vec{x}_{j,i,G} & \end{cases}. \qquad (9)$$

Step 12 For each trial vector $\vec{U}_{j,i,G},i = 1, 2, 3, ., ., .NP$, evaluate $f(U_{i,G})$, which is a mean square error (MSE). (Similar to step 3)

Step 13 Repeat the step14 for $i = 1, 2, 3, ., ., .NP$

Step 14 Finally, the next generation of NP number of target/population vector $\vec{x}_{i,G+1}$ is selected based on survival of the fittest criteria as

$$\vec{x}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & if\ f\left(\vec{U}_{i,G}\right) \le f\left(\vec{x}_{i,G}\right) \\ else \\ \vec{x}_{i,G} \end{cases}. \tag{10}$$

Step 15 Obtain $f_{min}\left(\vec{x}_{i,G+1}\right)$ and represent it as $\vec{x}_{best,G+1}$ for the next generation.

Step 16 Stop

Pictorial representation of the DE algorithm is shown in Fig. 4.

2.2.3 Particle swarm optimization based RBF neural network estimator

In this approach parameters of the RBFNN model i.e. $\{w_i, c_i, \sigma_j\}$, as described in Sect. 2.2.1, are updated using the PSO algorithm. The PSO [51–53] is a metaheuristics optimization algorithm inspired by the paradigm of swarm intelligence which mimics the social behavior of animals like fish and birds. It is successfully applied to various applications in engineering and science [54–56]. The algorithm uses a fixed number of particles that represent the parameters of RBFNN. Each particle updates its current velocity and position by its own experience called personal best (*p-best*) and by the social experience of the swarm called global best (*g-best*). Steps involved in PSO are briefly described below:

Step 1 Initialize fix number of particles with random position and velocity uniformly distributed over the search space.

Step 2 Evaluate the fitness of each particle according to the objective function

Step 3 Record *pbest* for each particle and *g-best* of the swarm.

Step 4 Update velocity of each particle

Step 5 Update the position of each particle.

Step 6 Update *pbest* and *gbest*

Step 7 Repeat the steps from 2 to 6 until the termination condition is satisfied and stop.

Pictorial representation of the PSO algorithm is shown in Fig. 5.

2.2.4 Multi-layer artificial neural network (MLANN)

MLANN, suggested by Haykin [57] is successfully employed in many applications to solve the regression problem. MLANN architecture considered for this proposed investigation consists of an N-5-1 structure. N represents the number of input features. Optimum results are obtained with 5 neurons in the intermittent hidden layer. Desired ET_0 estimates are obtained at output neurons. Hyperbolic tangent (tanh) is used as an activation function in every processing neuron. The training of the network is done by a conventional back-propagation algorithm which is based on the error-correcting learning rule to update the weights and bias of each neuron in different layers.

2.3 Empirical models

Weekly ET_0 for the study locations is also computed using empirical methods of FAO56-PM, Blaney-Criddle, Open Pan, Turc, and Hargreaves from available meteorological data. A brief description regarding empirical approaches considered in this investigation and the corresponding input meteorological parameter requirement are listed in Table 2. The description regarding different climate based empirical methods considered in this investigation is not included in this paper. More details regarding these empirical approaches can be obtained from basic references [1, 5–7].

2.4 Performance evaluation measures

Comparative analysis of estimated ET_0 obtained with different soft computing models and empirical methods considered for the investigation is carried out by computing performance evaluation measures, namely, mean square percentage error (MAPE), root mean square error (RMSE), determination coefficient (R^2) and efficiency factor (EF) proposed by Nash and Sutcliffe (NSE) [58]. The mathematical expression of different evaluation measures is as follows.

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \frac{\left|(Out_{obs} - Out_{est})\right|}{Out_{obs}} \times 100 \tag{11}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (Out_{est} - Out_{obs})^2} \tag{12}$$
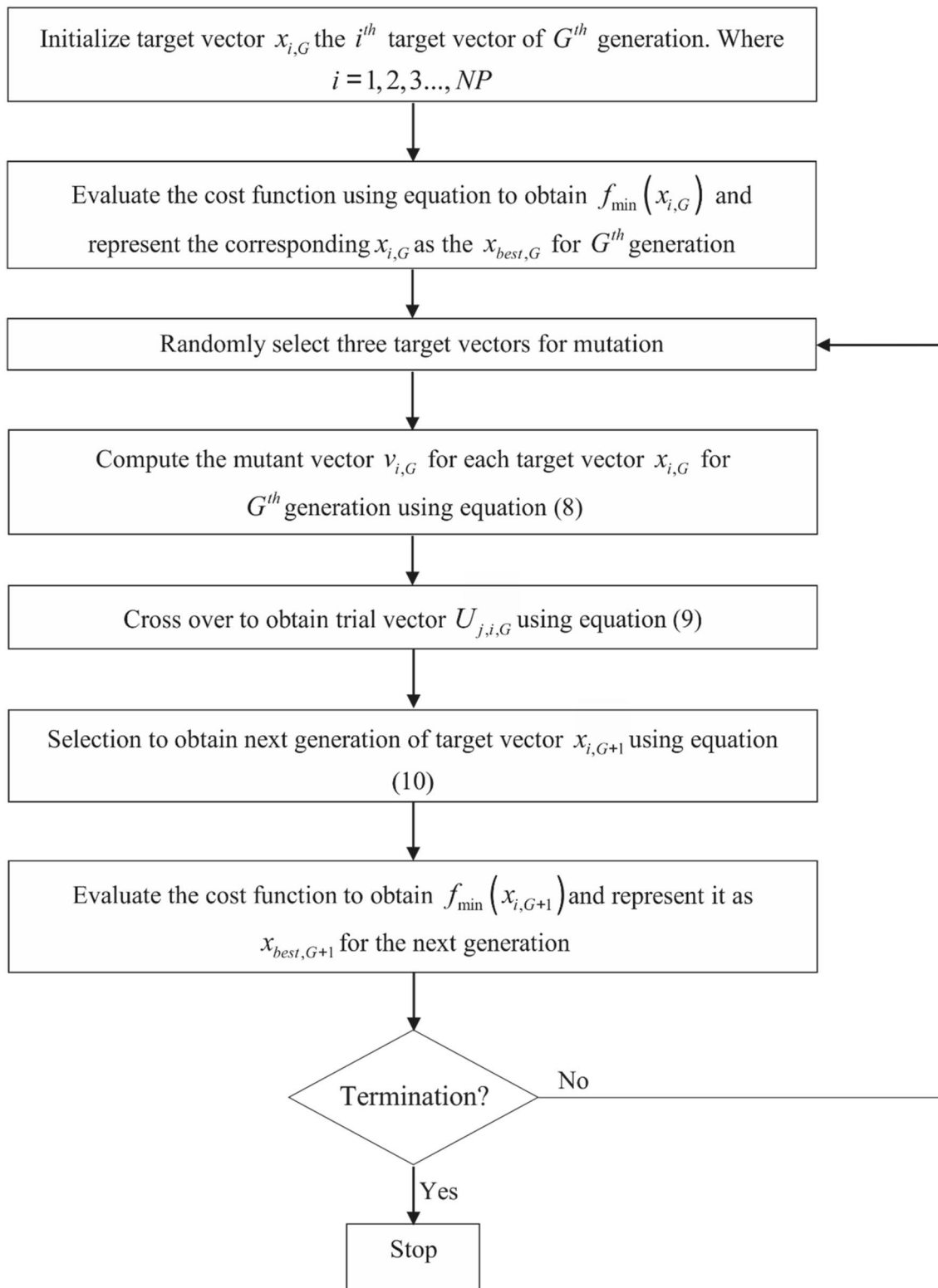
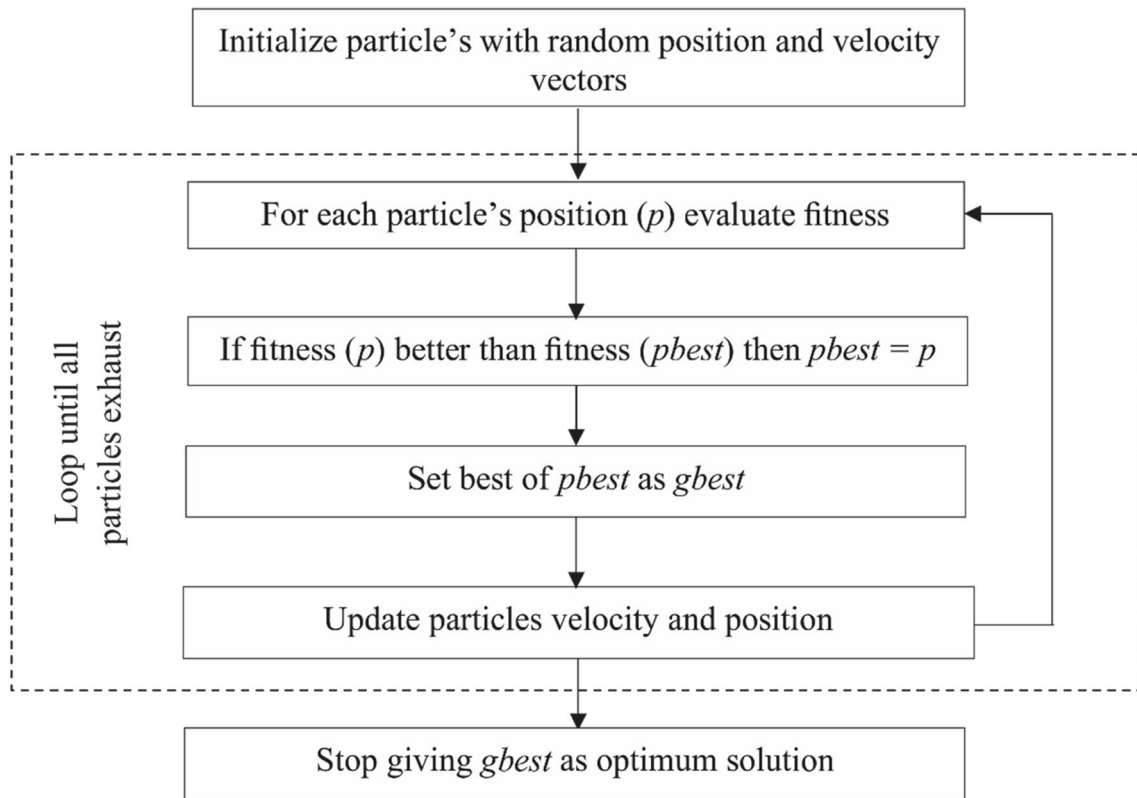**Fig. 4** Flowchart differential evolution algorithm

Fig. 5 Flowchart particle swarm optimization algorithm

Table 2 Details of empirical models used to compute ET_0

Empirical methods	Input combination	Empirical equation
Hargreaves [5]	T_{max}, T_{min}	$ET_0 = 0.0023 \times R_a \times T_d \times 0.5 \times (T_m + 17.8)$
Turc [6]	T_{max}, T_{min}, BSS	$ET_0 = \frac{0.40 \times T_m (R_s + 50)}{(T+15)}$
Blaney-Criddle [7]	$T_{max}, T_{min}, RH_1 \& RH_2, WS, BSS$	$ET_0 = a + b \times p(0.46 \times T_m + 8.13)$
Open Pan [7]	$WS, EP, RH_1 \& RH_2$	$ET_0 = k_p \times EP$
FAO56-PM [1]	$T_{max}, T_{min}, RH_1 \& RH_2, WS, BSS$	$ET_0 = \frac{0.408\Delta(R_n - G) + \gamma \frac{900}{T+273} u_2 (e_s - e_a)}{\Delta + \gamma(1 + 0.34 u_2)}$

$$R^2 = \frac{\left(\sum_{i=1}^{n} \left(Out_{obs} - \overline{Out_{obs}} \right) \left(Out_{est} - \overline{Out_{est}} \right) \right)^2}{\sum_{i=1}^{n} \left(Out_{obs} - \overline{Out_{obs}} \right)^2 \sum_{i=1}^{n} \left(Out_{est} - \overline{Out_{est}} \right)^2}$$

(13)

$$EF = NSE = 1 - \frac{\sum_{i=1}^{n} \left(Out_{est} - Out_{obs} \right)^2}{\sum_{i=1}^{n} (Out_{obs} - \overline{Out_{obs}})^2} (-\infty \leq EF \leq 1)$$

(14)

where Out_{obs} denotes the target and Out_{est} denotes model estimated ET_0 values. n is the number of testing patterns. Low MAPE and RMSE values represent the close agreement between desired and estimated output. Similarly,

R^2 and EF values close to 1 are also indicators of a higher accuracy level of the model.

3 Results and discussion

The key objective of this investigation is to examine the potential of different evolutionary optimized hybrid (RBFDE, RBFPSO) and conventional (RBFNN, MLANN) soft computing approaches with available climatic features for estimation of ET_0 comparable to FAO56-PM ET_0. Input features combination of different models is decided based on empirical approaches of Hargreaves, Turc, Open Pan, Blaney-Criddle, and FAO56-PM ET_0 listed in the previous section. These soft computing models are categorized

into type I to type V models. Like the Hargreaves method, type I models include only T_{max} and T_{min} as input features, whereas Type II soft computing models include BSS with temperature, which is equivalent to the Turc approach. In type III soft computing models, EP, RH_1, RH_2, and WS are considered as input features similar to that of the Open Pan empirical approach. Type IV models include six weather parameters (T_{max}, T_{min}, BSS, WS, and RH_1 and RH_2) equivalent to the Blaney-Criddle empirical approach. Another category of soft computing model termed Type V models are developed using T_{max}, T_{min}, BSS, and WS since these weather parameters exhibit positive correlations with ET_0. The input feature EP, which is also positively correlated with ET_0, is not included in type V soft computing models as obtaining EP data is very difficult. Input feature combinations used in different types of soft computing models and their equivalent empirical models are shown in Table 3. Weekly meteorological data of Raipur, Jagdalpur, and Ambikapur from 2001 to 2015 (80%) are used for model calibration or training, whereas the recent 4 years (20%) of the weekly meteorological data from 2016 to 2019 are used for model validation.

Soft computing models RBFDE, RBFPSO, RBFNN, and MLANN are coded in MATLAB as per the design and the learning algorithm described earlier. Simulation studies are carried out with a different input features combination to test the sensitivity of the soft computing approach to control parameters until a satisfactory accuracy level is achieved for estimation of FAO56-ET_0 for different study locations. Detailed information regarding modeling strategies and respective control parameters that produce optimum results during the simulation process are shown in Table 4 for different soft computing models.

Calibration of RBFDE, RBFPSO, RBFNN, and MLANN models is done using the above-listed network parameters with training datasets of all the three study locations, Raipur, Jagdalpur, and Ambikapur. During the training process, input patterns are given to the model sequentially and the corresponding estimated output is obtained at the output layer after completion of the forward pass (Fig. 3). The estimated output is compared with the corresponding target FAO56-ET_0 output to compute the instantaneous error cost function. Real-time update of the model parameters is done in each instance to minimize the squared error using respective evolutionary (DE and PSO) and conventional back-propagation learning algorithms (RBFNN and MLANN). The process continues until all the available training input patterns for model calibration gets

Table 3 Input feature combinations used in soft computing and empirical models

Type	Soft computing models				Equivalent empirical model	Input feature combinations
	RBFDE	RBFPSO	RBFNN	MLANN		
I	RBFDE1	RBFPSO1	RBFNN1	MLANN1	Hargreaves	T_{max}, T_{min}
II	RBFDE2	RBFPSO2	RBFNN2	MLANN2	Turc	T_{max}, T_{min}, BSS
III	RBFDE3	RBFPSO3	RBFNN3	MLANN3	Open Pan	EP, RH_1 & RH_2, WS
IV	RBFDE4	RBFPSO4	RBFNN4	MLANN4	Blaney-Criddle	T_{max}, T_{min}, RH_1 & RH_2, WS, BSS
V	RBFDE5	RBFPSO5	RBFNN5	MLANN5	–	T_{max}, T_{min}, BSS, WS

Table 4 Parameters of the soft computing models used for simulation

Model parameters	RBFDE	RBFPSO	RBFNN	MLANN
Dimension of input features (D)	2, 3, 4 & 6	2,3,4 & 6	2, 3 4 & 6	2,3,4 & 6
Normalization of input features	0 to 1	0 to 1	0 to 1	− 1 to 1
Normalization method	$\frac{x_k - x_{min}}{x_{min} - x_{man}}$, where x_k represents kth instance			
Initial values of the model parameters	0 to 1	0 to 1	0 to 1	− 1 to 1
Number of centers for RBF based models (c)	10	10	10	–
Number of neurons in intermediate layer	–	–	–	5
Output neuron	1	1	1	1
Activation function	Gaussian	Gaussian	Gaussian	*tanh*
Learning algorithm	DE	PSO	RBF update rules	Back-propagation
Convergence coefficients/control parameters/	F = 0.9 CR = 0.9	Constriction coefficient $C_1 = C_2 = 1.49$ Inertia weight w Linearly decreased from 0.9 to 0.4	$\eta_1 = \eta_2 = \eta_3 = 0.01$	$\eta = 0.01$
Number of population/particles	$5 \times d$	50	–	–

exhausted. This completes one cycle called an epoch. At the end of each epoch, the mean square error is computed and stored for each epoch to examine the learning characteristic of soft computing models. The iterative process is repeated several times until MSE is minimized to a desired low value nearly close to zero. This completes the supervised learning process and model parameters are then fixed to constitute soft computing models. A similar calibration process is adopted for all soft computing approaches.

To test the performance of different soft computing models, test data patterns are then presented sequentially at the input layer of the model and through forward pass respective estimated ET_0 is obtained at the output layer for all the test patterns. These ET_0 estimates are then compared with corresponding target FAO56-PM ET_0 values. Performance evaluation measures, MAPE (%), RMSE (mm week^{-1}), R^2, and NSE as described in the previous section are then computed using desired and estimated output of different types of soft computing models and equivalent empirical approaches for comparison of model performance, which ultimately leads to model selection. The computed values of performance evaluation measures for different types of soft computing models and equivalent empirical approaches considered are listed in Tables 5 and 6 for all three locations. Comparative results of the analysis are discussed below:

i. For type I soft computing models, MAPE ranges from lowest of 7.4 for RBFDE1 and RBFDE2 (at Raipur) to highest of 11.8 for MLANN1 (at Jagdalpur), whereas MAPE obtained with Hargreaves model is comparatively very high and ranges between 22.6 (at Raipur) to 30.3 (at Ambikapur).

ii. Type II soft computing models produce improved ET_0 estimates with low MAPE compared to type I models. For type II models, MAPE ranges from the lowest of 4.9 with RBFDE2 (at Raipur) to a high of 10.2 with MLANN2 (at Raipur). MAPE is again quite higher with the equivalent empirical approach of Turc, which is obtained between 10.1 (at Jagdalpur) to 13.9 (at Raipur).

iii. Subsequently, for type III models, MAPE values are computed close to that of type II models, which varied between a lowest of 4.7 with RBFDE3 & RBFPSO3 (at Raipur) to a high of 8.0 with MLANN3 (at Jagdalpur). MAPE for the Open Pan approach varies between 12.2 (at Raipur) to 22.2 (at Ambikapur),

Table 5 MAPE (%) and RMSE (mm week^{-1}) for different types of soft computing and equivalent empirical models with test data sets of Ambikapur, Jagdalpur, and Raipur

Type	MODEL	Ambikapur		Jagdalpur		Raipur	
		MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
I	RBFDE1	**8.9***	**2.96**	**10.6**	**3.63**	**7.4**	**2.98**
	RBFPSO1	9.3	3.04	10.7	3.67	**7.4**	**2.98**
	RBFNN1	9.3	3.06	10.5	3.79	8.5	3.22
	MLANN1	10.0	3.11	11.8	3.88	9.0	3.23
	Hargreaves	30.3	7.15	29.0	7.53	22.6	6.13
II	RBFDE2	**4.9**	**2.10**	**5.3**	**2.74**	**5.7**	**2.63**
	RBFPSO2	5.0	**2.10**	5.4	2.78	5.9	2.68
	RBFNN2	4.9	2.23	5.2	2.81	5.9	3.14
	MLANN2	5.1	2.31	6.2	2.93	10.2	3.45
	Turc	13.9	3.73	10.1	4.26	13.9	6.54
III	RBFDE3	**5.9**	**1.82**	**7.0**	**2.18**	**4.7**	**1.75**
	RBFPSO3	6.0	1.89	6.9	2.21	4.7	1.77
	RBFNN3	6.3	1.95	7.0	2.27	4.9	1.80
	MLANN3	7.8	2.43	8.0	2.48	6.6	2.28
	Open-Pan	22.2	6.55	21.7	6.30	16.2	5.05
IV	RBFDE4	**2.1**	**0.68**	**3.7**	**1.06**	**1.1**	**0.36**
	RBFPSO4	3.4	1.02	3.8	1.10	1.3	0.43
	RBFNN4	4.4	1.32	4.1	1.22	2.4	0.82
	MLANN4	4.6	1.36	4.4	1.25	3.9	1.29
	Blaney-Criddle	22.1	6.12	15.5	4.27	22.6	7.20
V	RBFDE5	**2.2**	**0.66**	**3.4**	**1.06**	**1.9**	**0.80**
	RBFPSO5	3.9	1.08	3.6	1.08	2.2	0.87
	RBFNN5	4.6	1.38	4.7	1.39	2.9	1.07
	MLANN5	5.2	1.48	5.2	1.46	4.1	1.31

*boldface numbers highlight the best results

Table 6 R^2 and NSE for different types of soft computing and equivalent empirical models with test datasets of Ambikapur, Jagdalpur, and Raipur

Type	MODEL	Ambikapur		Jagdalpur		Raipur	
		R^2	NSE	R^2	NSE	R^2	NSE
I	RBFDE1	0.949	**0.892***	0.878	**0.785**	**0.955**	0.932
	RBFPSO1	**0.950**	0.886	**0.880**	0.781	0.954	**0.933**
	RBFNN1	**0.950**	0.884	0.876	0.766	0.948	0.922
	MLANN1	0.938	0.881	0.871	0.755	0.946	0.921
	Hargreaves	0.933	0.371	0.809	0.077	0.900	0.716
II	RBFDE2	**0.981**	**0.946**	**0.955**	**0.950**	**0.969**	**0.948**
	RBFPSO2	0.980	0.945	0.954	0.948	**0.969**	0.946
	RBFNN2	0.978	0.939	0.950	0.947	0.946	0.926
	MLANN2	0.976	0.934	0.948	0.939	0.929	0.910
	Turc	0.905	0.829	0.795	0.704	0.833	0.677
III	RBFDE3	**0.960**	**0.959**	**0.934**	**0.922**	**0.978**	**0.977**
	RBFPSO3	0.956	0.956	0.930	0.920	0.977	0.976
	RBFNN3	0.954	0.953	0.931	0.916	0.977	0.976
	MLANN3	0.928	0.927	0.908	0.900	0.961	0.961
	Open-Pan	0.743	0.472	0.827	0.353	0.947	0.808
IV	RBFDE4	**0.995**	**0.994**	**0.988**	**0.982**	**0.999**	**0.999**
	RBFPSO4	0.989	0.987	0.986	0.980	0.999	0.999
	RBFNN4	0.982	0.978	0.985	0.976	0.996	0.995
	MLANN4	0.982	0.977	0.983	0.975	0.990	0.988
	Blaney-Criddle	0.826	0.538	0.827	0.703	0.821	0.608
V	RBFDE5	**0.995**	**0.995**	**0.983**	**0.982**	**0.996**	**0.995**
	RBFPSO5	0.986	0.986	0.983	0.981	0.995	0.994
	RBFNN5	0.980	0.976	0.973	0.969	0.993	0.991
	MLANN5	0.974	0.973	0.974	0.965	0.988	0.987

*boldface numbers highlight the best results

which is very high as compared to type III soft computing models.

iv. Type IV soft computing models yield better results as compared to all other types of soft computing and empirical models. MAPE ranges between a low of 1.1 to a high of 3.9 at Raipur, followed by 3.7 to 4.4 at Jagdalpur and 2.2 to 4.6 at Ambikapur with RBFDE4 and MLANN4 respectively. MAPE with the Blaney-Criddle method is again quite inferior as compared to type IV soft computing approaches and ranges from 15.5 (at Jagdalpur) to 22.6 (at Raipur).

v. Type V models also produced good results, as reasonably fair estimates of ET_0 can be obtained between a low MAPE of 1.9 with RBFDE5 (at Raipur) to 5.2 with MLANN5 (at Jagdalpur and Ambikapur), which is very much comparable to that of type IV models, even without taking humidity data as one of the input features.

vi. Regarding RMSE, type I soft computing models have resulted in RMSE between 2.98 mm week^{-1} (at Raipur) with RBFDE1 and RBFPSO1 to 3.88 week^{-1} (at Jagdalpur) with MLANN1, as against the higher

RMSE of 6.13 week^{-1} (at Raipur) to 7.53 mm week^{-1} (at Jagdalpur) obtained with Hargreaves approaches.

vii. Type II soft computing models have produced improved RMSE as compared to type I models, which ranges between a low of 2.10 mm week^{-1} with RBFDE2 (at Ambikapur) to a high of 3.45 mm week^{-1} with MLANN2 (at Raipur). Interestingly, at Jagdalpur the soft computing models produce comparatively better estimates of FAO56-PM ET_0 in terms of RMSE as compared with similar models at Jagdalpur and Raipur. In general, type II soft computing models have yielded better ET_0 estimates as compared to Turc methods, for which RMSE ranges between 3.73 (at Ambikapur) to 6.54 mm week^{-1} (at Raipur).

viii. Regarding type III models, RMSE has improved further and is computed between a low of 1.75 mm week^{-1} with RBFDE3 to a high of 2.28 mm week^{-1} with MLANN3 at Raipur, whereas the same for Jagdalpur and Ambikapur, it varied between a low of 1.82 mm week^{-1} with RBFDE3 to a high of 2.48 mm week^{-1} with MLANN3. The equivalent empirical method of Open Pan has produced higher RMSE, which varied between 5.05 mm week^{-1}

in Raipur to 6.55 mm week^{-1} at Ambikapur, which is almost three times more as compared to type III soft computing models.

ix. Similar to MAPE, type IV soft computing models have yielded excellent results in terms of RMSE also. In Raipur, RMSE ranges between the lowest of 0.36 mm week^{-1} with RBFDE4 to the highest of 1.29 mm week^{-1} with MLANN4. At Jagdalpur, it ranges between 1.06 mm week^{-1} with RBFDE4 to 1.25 mm week^{-1} with MLANN4, whereas at Ambikapur, RMSE ranges between 0.68 mm week^{-1} with RBFDE4 to 1.36 mm week^{-1} with MLANN4. The low RMSE values (< 1 mm week^{-1}) obtained with evo-

lutionary optimized hybrid soft computing models (RBFDE4 and RBFDE5) are quite encouraging. This demonstrates the potential of the RBFDE4 and RBFPSO4 models and these models may consider as an alternative to the FAO56-PM empirical approach for ET$_0$ estimation in the study area. In contrast, Blaney-Criddle has produced very high RMSE, which ranges between 4.27 to 7.29 mm/week at different locations, similar to that of the Open Pan method.

x. Type V, soft computing models have also produced better results which is quite identical with type IV models even without including humidity data as an input feature. RMSE with type V models ranges
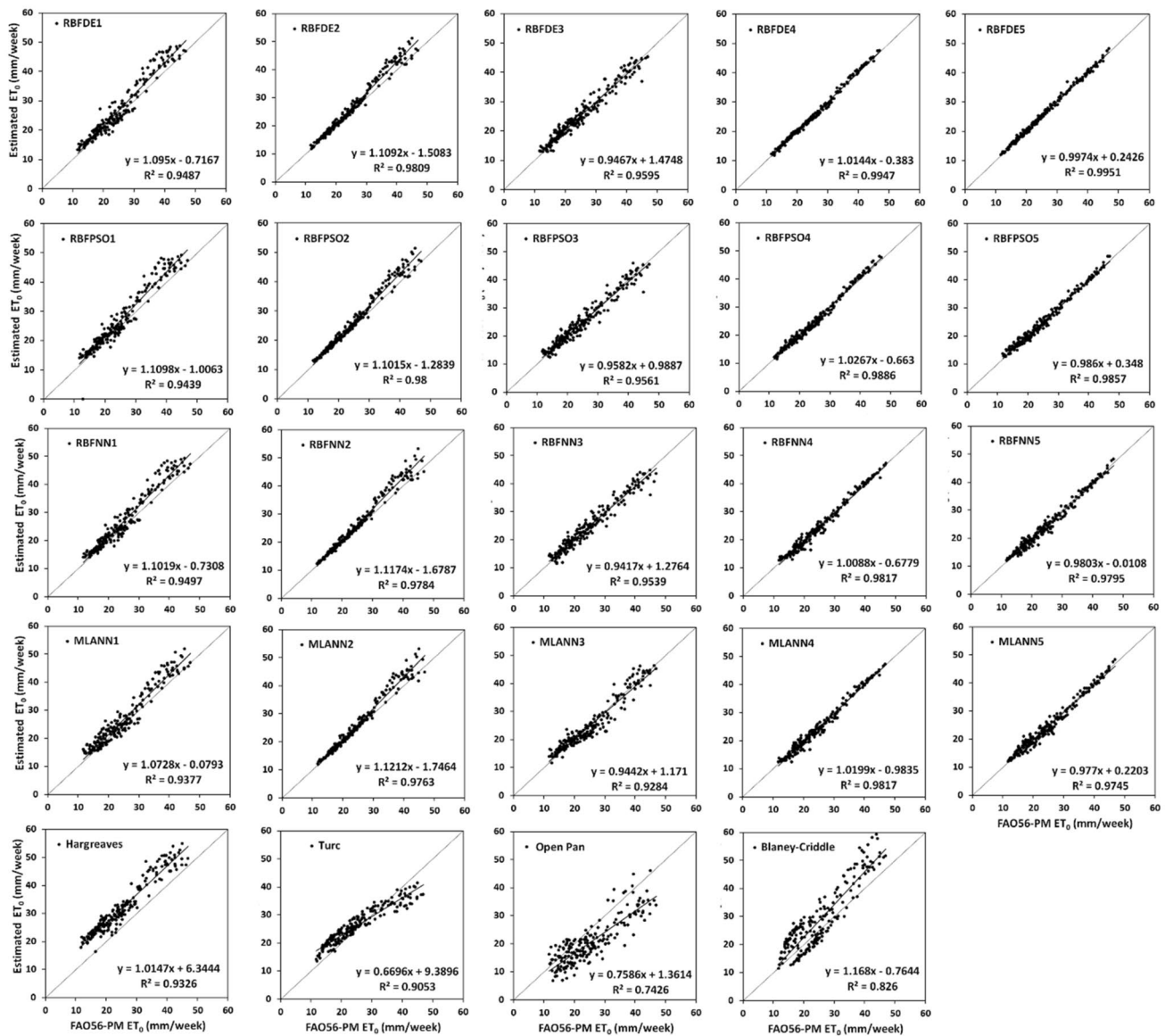
Fig. 6 Relationship between estimated ET$_0$ and FAO56-PM ET$_0$ for different soft computing and empirical models with test data sets at Ambikapur
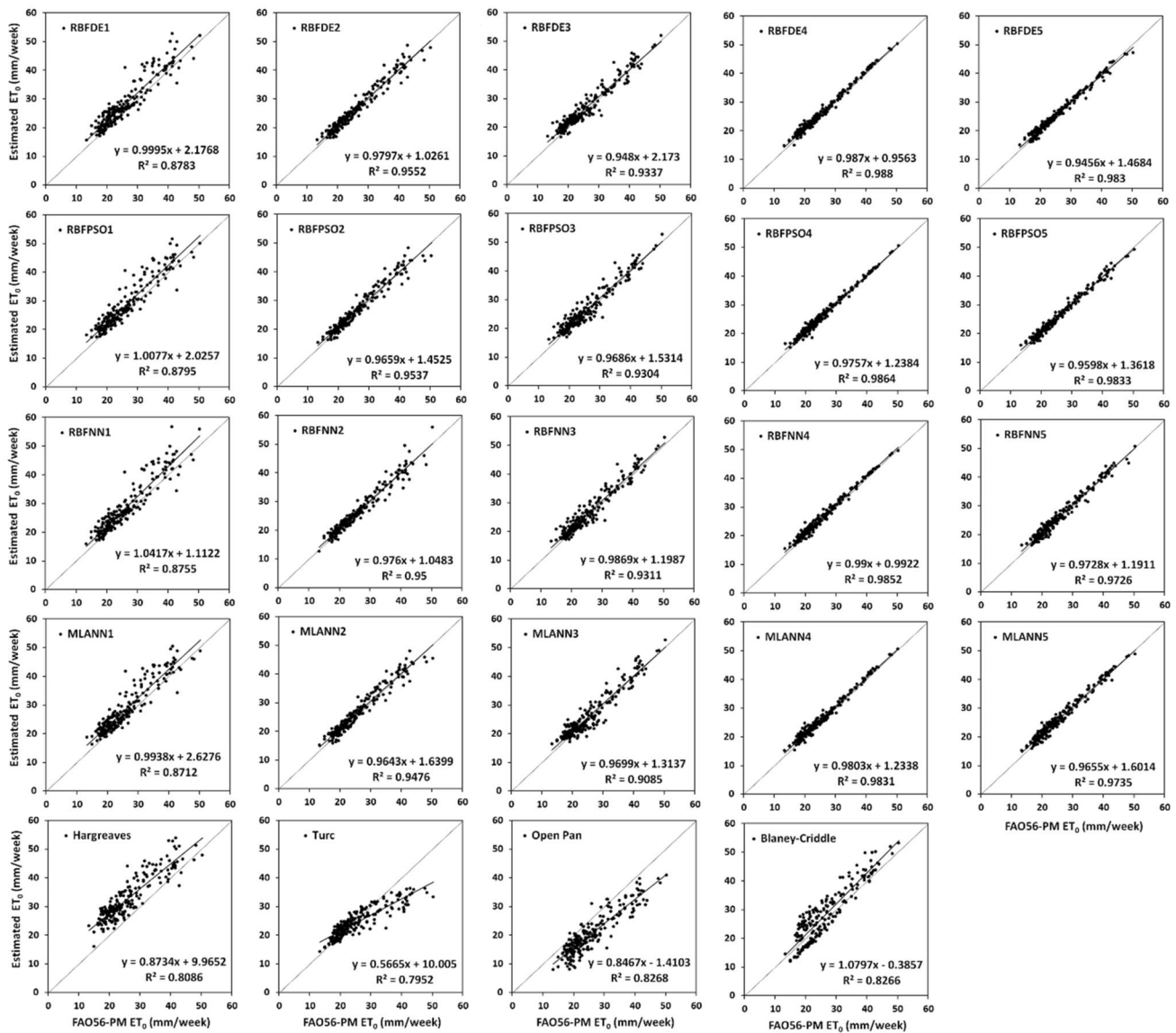
Fig. 7 Relationship between estimated ET_0 and FAO56-PM ET_0 for different soft computing and empirical models with test data sets at Jagdalpur

between lowest 0.66 mm week^{-1} with RBFDE5 to highest of 1.48 mm week^{-1} with MLANN5 at Ambikapur. At Raipur and Jagdalpur, RMSE ranges between 0.80 to 1.31 and 1.06 to 1.46 mm week^{-1} with RBFDE5 and MLANN5 models respectively.

To further examine the relationship between the estimated and FAO56-PM ET_0, two more statistical measures, R^2 and NSE are computed for different soft computing and empirical models and shown in Table 6. The linear relationship between estimated ET_0 and FAO56-ET_0 is also depicted in Figs. 6, 7, and 8 in Ambikapur, Jagdalpur, and Raipur respectively. In general, both R^2 and NSE convey similar information about the model performance and therefore,

the marginal difference is observed between these two performance evaluation measures within a similar type of model in different locations. However, sometimes R^2 values give a false indication and produce higher values close to 1 despite a very high intercept. In such cases corresponding NSE helps in evaluating the model performance.

xi. Type IV soft computing models have produced better R^2 and NSE values as compared to all other models considered for investigation. The highest R^2 values of 0.999, 0.988, and 0.995 are obtained with RBFDE4 and RBFPSO4 in Raipur, Jagdalpur, and Ambikapur respectively with test data sets. The remaining type IV soft computing models also pro-
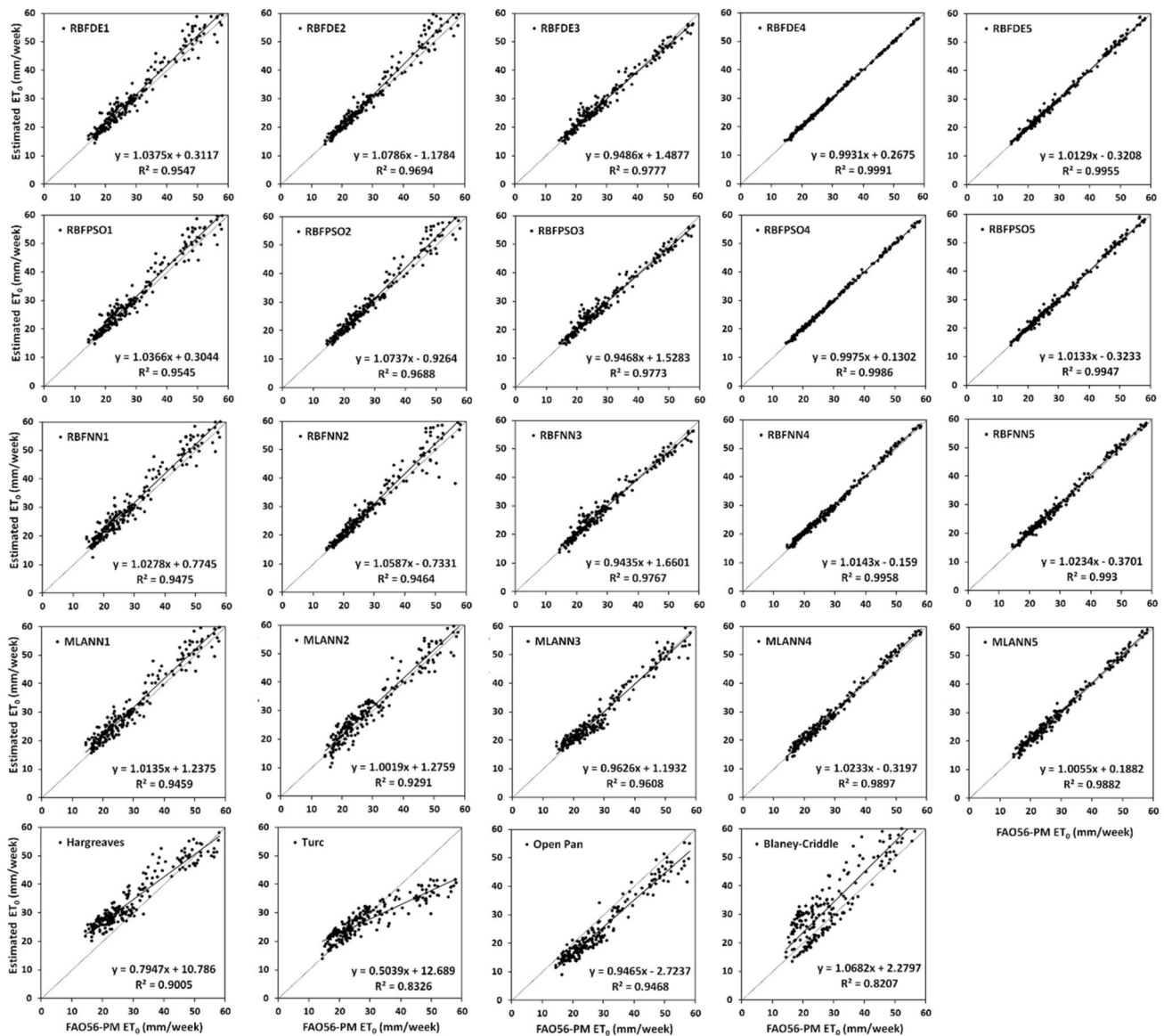
Fig. 8 Relationship between estimated ET_0 and FAO56-PM ET_0 for different soft computing and empirical models with test data sets at Raipur

duced good R^2 and NSE values as compared to the equivalent empirical approach.

xii. Similar results are obtained with type V soft computing models, as R^2 and the corresponding NSE vary between 0.965 to 0.996 in different locations, with RBFDE5 and RBFPSO5 being the best models.

xiii. For the type III model, R^2 and NSE range between 0.961 to 0.978 at Raipur and between 0.900 to 0.960 at Jagdalpur and Ambikapur. R^2 and NSE with type II soft computing models vary between 0.910 to 0.981. Hence, it can be stated that consistent ET_0 estimates with a fair degree of agreement between estimated

and target ET_0 can be obtained using Type both II and III soft computing models.

xiv. Type I soft computing models of RBFDE1 and RBFPSO1 have resulted in slightly lower R^2 and NSE values than RBFNN1 and MLANN1 as compared to remaining types mainly as fewer input features are involved in computations.

xv. Inconsistence and low R^2 and NSE values that have obtained with empirical approaches of Hargreaves, Turc, Open Pan, and Blaney Criddle as compared to their equivalent soft computing models of respective types, clearly establish the fact that soft comput-

ing models produce better estimates of FAO56-PM ET_0 than empirical models.

Results of the performance evaluation analysis indicate that the evolutionary optimized hybrid soft computing models considered for the investigation (RBFDE and RBFPSO) performed consistently better than other conventional soft computing techniques (RBFNN and MLANN) and empirical approaches in all the objectives. From the inferences, it is also evident that when a complete set of the climatic variable is involved in the computation of ET_0 using these models, it looks very difficult to choose between RBFDE and RBFPSO as they look statistically similar in some cases. However, the proposed RBFDE is recommended because of its preciseness and generalization performance in estimating ET_0 in all the stations considered for the study.

4 Conclusions ET_0

The present investigation is carried out to examine the generalized potential of evolutionary optimized hybrid soft computing techniques RBFDE and RBFPSO for the estimation of ET_0 in different ACZs. The ET_0 estimates obtained with proposed RBFDE and RBFPSO models are compared to the conventional neural network (RBFNN, MLANN) and existing empirical approaches. Looking to the scarcity of complete datasets required for computation of FAO-PM ET_0, four variants of each category of soft computing models (RBFDE, RBFPSO, RBFNN, and MLANN) equivalent (in terms of input feature combination) to empirical approaches (Hargreaves, Turc, Open Pan and Blaney-Criddle) is examined. It can be concluded that different soft computing models considered in this investigation, have resulted in improved and more consistent FAO56-PM ET_0 estimates as compared to equivalent empirical approaches. Among the soft computing models, evolutionary models RBFDE and RBFPSO produced a more precise estimation of FAO56-PM ET_0 than conventional RBFNN and MLANN as proposed RBFDE and RBFPSO models resulted in low MAPE and RMSE and high R^2 and NSE close to 1 in most of the cases. However, ET_0 estimates obtained with the proposed RBFDE seems to be slightly better than RBFPSO. Hence, appropriate soft computing models may be recommended for the estimation of ET_0 in other stations of respective ACZs of the study area. The proposed soft computing models may be embedded in crop weather simulation models as subroutines for precise estimation ET_0 with available input features. However, re-calibration and re-validation of these data-driven models are essentially required for their effective implantation in other parts of the world.

Compliance with ethical standards

References

1. Allen RG, Pereira LS, Raes D, Smith M (1998) Introduction to evapotranspiration. In: Crop evapotranspiration-guidelines for computing crop water requirements-FAO Irrigation and drainage paper, vol 56, pp 1–13. FAO, Rome. https://doi.org/10.1016/j.eja.2010.12.001

2. Elizabeth AH, Robert EC (2013) Water balance estimates of evapotranspiration rates in areas with varying land use. In: Evapotranspiration-an overview. InTech. https://doi.org/10.5772/52811

3. Allen RG, Jensen ME, Wright JL, Burman RD (1989) Operational estimates of reference evapotranspiration. Agron J 81(4):650–662. https://doi.org/10.2134/agronj1989.00021962008100040019x

4. Droogers P, Allen RG (2002) Estimating reference evapotranspiration under inaccurate data conditions. Irrigat Drain Syst 16(1):33–45. https://doi.org/10.1023/A:1015508322413

5. Hargreaves GL, Hargreaves GH, Riley JP (1985) Agricultural, benefits for Senegal River Basin. J Irrig Drain Eng ASCE 111:113–124

6. Turc L (1961) Evaluation des besoins en eau d'irrigation, evapotranspiration potentielle, formule climatique simplifice et mise a jour. Ann Agron 12:13–49 **(in French)**

7. Doorenbos J, Pruitt WO (1977) Guidelines for predicting crop water requirements. Irrigation and drainage paper No 24, 2nd edn, Food and Agriculture Organization, Rome, p 156

8. Christiansen JE (1968) Pan evaporation and evapotranspiration by climatic data. J Irrig Drain Div Am Soc Civil Eng 94:243–263

9. Bapuji RB, Sandeep VM, Rao VUM, Venkateswarlu B (2012) Potential Evapotranspiration estimation for Indian conditions: Improving accuracy through calibration coefficients. Technical Bull No 1/2012. All India Co-ordinated Research Project on Agrometeorology, Central Research Institute for Dryland Agriculture, Hyderabad, p 60

10. Chauhan S, Shrivastava RK (2009) Performance evaluation of reference evapotranspiration estimation using climate based methods and artificial neural networks. Water Resour Manag 23(5):825–837

11. Kisi O, Guven A (2010) Evapotranspiration modeling using linear genetic programming technique. J Irrig Drain Eng 136(10):715–723. https://doi.org/10.1061/(asce)ir.1943-4774.0000244

12. Kumar M, Raghuwanshi NS, Singh R (2011) Artificial neural networks approach in evapotranspiration modeling: a review. Irrig Sci. https://doi.org/10.1007/s00271-010-0230-8

13. Mallikarjuna P, Jyothy SA, Sekhar Reddy KC (2012) Daily reference evapotranspiration estimation using linear regression and ANN models. J Inst Eng Ser A 93(4):215–221. https://doi.org/10.1007/s40030-013-0030-2

14. Kisi O (2016) Modeling reference evapotranspiration using three different heuristic regression approaches. Agric Water Manag 169:162–172. https://doi.org/10.1016/j.agwat.2016.02.026

15. Yassin MA, Alazba AA, Mattar MA (2016) Artificial neural networks versus gene expression programming for estimating reference evapotranspiration in an arid climate. Agric Water Manag 163:110–124. https://doi.org/10.1016/j.agwat.2015.09.009

16. Antonopoulos VZ, Antonopoulos AV (2017) Daily reference evapotranspiration estimates by artificial neural networks technique and empirical equations using limited input climate variables. Comput Electron Agric 132:86–96. https://doi.org/10.1016/j.compag.2016.11.011

17. Nema MK, Khare D, Chandniha SK (2017) Application of artificial intelligence to estimate the reference evapotranspiration in sub-humid Doon valley. Appl Water Sci 7(7):3903–3910. https://doi.org/10.1007/s13201-017-0543-3

18. Pandey PK, Nyori T, Pandey V (2017) Estimation of reference evapotranspiration using data-driven techniques under limited data conditions. Model Earth Syst Environ 3(4):1449–1461. https://doi.org/10.1007/s40808-017-0367-z

19. Patil AP, Deka PC (2017) Performance evaluation of hybrid Wavelet-ANN and Wavelet-ANFIS models for estimating evapotranspiration in arid regions of India. Neural Comput Appl 28(2):275–285. https://doi.org/10.1007/s00521-015-2055-0

20. Wen X, Si J, He Z, Wu J, Shao H, Yu H (2015) Support-vector-machine-based models for modeling daily reference evapotranspiration with limited climatic data in extreme arid regions. Water Resour Manag 29(9):3195–3209. https://doi.org/10.1007/s11269-015-0990-2

21. Partal T (2016) Comparison of wavelet-based hybrid models for daily evapotranspiration estimation using meteorological data. KSCE J Civil Eng 20(5):2050–2058. https://doi.org/10.1007/s12205-015-0556-0

22. Kisi O, Demir V (2016) Evapotranspiration estimation using six different multi-layer perceptron algorithms. Irrig Drain Syst Eng. https://doi.org/10.4172/2168-9768.1000164

23. Dou X, Yang Y (2018) Evapotranspiration estimation using four different machine learning approaches in different terrestrial ecosystems. Comput Electron Agric 148:95–106. https://doi.org/10.1016/j.compag.2018.03.010

24. Adamala S (2018) Temperature based generalized wavelet-neural network models to estimate evapotranspiration in India. Inf Process Agric 5(1):149–155. https://doi.org/10.1016/j.inpa.2017.09.004

25. Sanikhani H, Kisi O, Maroufpoor E, Yaseen ZM (2018) Temperature-based modeling of reference evapotranspiration using several artificial intelligence models: application of different modeling scenarios. Theoret Appl Climatol 135(1–2):449–462. https://doi.org/10.1007/s00704-018-2390-z

26. Ozkan C, Kisi O, Akay B (2011) Neural networks with artificial bee colony algorithm for modeling daily reference evapotranspiration. Irrig Sci 29(6):431–441. https://doi.org/10.1007/s00271-010-0254-0

27. Eslamian SS, Gohari SA, Zareian MJ, Firoozfar A (2012) Estimating Penman-Monteith reference evapotranspiration using artificial neural networks and genetic algorithm: a case study. Arab J Sci Eng 37(4):935–944. https://doi.org/10.1007/s13369-012-0214-5

28. Aghajanloo MB, Sabziparvar AA, Hosseinzadeh TP (2013) Artificial neural network-genetic algorithm for estimation of crop evapotranspiration in a semi-arid region of Iran. Neural Comput Appl 23(5):1387–1393. https://doi.org/10.1007/s00521-012-1087-y

29. Petković D, Gocic M, Shamshirband S, Qasem SN, Trajkovic S (2016) Particle swarm optimization-based radial basis function network for estimation of reference evapotranspiration. Theoret Appl Climatol 125(3–4):555–563. https://doi.org/10.1007/s00704-015-1522-y

30. Jovic S, Nedeljkovic B, Golubovic Z, Kostic N (2018) Evolutionary algorithm for reference evapotranspiration analysis. Comput Electron Agric 150:1–4. https://doi.org/10.1016/j.compag.2018.04.003

31. Feng Y, Cui N, Zhao L, Hu X, Gong D (2016) Comparison of ELM, GANN, WNN and empirical models for estimating reference evapotranspiration in the humid region of Southwest China. J Hydrol 536:376–383. https://doi.org/10.1016/j.jhydrol.2016.02.053

32. Gocić M, Motamedi S, Shamshirband S, Petković D, Ch S, Hashim R, Arif M (2015) Soft computing approaches for forecasting reference evapotranspiration. Comput Electron Agric 113:164–173. https://doi.org/10.1016/j.compag.2015.02.010

33. Mehdizadeh S, Behmanesh J, Khalili K (2017) Using MARS, SVM, GEP and empirical equations for estimation of monthly mean reference evapotranspiration. Comput Electron Agric 139:103–114. https://doi.org/10.1016/j.compag.2017.05.002

34. Mattar MA, Alazba AA (2018) GEP and MLR approach for the prediction of reference evapotranspiration. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3410-8

35. Baltzis K (2013) Patented applications of differential evolution in microwave and communication engineering. Recent Patents Comput Sci 6(2):115–128. https://doi.org/10.2174/22132759113069990004

36. Tenaglia GC, Lebensztajn L (2014) A multiobjective approach of differential evolution optimization applied to electromagnetic problems. IEEE Trans Magn. https://doi.org/10.1109/TMAG.2013.2285980

37. Uher V, Gajdoš P, Radecký M, Snášel V (2016) Utilization of the discrete differential evolution for optimization in multidimensional point clouds. Comput Intell Neurosci. https://doi.org/10.1155/2016/6329530

38. Das S, Abraham A, Konar A (2008) Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives. Stud Comput Intell 116:1–38. https://doi.org/10.1007/978-3-540-78297-1_1

39. Hui S, Suganthan PN (2016) Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization. IEEE Trans Cybern 46(1):64–74. https://doi.org/10.1109/TCYB.2015.2394466

40. Ramadas M, Abraham A, Kumar S (2018) RDE reconstructed mutation strategy for differential evolution algorithm. Adv Intell Syst Comput 614:76–85. https://doi.org/10.1007/978-3-319-60618-7_8

41. Ramadas M, Pant M, Abraham A, Kumar S (2019) Segmentation of weather radar image based on hazard severity using RDE: reconstructed mutation strategy for differential evolution algorithm. Neural Comput Appl 31:1253–1261. https://doi.org/10.1007/s00521-017-3091-8

42. Ghosh A, Mallipeddi R, Das S, Das AK (2018) A switched parameter differential evolution with multi-donor mutation and annealing based local search for optimization of lennard-jones atomic clusters. In: 2018 IEEE congress on evolutionary computation,

CEC 2018-Proceedings. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/CEC.2018.8477991

43. Biswas PP, Suganthan PN, Mallipeddi R, Amaratunga GAJ (2018) Optimal power flow solutions using differential evolution algorithm integrated with effective constraint handling techniques. Eng Appl Artif Intell 68:81–100. https://doi.org/10.1016/j.engappai.2017.10.019

44. Rout M, Majhi B, Majhi R, Panda G (2014) Forecasting of currency exchange rates using an adaptive ARMA model with differential evolution based training. J King Saud Univ Comput Inf Sci 26(1):7–18. https://doi.org/10.1016/j.jksuci.2013.01.002

45. Pujari P, Majhi B (2017) Application of natured-inspired technique to Odia handwritten numeral recognition, book chapter in handbook of research on modeling, analysis and application of nature-inspired Metaheuristic algorithms, IGI Global Publication, USA, pp 377–399 https://doi.org/10.4018/978-1-5225-2857-9.ch019

46. World Meteorological Organization (2012) WMO-No. 8-Guide to meteorological instruments and methods of observation, pp I.8-1–I.9-1

47. Broomhead D, Lowe D (1988) Multivariable functional interpolation and adaptive networks. Complex Syst 2:321–355

48. Fernández-Redondo M, Hernández-Espinosa C, Ortiz-Gómez M, Torres-Sospedra J (2004) Training radial basis functions by gradient descent. In: Lecture notes in artificial intelligence (subseries of lecture notes in computer science), Vol 3070, pp 184–189. Springer Verlag. https://doi.org/10.1007/978-3-540-24844-6_23

49. Storn R, Price K (1995) Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, pp 1–12 https://doi.org/10.1.1.1.9696

50. Storn R, Price K (1997) Differential evolution-a simple and efficient Heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359. https://doi.org/10.1023/A:1008202821328

51. Kennedy J, Eberhart R (1995) Particle swarm optimization. Proc IEEE Int Conf Neural Netw 4:1942–1948

52. Eberhart R, Kennedy J (2002) A new optimizer using particle swarm theory. In: Institute of Electrical and Electronics Engineers (IEEE), pp 39–43 https://doi.org/10.1109/mhs.1995.494215

53. Eberhart R, Yuhui S (2002) Particle swarm optimization: developments, applications and resources. In: Institute of Electrical and Electronics Engineers (IEEE), pp 81–86 https://doi.org/10.1109/cec.2001.934374

54. del Valle Y, Venayagamoorthy GK, Mohagheghi S, Hernandez JC, Harley RG (2008) Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Trans Evol Comput. https://doi.org/10.1109/TEVC.2007.896686

55. Chen LF, Su CT, Chen KH (2012) An improved particle swarm optimization for feature selection. Intell Data Anal 16(2):167–182. https://doi.org/10.3233/IDA-2012-0517

56. Zhang Y, Wang S, Ji GA (2015) Comprehensive survey on particle Swarm optimization algorithm and its applications. Mathematical Problems in Engineering, pp 1–38 https://doi.org/10.1155/2015/931256

57. Haykin S (1998) Neural networks–a comprehensive foundation, 2nd edn. Prentice-Hall, Upper Saddle River, pp 26–32

58. Nash JE, Sutcliffe JV (1970) River flow forecasting through conceptual models part I-a discussion of principles. J Hydrol 10(3):282–290