



Improved prediction of daily pan evaporation using Deep-LSTM model

Babita Majhi¹ · Diwakar Naidu^{1,2} · Ambika Prasad Mishra³ · Suresh Chandra Satapathy⁴

Received: 28 November 2018 / Accepted: 26 February 2019 / Published online: 6 March 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

Precise measurement or estimation of evaporation losses is extremely important for the development of water resource management strategies and its effective implementation, particularly in drought-prone areas for increasing agricultural productivity. Evaporation can either be measured directly using evaporimeters, or it can be estimated by means of empirical models with the help of climatic factors influencing evaporation process. In general, variations in climatic factors such as temperature, humidity, wind speed, sunshine and solar radiation influence and control the evaporation process to a great extent. Due to the highly nonlinear nature of evaporation phenomenon, it is invariably very difficult to model the evaporation process through climatic factors especially in diverse agro-climatic situations. The present investigation is carried out to examine the potential of deep neural network architecture with long short-term memory cell (Deep-LSTM) to estimate daily pan evaporation with minimum input features. Depending upon the availability of climatic data Deep-LSTM models with different input combinations are proposed to model daily evaporation losses in three agro-climatic zones of Chhattisgarh state in east-central India. The performance of the proposed Deep-LSTM models are compared with commonly used multilayer artificial neural network and empirical methods (Hargreaves and Blaney–Criddle). The results of the investigations in terms of various performance evaluation criteria reveal that the proposed Deep-LSTM structure is able to successfully model the daily evaporation losses with improved accuracy as compared to other models considered in this study.

Keywords Evaporation estimation · Deep-LSTM · MLANN · Empirical methods · ACZs

Abbreviations

EP Pan evaporation
Deep-LSTM Deep neural network architecture with long short-term memory cell

ACZs	Agro-climatic zones
MLANN	Multilayer artificial neural network
T_{\max}	Maximum temperature
T_{\min}	Minimum temperature
RH_I	Relative humidity morning
RH_{II}	Relative humidity afternoon
WS	Wind speed
BSS	Bright sunshine hours
SD	Standard deviation
ET_0	Reference evapotranspiration
CV	Coefficient of variation
R	Correlation coefficient
RMSE	Root-mean-square error
R^2	Coefficient of determination
EF	Efficiency factor
AIC	Akaike information criterion

✉ Babita Majhi
babita.majhi@gmail.com

Diwakar Naidu
dnaidu1971@gmail.com

Ambika Prasad Mishra
ambikaprasad.mishra@gmail.com

Suresh Chandra Satapathy
sureshsatapathy@ieee.org

¹ Department of Computer Science and Information Technology, Guru Ghasidas Vishwavidyalaya, Central University, Bilaspur 495009, India

² BRSM College of Agricultural Engineering and Technology and Research Station, Indira Gandhi Krishi Vishwavidyalaya, Mungeli, India

³ Department of CSE, ITER, Siksha 'O' Anusandhan University, Bhubaneswar, India

⁴ School of Computer Engineering, KIIT University, Bhubaneswar, India

Symbols used

g_t	Input node at time t
\tanh	Hyperbolic tangent function
x_t	Input to the memory cell at time t

W_{gx}	Weight matrix between input layer of the network and input node of the memory cell
h_{t-1}	Hidden state input at time $t - 1$
W_{gh}	Weight matrix between hidden states at different time steps
$\text{bias}_{\text{input node}}$	Bias to the input node
i_t	Input gate at time t
σ	Sigmoidal activation function
s_t	Internal state at time t
s_{t-1}	Internal state at time $t-1$
$\text{bias}_{\text{input gate}}$	Bias to the input gate
f_t	Forget state at time t
\odot	Point-wise linear operator
W_{fx}	Weight matrix between forget gates and input layer
W_{fh}	Weight matrix between forget gates and hidden states
h_t	Final output of memory cell at time t
$\text{bias}_{\text{forget}}$	Bias for the forget gate
O_t	Output gate at time t
W_{ox}	Weight matrix between output gates and input layers
W_{oh}	Weight matrix between output gates and hidden states
$\text{bias}_{\text{output gates}}$	Bias for the output gate

1 Introduction

Agricultural production is mainly dependent on the effective utilization of the available water resources, especially under drought-prone, dry, sub-humid and semi-arid climatic regions. For efficient water resource management, measurement or accurate estimation of evaporation losses is extremely important [1, 2]. The pan evaporation (EP) is considered as most valuable input for determining crop water requirement, irrigation scheduling, rainfall runoff modeling, computation of balance parameters, etc., to ensure judicious use of available water resources. Evaporation is a surface phenomenon in which liquid water gets converted to a gaseous form below its boiling point. The state of climatic variables such as temperature, humidity, wind speed and sunshine surrounding the evaporating water surface hugely influences the process of evaporation from the water bodies. Higher temperature increases the kinetic energy of water particles at the surface, and the inter-particle space between water particles also gets increased. As a result, the inter-particle force of attraction between the water particles at the water surface decreases, and because of that, liquid water gets converted into gaseous form and gets evaporated. Another climatic factor that influences the evaporation process is humidity which is a

measure of water vapor present in the air. When air humidity is less, more amount of water vapor gets accommodated in the air and thereby the rate of evaporation from the evaporating surface increases with the decrease in air humidity. Wind speed is another climatic factor which triggers the evaporation process. When wind speed increases, it carries away greater amount of water vapor present in the air surrounding the water surface and hence the amount of water vapor that can be accommodated in the air increases and thereby more water particles get converted into gaseous form and the rate of evaporation increases.

Evaporation is measured precisely using Class A evaporation pan standardized by the National Weather Service of the USA. Installation and maintenance of such equipment for recording evaporation on a daily basis is a cumbersome task and requires a skilled workforce [3]. Alternatively, it is often estimated using different climatic variables affecting the evaporation process through an empirical approach. Due to the highly complex physical and nonlinear nature of the evaporation process, it is furthermore difficult to model evaporation through empirical methods as well [4]. Moreover, empirical model developed for one agro-climatic situation may not perform well in other situations and needs recalibration of model coefficients before its implementation. Attempts are made by researchers to model evaporation process and to develop several empirical formulae in the past which are discussed in the literature [5–10]. Among empirical methods, evaporation estimates obtained through Penman equation are considered as most precise, and therefore, it is widely used and globally accepted. However, application of Penman method is limited as it requires other climatic inputs such as net radiation and vapor pressure deficit.

Considering the limitations associated with both measurement and empirical approaches discussed so far for evaporation estimation, in the recent past researchers employed several data-driven computational intelligence and machine learning techniques with different optimization algorithms and have provided some alternate machine learning solutions to the problem with different input combinations of available climatic variables such as temperature, humidity, wind speed, sunshine, solar radiation and vapor pressure [11–22]. A comprehensive review of the available literature has been carried out, and significant results of some of the recently published research articles are discussed briefly in this section. In a study carried out by Deo et al. [23] monthly evaporative losses had been estimated using three machine learning techniques, namely relevance vector machine (RVM), extreme learning machine (ELM) and multivariate adaptive regression spline, using meteorological parameters as predictor variable and RVM was found to be the best predictor among

these. Wang et al. [24] have investigated the potential of multilayer perceptron (MLP), generalized regression neural network (GRNN), fuzzy genetic (FG), least square support vector machine (LSSVM), multivariate adaptive regression spline (MARS) and adaptive neuro-fuzzy inference systems with grid partition (ANFIS-GP) for estimating evaporation and compared the results with regression methods in different climates of China. They have found that heuristic technique generally performed better than regression and empirical methods. The MLP ranked first concerning accuracy among other complex nonlinear heuristic models considered in this study. In another investigation carried out by Wang et al. [25], daily EP is estimated using fuzzy genetic (FG), least square support vector regression (LSSVR), multivariate adaptive regression spline (MARS), M5 model tree (M5Tree) and multiple linear regression (MLR) for eight stations around Dongting Lake basin in China. Investigations suggest that FG and LSSVR provide better performance over other machine learning techniques. Monthly EP has been estimated by Malik et al. [26] in the Indian central Himalayas region, employing MLPNN, co-active neuro-fuzzy inference system (CANFIS), radial basis neural network (RBNN) and self-organizing map neural network (SOMNN). Gamma test is used for the selection of appropriate input combination. They reported the superiority of CANFIS over other techniques. Tezal and Buyukyildiz [27] have studied the applicability of MLP, RBFN and ε -support vector regression (SVR) using different training algorithms. Both ANNs and SVR with a scaled conjugate gradient (SCG) learning have performed better as compared to empirical methods. In one of the studies, Kisi et al. [28] have explored the potential of decision tree-based machine learning methods such as Chi-square automatic interaction detector (CHAID) and classification and regression tree (CART) and compared these with the neural network model for daily EP estimation in Turkey. Comparison of results show that neural networks performed better as compared to other models in different scenarios. The conjugate gradient optimization method is employed to calibrate three nonlinear mathematical models in a few locations of Iran by Keshtegar et al. [29]. The results indicate that proposed models ranked higher than adaptive neuro-fuzzy inference system (ANFIS) and M5 model tree (M5Tree) models. Goyal et al. [30] have examined the applicability of ANN, LSSVR, fuzzy logic (FL) and ANFIS techniques in estimating daily EP, and the results are compared with empirical methods suggested by Hargreaves and Samani (HGS) and the Stephens–Stewart (SS). Investigation unveils that daily evaporation can be modeled successfully and more accurately by FL and LSSVR techniques which are superior to traditional approaches. In addition, machine learning and evolutionary techniques have been

successfully implemented in various other fields including biomedical science for prediction of proteins' secondary structure [31], load frequency controller design and renewable distributed generations [32–34] and for solving second-order boundary value problems and fuzzy differential equations [35–38].

It is learnt from the literature review that among different machine learning methods applied so far, ANNs with appropriate learning algorithm have proven potentially capable of modeling evaporation process in diverse locations and have performed better than more complex structures. Prediction task is nonlinear in nature, and hence, the adaptive model for prediction should have nonlinear characteristics. Out of different ANN structures reported in the literature, the Deep-LSTM is capable of capturing higher-order nonlinear features. The Deep-LSTM is a stack of LSTM units where different orders of nonlinear feature representation are captured by LSTM unit at different depths, which explores the inherent features of time series over longer time period to attain improved prediction performance [39, 40]. Since nonlinear features are more suited for nonlinear prediction, the Deep-LSTM is a better candidate for prediction of daily EP. Hence, Deep-LSTM is employed for prediction purpose in this paper.

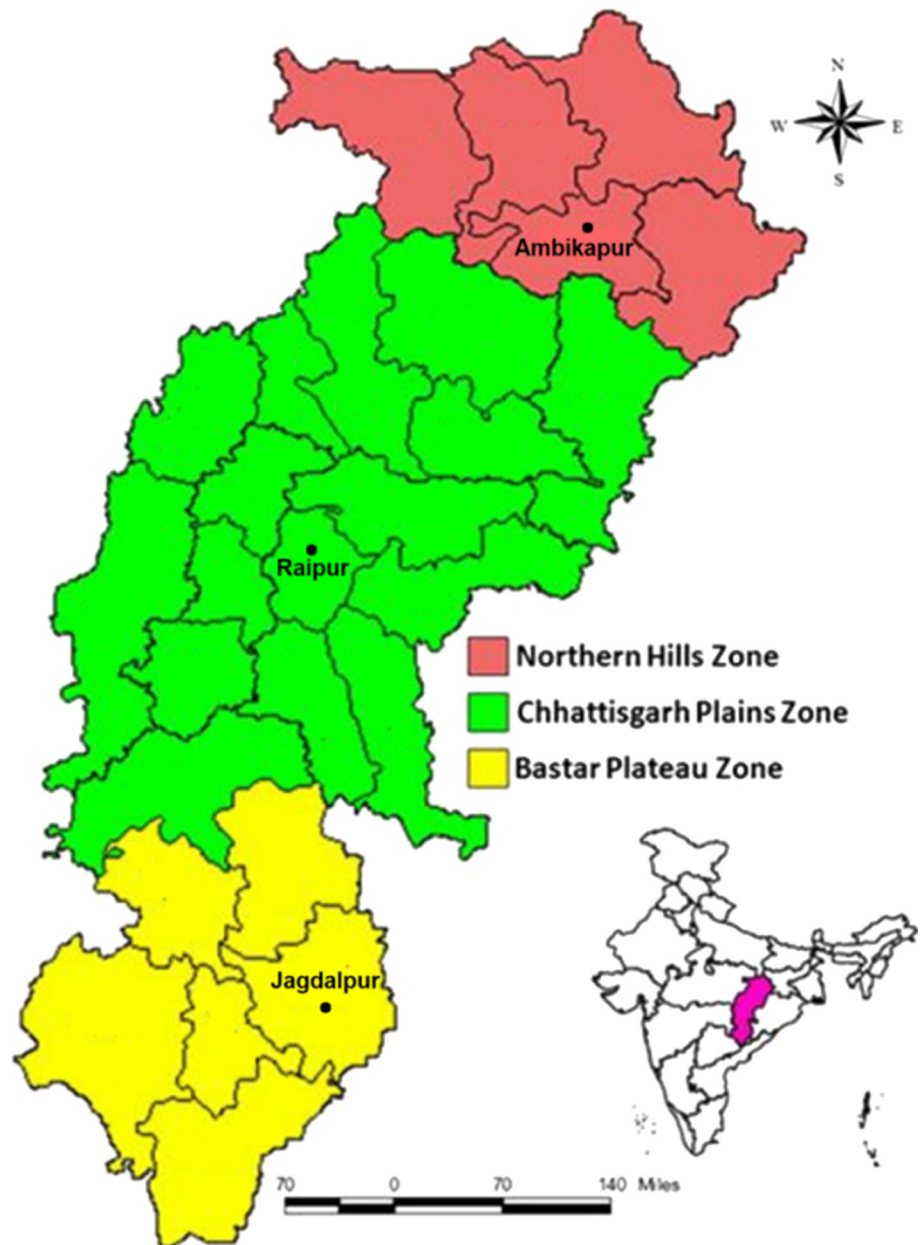
The methodology section provides detailed description about the study locations, data sets, architecture and implementation of Deep-LSTM neural network, MLANN and empirical methods (Blaney–Criddle and Hargreaves) considered in this paper. Simulation study and the results obtained are elaborated in the subsequent section. Finally, the contribution of the study is summarized in the concluding section at the end.

2 Methodology

2.1 Study area

The present investigation is carried out for three representative stations: Raipur, Jagdalpur and Ambikapur, from three distinct agro-climatic zones (ACZs) of Chhattisgarh state in east-central India (Fig. 1). ACZ refers to a land unit in terms of its major climate and growing period which is climatically suitable for certain range of crops and cultivars. The climate of Chhattisgarh is dry and sub-humid in general with potential evaporation losses being more than the average annual rainfall of the state, which is about 1400 mm. Raipur is located in Chhattisgarh plains ACZ with average annual of about 1200 mm, whereas Jagdalpur and Ambikapur are located in Bastar plateau and Northern hills ACZs with an average annual rainfall of 1400 mm and 1600 mm, respectively. Long-term daily weather data on maximum temperature (T_{max}), minimum temperature

Fig. 1 Location map of the study area



(T_{\min}), morning and afternoon relative humidity (RH_I and RH_{II}), wind speed (WS), bright sunshine hours (BSS) and EP (EP) are collected from meteorological observatories located at respective stations. All these meteorological observatories are well maintained and certified by the India Meteorological Department, Govt. of India. The details about data sets are given in Table 1. Descriptive statistics of the data sets considered for this investigation are presented in Table 2.

2.2 Deep-LSTM architecture

Deep neural networks are a class of recursive feedforward networks, which can extract and learn features which are deeply embedded in the data. Deep networks are broadly categorized into two classes: classical and modern deep networks. Recently, deep learning techniques have been successfully employed in natural language processing [41], sequence learning [42] and time series predictions such as financial market and wind forecasting [43, 44]. Deep networks differ from other feedforward networks in terms of independence of the connecting nodes. In traditional

Table 1 Data sets used for the study

Perimeters	Raipur	Jagdapur	Ambikapur
Latitude (°North)	21.14	19.08	23.12
Longitude (°East)	81.38	82.01	83.20
Altitude (m)	289	564	604
Total number of patterns	12,783	9131	6210
Patterns used for training (80%)	10,277	7305	4968
Period of the training data set	1981–2008	1993–2013	1999–2011
Patterns used for testing (20%)	2556	1826	1242
Period of the testing data set	2009–2015	2013–2017	2012–2015

Table 2 Descriptive statistics of daily climatic variables of Raipur, Jagdalpur and Ambikapur

Station	Maximum	Minimum	Range	Mean	SD	CV	<i>R</i>
<i>T</i> _{max} (°C)							
Raipur	47.6	15.4	32.2	32.7	5.3	16.1	0.90
Jagdapur	44.4	18.0	26.4	30.9	4.3	13.8	0.85
Ambikapur	45.6	15.6	30.0	30.3	5.4	17.9	0.79
<i>T</i> _{min} (°C)							
Raipur	34.1	3.6	30.5	19.8	6.0	30.3	0.49
Jagdapur	32.5	1.0	31.5	18.3	5.8	31.9	0.31
Ambikapur	31.9	0.4	31.5	17.7	6.6	36.9	0.48
RH _I (%)							
Raipur	100	9	91	79.4	18.8	23.7	− 0.86
Jagdapur	100	21	79	86.3	11.6	13.4	− 0.74
Ambikapur	100	19	81	78.9	18.8	23.8	− 0.78
RH _{II} (%)							
Raipur	100	3	97	44.2	24.1	54.5	− 0.53
Jagdapur	100	3	97	49.5	23.1	46.7	− 0.54
Ambikapur	100	5	95	46.4	23.9	51.6	− 0.49
WS (kmph)							
Raipur	31.4	0.1	31.3	5.5	3.9	70.7	0.36
Jagdapur	22.3	0.1	22.2	4.8	2.7	57.1	0.24
Ambikapur	16.3	0.0	16.3	3.6	2.3	63.2	0.38
BSS hours							
Raipur	12.0	0.0	12.0	6.9	3.4	49.2	0.33
Jagdapur	12.0	0.0	12.0	6.4	3.4	53.8	0.44
Ambikapur	12.3	0.0	12.3	7.2	3.3	45.3	0.34
EP (mm)							
Raipur	21.5	0.1	21.4	5.5	3.5	64.5	1.00
Jagdapur	15.2	0.0	15.2	4.2	2.4	55.6	1.00
Ambikapur	14.8	0.0	14.8	4.6	2.5	53.8	1.00

feedforward network, nodes receive input from the previous node only and are independent of all other nodes. In deep networks, the nodes are massively interdependent and share weights which signify the idea of long-term dependencies. In other words, the current node receives input not only from the previous node but also from many other previous nodes. There may be nodes which have self-connection loops as well, which signifies interconnected hidden states for the same node in the time domain. The

long-term dependency on the input requires the network to keep previous states in memory. Conventional recurrent networks face a vanishing gradient problem while dealing with the need for storing information about long-term inputs. The vanishing gradient problem is a condition in which the input between different hidden states at different time steps decreases exponentially. Long short-term memory (LSTM) networks are a class of recurrent networks which handle vanishing gradient problem efficiently.

LSTM networks have been successfully applied in natural language processing. In this study, we have constructed a deep recurrent network comprising of layers of LSTM networks subsequently named as Deep-LSTM network. The synthesized network suitably utilizes the advantages of hugely successful deep networks and LSTM recurrent networks. Deep-LSTM networks [45] manage the vanishing gradient problem by incorporating the idea of memory cells. Deep-LSTM neural network consists of some internal contextual state cells that act as long-term or short-term memory cells. The output of Deep-LSTM neural network is dependent on the state of these cells. This feature assists the network for the prediction purpose because such task needs the historical context of inputs, rather only the last input. The working mechanism of Deep-LSTM networks is solely dependent upon the memory cell. Memory cells have different subunits with different objectives as shown in Fig. 2. The working principle of Fig. 2 is dealt in brief.

The input node g_t receives input x_t , from the input layer of the deep network and from the previous hidden states h_{t-1} of the node itself in time steps. The data to be predicted are nonlinear in nature. Hence, the model which is used to predict nonlinear output should have nonlinear element. The sigmoid is a nonlinear function and helps improve prediction accuracy. Therefore, the weighted sum of x_t and h_{t-1} is passed through a tanh function, as given in Eq. 1.

$$g_t = \tanh(x_t \cdot W_{gx} + h_{t-1}W_{gh} + bias_{input\ node}) \quad (1)$$

The input gate (i_t) is similar to the input node as this also receives the same inputs as the input node. This is a unit sigmoidal activation function. It is termed as input gate as

it blocks the flow of inputs from other nodes to the current node, if its net value is zero. It allows the values to pass through if the net value is one. Its operation is represented by Eq. 2.

$$i_t = \sigma(x_t \cdot W_{gx} + h_{t-1}W_{gh} + bias_{input\ gate}) \quad (2)$$

The internal state s_t is a node with a self-loop recurrent edge of unit weight and a linear activation function, which is updated using Eq. 3.

$$s_t = i_t \odot g_t + s_{t-1} \quad (3)$$

The forget state (f_t) is a subunit to reinitiate the internal state of the memory cell and is formulated as Eq. 4.

$$f_t = \sigma(x_t \cdot W_{hx} + h_{t-1}W_{fh} + bias_{forget}) \quad (4)$$

Finally, the output gates O_t perform the task given in Eq. 5.

$$O_t = \sigma(x_t \cdot W_{ox} + h_{t-1}W_{oh} + bias_{output\ gate}) \quad (5)$$

The final output of the memory cell is computed using Eq. 6.

$$h_t = \tanh(s_t) \odot O_t \quad (6)$$

where $s_t = g_t \odot i_t + s_{t-1} \odot f_t$

The network architecture of Deep-LSTM used for simulation study is shown in Table 3. The output layer has only one node and hence not provided in this table.

2.3 Multilayer artificial neural network (MLANN)

A commonly used neural network structure, MLANN suggested by Haykin [46] consists of an input layer, one

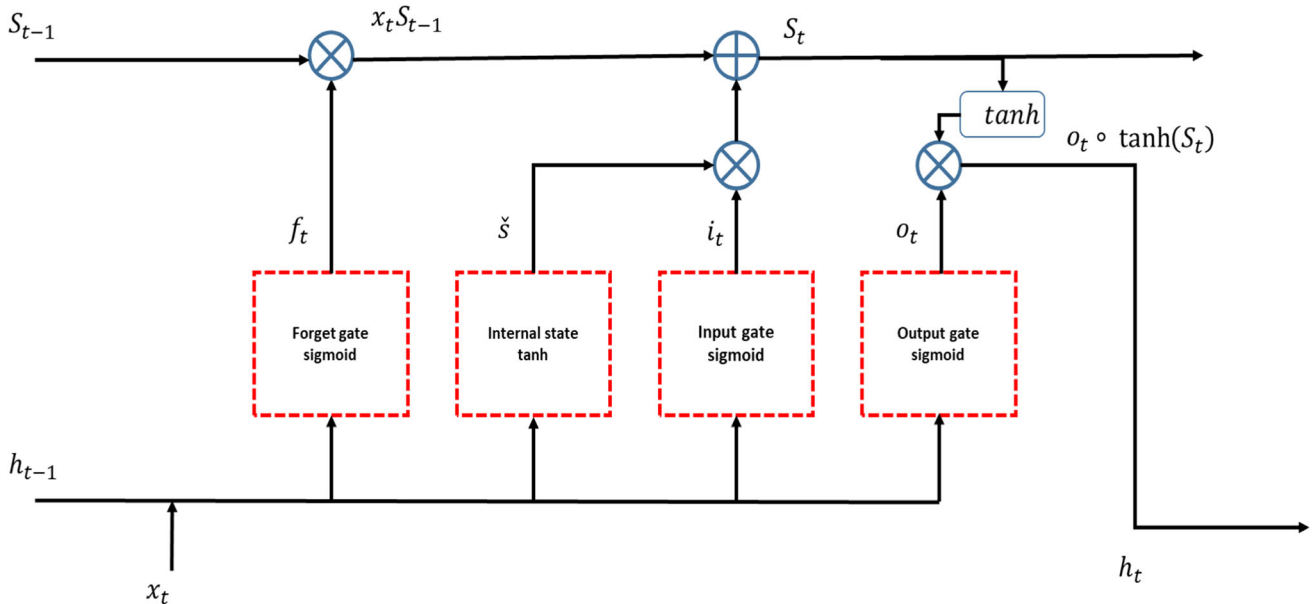


Fig. 2 A block diagram of an LSTM network

Table 3 Architecture of LSTM model

LSTM layer nodes	Dense layer-1 nodes	Dense layer-2 nodes	Batch size	Epoch
32	20	20	72	300

Table 4 Architecture of MLANN models used

Parameters	MLANN
Number of input features	$N = 2, 4, 5$ and 6
Number of neurons in the hidden layer	5
Number of neurons in the output layer	1
Activation function	Hyperbolic tangent (tanh)

intermediate hidden layer and an output layer. An N-5-1 structure of MLANN, with N the number of input nodes, five neurons in the hidden layer and one neuron at the output layer, is considered in this study. The training of the weights of different layers is carried out by conventional backpropagation algorithm. Basically, there are two passes through the different layers of the network: forward pass and the backward pass. The forward pass produces an estimated output. The output error term in a modified form is backpropagated from last layer to input layer to adjust the connecting bias and weights of different layers. The specifications of the MLANN structure used in this study are given in Table 4.

2.4 Hargreaves method

Hargreaves et al. [47] have suggested computing the potential atmospheric evaporative demand termed as reference evapotranspiration (ET_0) with maximum and minimum temperatures as

$$ET_0 = 0.0023R_a T_d 0.5(T_m + 17.8) \tag{7}$$

where R_a = water equivalent of extra-terrestrial radiation (mm day^{-1}), T_d = difference between maximum and minimum temperatures ($^{\circ}\text{C}$), and T_m = mean temperature ($^{\circ}\text{C}$).

2.5 Blaney–Criddle method

Blaney–Criddle empirical equation reported in FAO irrigation and drainage paper no. 24 [48] is used to compute reference evapotranspiration (ET_0) with available data on temperature, humidity, wind speed and sunshine hours. The empirical equation is given as

$$ET_0 = a + b[p(0.46T + 8.13)] \tag{8}$$

where $a = 0.0043 \text{ RH}_{II} - (n/N) - 1.41$, $b = a_0 + a_1 \text{ RH}_{II} + a_2(n/N) + a_3 U_d + a_4 \text{ RH}_{II}(n/N) + a_5 \text{ RH}_{\text{min}} U_d$,

ET_0 = reference evapotranspiration in mm day^{-1} , $T = (T_{\text{max}} + T_{\text{min}})/2$ = mean daily temperature in $^{\circ}\text{C}$, p = mean daily percentage of total annual daytime hours, n/N = ratio of possible to actual sunshine hours, RH_{min} = minimum daily relative humidity in percentage, U_d = daytime wind at 2 m height (ms^{-1}), $a_0 = 0.81917$, $a_1 = 0.0040922$, $a_2 = 1.0705$, $a_3 = 0.065649$, $a_4 = 0.0059684$, and $a_5 = 0.0005967$.

3 Performance evaluation criteria

The performance of the proposed prediction model is evaluated by computing root-mean-square error (RMSE), coefficient of determination (R^2) and model efficiency factor (EF) [49] between desired and estimated values of evaporation for the data sets considered. These are defined as

$$RMSE = \sqrt{\frac{1}{T} \sum_{i=1}^T (\text{Out}_{\text{est}} - \text{Out}_{\text{obs}})^2} \tag{9}$$

$$R^2 = \frac{(\sum_{i=1}^T (\text{Out}_{\text{obs}} - \overline{\text{Out}_{\text{obs}}})(\text{Out}_{\text{est}} - \overline{\text{Out}_{\text{est}}}))^2}{\sum_{i=1}^T (\text{Out}_{\text{obs}} - \overline{\text{Out}_{\text{obs}}})^2 \sum_{i=1}^T (\text{Out}_{\text{est}} - \overline{\text{Out}_{\text{est}}})^2} \tag{10}$$

$$EF = 1 - \frac{\sum_{i=1}^T (\text{Out}_{\text{est}} - \text{Out}_{\text{obs}})^2}{\sum_{i=1}^T (\text{Out}_{\text{obs}} - \overline{\text{Out}_{\text{obs}}})^2} \quad (-\infty \leq EF \leq 1) \tag{11}$$

where Out_{obs} and Out_{est} represent the desired and estimated evaporation values, respectively. T is the total number of input patterns, and i denotes the number of particular instances of input patterns. RMSE value should be close to 0, and R^2 and EF values should be near to 1.

4 Simulation study and results

To estimate the daily evaporation with the help of data, the Deep-LSTM and MLANN models are simulated in Python and MATLAB, respectively, with different input combinations as shown in Table 5. The number followed by model name represents the number of input parameters. Availability of consistent long-term weather data has always been one of the significant constraints in deciding input combination. Hence, the correlation coefficient between daily climatic factors influencing evaporation

Table 5 Input data combination used in Deep-LSTM, MLANN and empirical models

Model	Input combination	Number of inputs
Deep-LSTM-2, MLANN-2 and Hargreaves	T_{\max} , T_{\min}	2
Deep-LSTM-4 and MLANN-4	T_{\max} , T_{\min} , RH _I , RH _{II}	4
Deep-LSTM-5 and MLANN-5	T_{\max} , T_{\min} , RH _I , RH _{II} , WS	5
Deep-LSTM-6, MLANN-6 and Blaney–Criddle	T_{\max} , T_{\min} , RH _I , RH _{II} , WS, BSS	6

process and EP values (Table 2) forms the basis for selection of input combinations. The main focus of this study is to make effective utilization of available climatic data and to model the daily evaporation process with minimum input parameters with higher accuracy.

A significantly large number of available daily input patterns, i.e., 17–35 years of daily weather data, are used for simulation study for each of the data sets as shown in Table 1. To attain consistency, each data set is normalized between 0 and 1 before presenting to the model for training and testing and then renormalized to its original unit for final comparison between actual and estimated outputs.

$$(X_k - X_{\min}) / (X_{\max} - X_{\min}) \quad (12)$$

where $X_k = k$ th sample value of the input parameter, X_{\min} = minimum of the input parameter, and X_{\max} = maximum of the input parameter.

Training of the proposed models with desired input combination is done with 80% of the available data for model development, and the remaining 20% data are used to test the model performance. In order to train the MLANN model, the first pattern is given as input to neural network, and after the forward pass and estimation process, the final output is obtained at the output node. All the training patterns are applied sequentially. The process continues till the remaining input patterns are exhausted. The outputs corresponding to each input pattern are compared with the desired output to produce error term. The change in weights in each path is calculated using backpropagation learning algorithm. The change in weight of each path of the model is stored for every input. Then the average change in weight in each path is calculated. The weights are then updated by adding the average change in weight of each path. This constitutes one iteration. The same process is continued for 5000 iterations. The value of the convergence coefficient (μ) is fixed at 0.01 as it provides better training. This completes one experiment, and the same experiment is repeated for ten independent times. For steady-state estimation of different weights, in each iteration, the root-mean-square error (RMSE) is computed using error value of each pattern. The training process is stopped when the RMSE value achieves the best possible minimum value. After the training process is over, the weights and biases at each layer of the neural network are fixed according to

final iteration. To validate the prediction performance of this model, test patterns are fed sequentially. For each test pattern, the estimated output is obtained and compared with the desired values using performance evaluation measures RMSE, R^2 and EF for each of the models and data sets. Model parameters during training are optimized to minimize the RMSE toward zero and maximize the R^2 and EF toward one for achieving improved performance.

Similarly, the Deep-LSTM is trained for three different data sets used for the study with the same basic configuration as given in Table 3. The network has one LSTM layer, two dense layers and one output layer. Hyperbolic tangent function (tanh) is used for activation and hard sigmoid function for recurrent activation, both of which are default in LSTM layer. For output layer sigmoid activation function is used. Default ‘glorot uniform,’ for kernel initializers and bias initializers as ‘zeros’ are used for LSTM layer. For better convergence, Adam optimizers [50] with default parameter settings ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and learning rate = 0.01) have been used. Layer_1 or Layer_2 regularization is not used as previous studies have found that model performance does not improve with regularization for sequence learning problem [51, 52]. Dropout affected the performance of the model. Hence, subsequently, dropout is not preferred in any of the layers. All the architectures that have been implemented for this study are achieved by using open-source software library Tensorflow [53], Keras high-level neural networks API [54] and scikit-learn [55] on a Dell PowerEdge T130 server, set to CPU execution. The stopping criterion used during training phase is attainment of minimum and consistent root-mean-square error. The training is stopped when the RMSE attains a possible minimum value and then remains almost constant.

Test performance based on evaluation criteria considered for the proposed models is shown in Table 6. It is observed from the inferences that marked improvement has been observed in terms of RMSE, R^2 and EF with the proposed Deep-LSTM models over conventional MLANN and empirical models for each of the data sets. The RMSE values at Raipur improved from 1.21 to 0.98 with a Deep-LSTM model against the RMSE values 1.40–1.15 with MLANN with increasing number of input features. The magnitude of improvement is less with the increasing number of input features. In the other two locations

Table 6 Comparison of test performance of neural network (Deep-LSTM and MLANN) and empirical models (Blaney–Criddle and Hargreaves) for daily EP estimation at Raipur, Jagdalpur and Ambikapur

Station	Models	Test performance			
		RMSE (mm/day)	R^2	EF	Rank
Raipur	Deep-LSTM-2	1.21	0.873	0.870	1
	MLANN-2	1.40	0.865	0.826	2
	Hargreaves	2.42	0.636	0.482	3
	Deep-LSTM-4	1.21	0.873	0.870	1
	MLANN-4	1.33	0.869	0.843	2
	Deep-LSTM-5	1.02	0.908	0.908	1
	MLANN-5	1.16	0.905	0.882	2
	Deep-LSTM-6	0.98	0.915	0.915	1
	MLANN-6	1.15	0.908	0.882	2
	Blaney–Criddle	1.99	0.725	0.649	3
Jagdalpur	Deep-LSTM-2	1.09	0.733	0.708	1
	MLANN-2	1.08	0.727	0.717	2
	Hargreaves	1.56	0.508	0.403	3
	Deep-LSTM-4	1.04	0.750	0.735	1
	MLANN-4	1.05	0.744	0.732	2
	Deep-LSTM-5	1.00	0.757	0.755	1
	MLANN-5	1.03	0.749	0.743	2
	Deep-LSTM-6	0.97	0.769	0.768	1
	MLANN-6	1.03	0.762	0.739	2
	Blaney–Criddle	1.27	0.605	0.604	3
Ambikapur	Deep-LSTM-2	1.07	0.670	0.526	1
	MLANN-2	1.12	0.671	0.481	2
	Hargreaves	1.27	0.418	0.388	4
	Deep-LSTM-4	0.99	0.712	0.596	1
	MLANN-4	1.00	0.702	0.585	2
	Deep-LSTM-5	0.94	0.699	0.635	1
	MLANN-5	0.99	0.701	0.598	2
	Deep-LSTM-6	0.93	0.716	0.638	1
	MLANN-6	0.96	0.709	0.617	2
	Blaney–Criddle	1.12	0.597	0.479	3

Jagdalpur and Ambikapur the improvement in RMSE values obtained with Deep-LSTM models is only slightly better as compared to MLANN models for all four input feature combinations. At Jagdalpur the RMSE values improved from 1.09 to 0.97 with Deep-LSTM model followed by 1.08 to 1.03 with MLANN models with increasing number of input features. Similar trend in RMSE values is observed in Ambikapur also where it improved from 1.07 to 0.93 for Deep-LSTM models followed by 1.12 to 0.96 with MLANN models as the number of input features increased.

With regard to R^2 and EF, both improved marginally with Deep-LSTM models over MLANN models; however, higher values of R^2 and EF ranging from 0.865 to 0.915 and 0.826 to 0.915, respectively, with increasing number of input combinations for neural network model (Deep-LSTM and MLANN) at Raipur are noticeable and

encouraging. A similar trend is observed at Jagdalpur also with a comparatively lower magnitude of R^2 and EF ranging from 0.727 to 0.769 and 0.708 to 0.768, respectively, with increasing number of input features for neural network models. Further, the magnitude of R^2 and EF at Ambikapur is even less as compared to other two stations. However, R^2 and EF improved from 0.670 to 0.716 and 0.481 to 0.638, respectively, for different neural network models with increasing number of input features. The performance of both Deep-LSTM and MLANN models is superior to empirical methods in terms of RMSE, R^2 and EF in all the three locations. Deep-LSTM model ranked top among the models as it performed better in all three objectives (RMSE, R^2 and EF) in most cases and at least two objectives in some cases under different scenarios of input combinations. The difference in magnitudes of RMSE values at Raipur, Jagdalpur and Ambikapur is

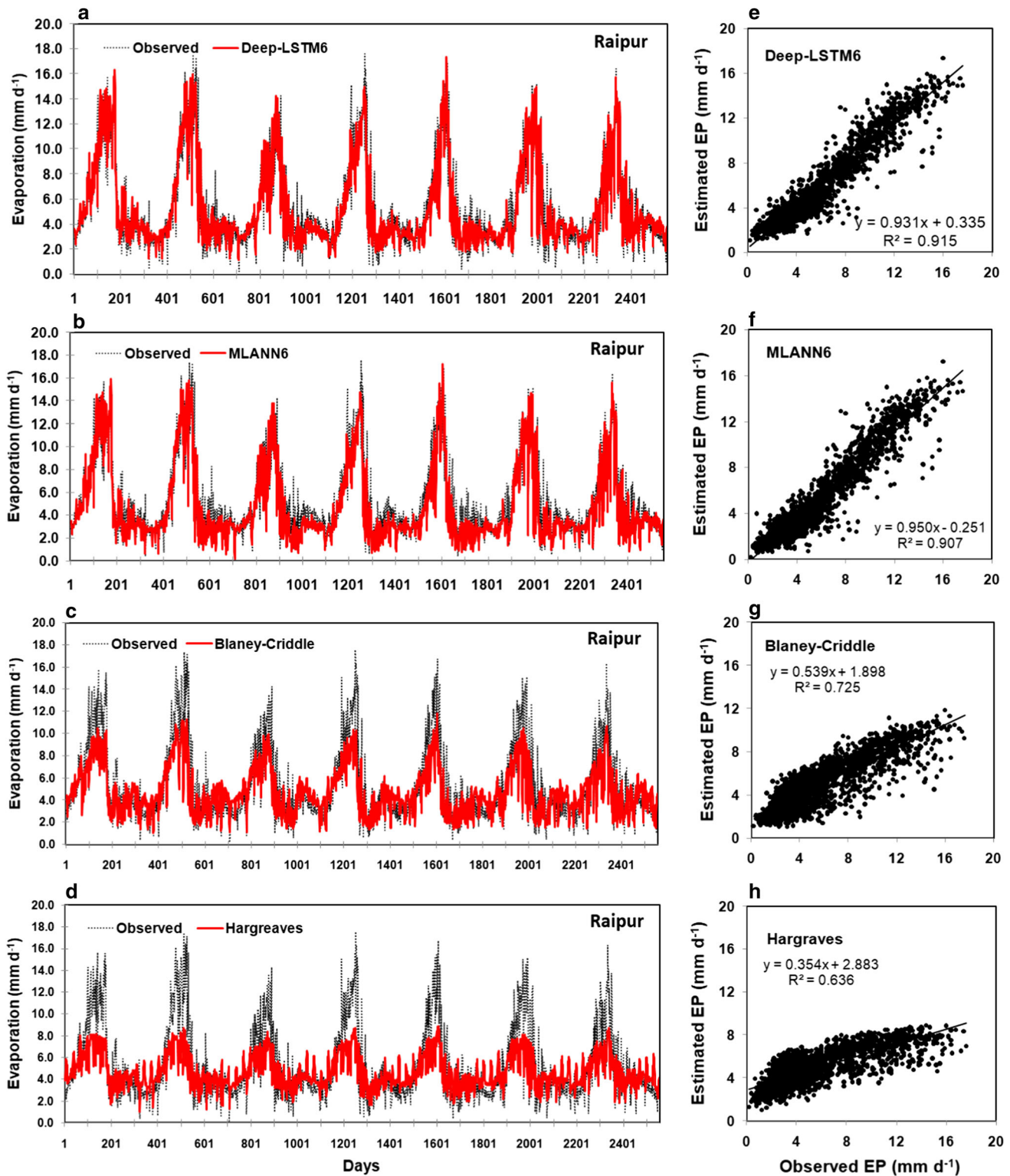


Fig. 3 a–h Comparison of observed and estimated daily EP and their relationship for Deep-LSTM-6, MLANN-6 and empirical models (Blaney-Cridde and Hargreaves) at Raipur

observed mainly because of difference in agro-climatic situation of the respective ACZ. The low R^2 and EF at Jagdalpur and Ambikapur as compared to Raipur may be

associated with the variation in correlation coefficient between climatic factors and EP in respective station. Availability of less number of input patterns for model

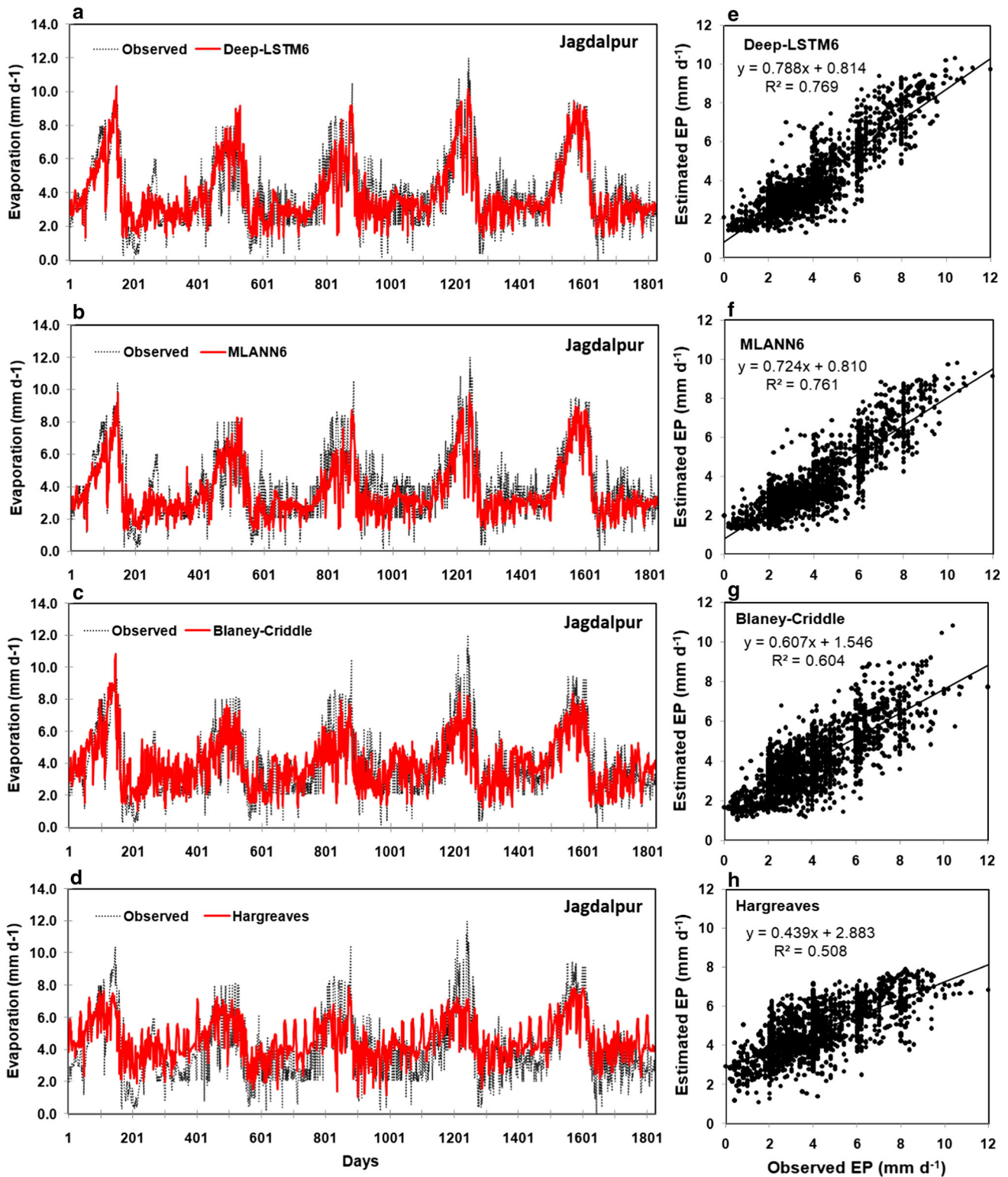


Fig. 4 a–h Comparison of observed and estimated daily *EP* values and their relationship for Deep-LSTM-6, MLANN-6 and empirical models (Blaney–Criddle and Hargreaves) at Jagdalpur

development may also be one of the reasons for poor predictive performance of the proposed models at Jagdalpur and Ambikapur as compared to Raipur.

Comparison between observed and predicted daily *EP* for Deep-LSTM-6, MLANN-6, Blaney–Criddle and Hargreaves models at Raipur, Jagdalpur and Ambikapur during

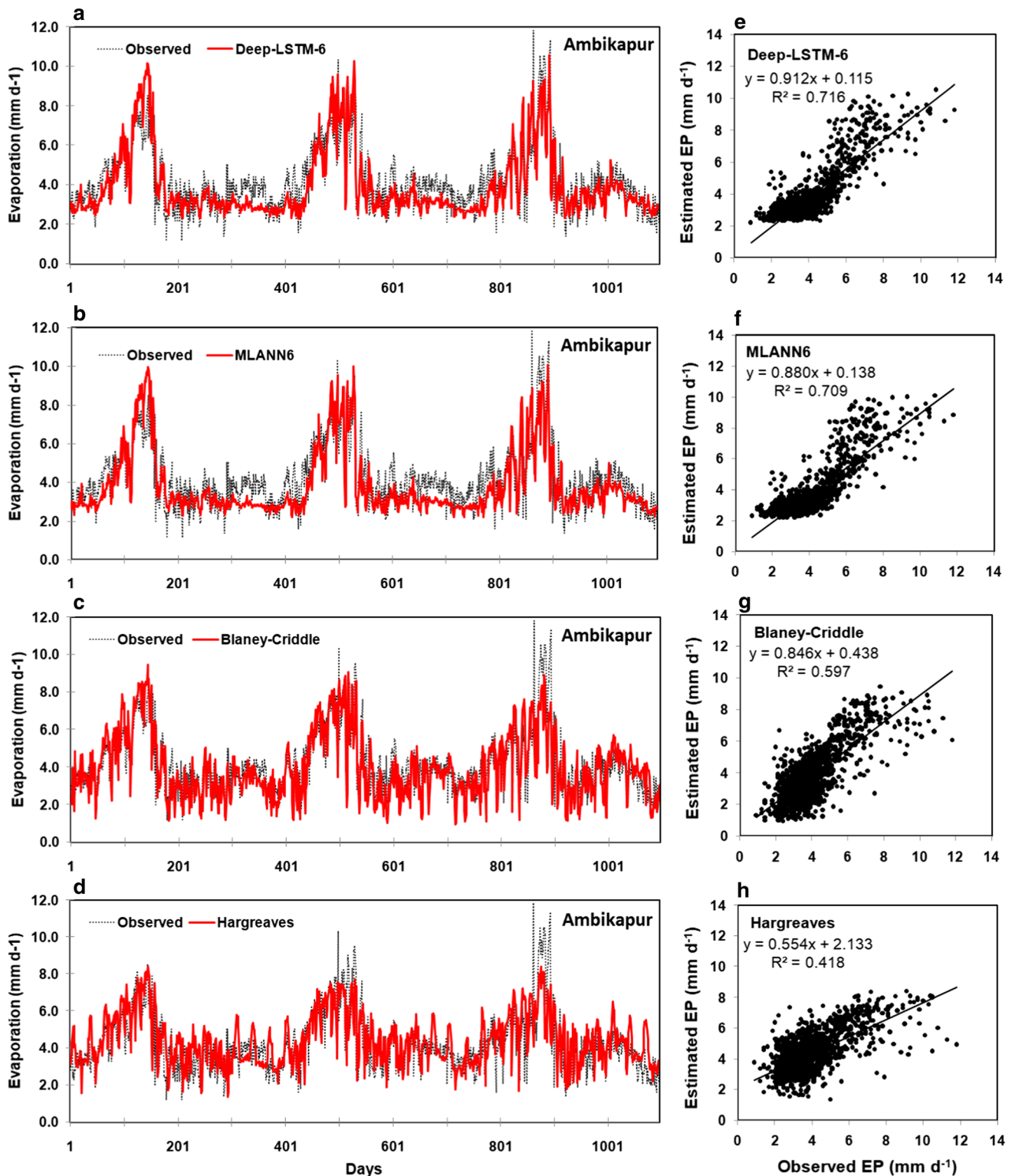


Fig. 5 a–h Comparison of observed and estimated daily EP values and their relationship for Deep-LSTM-6, MLANN-6 and empirical models (Blaney-Criddle and Hargreaves) at Ambikapur

the testing phase is shown in Figs. 3a–d, 4a–d and 5a–d, respectively. Relationship between observed and predicted values of daily EP for Deep-LSTM6, MLANN-6 and

Blaney-Criddle models at Raipur, Jagdalpur and Ambikapur is also shown in Figs. 3e–h, 4e–h and 5e–h, respectively. It is observed that, pictorially, it is difficult to

differentiate the performance of Deep-LSTM-6 and MLANN-6 on a daily scale; however, daily estimates obtained through Deep-LSTM-6 and MLANN-6 models seem to be in close agreement with the observed ones in most of the cases as compared to empirical models in all stations. The empirical models either underestimate (at Raipur) or overestimate (at Jagdalpur and Ambikapur) daily EP and are unable to predict peak evaporation rates during summer season. Further, smaller intercept values close to zero indicate that the proposed Deep-LSTM-6 estimate exhibits a close relationship with observed evaporation in most cases as compared to other models.

5 Statistical studies for model selection

For selection of appropriate regression models among the models under investigation, two statistical analyses, namely paired-*t* test and Akaike information criterion (AIC), have been conducted and the results obtained are discussed:

5.1 Paired *t* test

In order to further examine the performance of models under consideration a paired *t* test is conducted to test the null hypothesis that the pairwise difference between squared errors obtained for different models has a mean equal to zero or no significant difference exists in estimated outputs of compared models. The alternate hypothesis says that the difference among estimated outputs of compared model is statistically significant. The *t* test returns ‘*h*’ and ‘*p*’ values as result. The *h* value equivalent to 0 and a corresponding *p* value greater than 0.05 approves or, in other words, fail to reject the null hypotheses, which indicates that statistically no significant difference exists between the mean squared error obtained with Deep-LSTM and compared models, whereas *h* = 1 and *p* < 0.05 approve the rejection of null hypotheses at the 95% significance level. This indicates the fact that there exists a significant difference between the estimated output of

compared models. Comparative paired *t* test statistics (*p* and *h* values) for Deep-LSTM with equivalent (in terms of the number of input features) MLANN and empirical models are shown in Table 7. It is observed that in most of the cases *h* value of 1 and *p* value less than 0.05 approve the fact that Deep-LSTM-produced estimated outputs are significantly different than equivalent model. However, *h* value of 0 and *p* value greater than 0.05 at Jagdalpur indicate that it is not possible to prove that there exists any significant difference between Deep-LSTM-2 and MLANN-2 predictions. Similarly, *h* = 0 and *p* greater than 0.05 at Jagdalpur and Ambikapur indicate there is hardly any difference between the performance of Deep-LSTM-4 and MLANN-4 at these locations.

5.2 Akaike information criterion (AIC)

The AIC is widely used for model selection for regression problems [56, 57]. The AIC values are computed with the help of mean squared error (MSE) between observed and estimated evaporations for each model and using following Eq. (13).

$$AIC = N * \log(MSE) + 2 * k \tag{13}$$

where *N* = number of observations and *k* = number of parameters.

The AIC values for Deep-LSTM and corresponding MLANN and empirical models are shown in Table 8. The lower AIC values represent the better models. It is also seen that in all cases AIC values obtained with Deep-LSTM models are lower as compared to other models. The difference in magnitude for different data sets, i.e., Raipur, Jagdalpur and Ambikapur, is due to the difference in the number of observations considered for testing the models.

6 Conclusion

This study is carried out to assess the potentiality of Deep-LSTM structure for estimation of daily EP losses under different agro-climatic situations with the help of climatic

Table 7 Pair *t* test statistics for Deep-LSTM with corresponding MLANN and empirical models for different data sets

Models	Data sets					
	Raipur		Jagdalpur		Ambikapur	
	<i>h</i>	<i>p</i>	<i>h</i>	<i>p</i>	<i>h</i>	<i>p</i>
Deep-LSTM-2/MLANN-2	1	2.085e−45	0	0.162	1	0.002
Deep-LSTM-2/Hargreaves	1	6.372e−70	1	3.970e−63	1	1.298e−06
Deep-LSTM-4/MLANN-4	1	1.047e−33	0	0.533	0	0.093
Deep-LSTM-5/MLANN-5	1	7.917e−36	1	0.004	1	4.202e−35
Deep-LSTM-6/MLANN-6	1	9.300e−49	1	1.065e−11	1	6.634e−06
Deep-LSTM-6/Blaney–Criddle	1	1.775e−78	1	1.355e−42	1	1.846e−09

Table 8 AIC for Deep-LSTM with corresponding MLANN and empirical models for different data sets

Models	Data sets		
	Raipur	Jagdapur	Ambikapur
Deep-LSTM-2	9174	6079	3924
MLANN-2	9470	6098	3972
Hargreaves	10,681	6674	4116
Deep-LSTM-4	9147	6024	3841
MLANN-4	9359	6037	3856
Deep-LSTM-5	8771	6004	3788
MLANN-5	9051		3840
Deep-LSTM-6	8687	5923	3785
MLANN-6	9048	6020	3816
Blaney–Criddle	10,258	9316	3983

data influencing the evaporation process. The investigation has led to the following conclusions:

- Both Deep-LSTM and MLANN models are capable of estimating the daily EP with different input combinations.
- Deep-LSTM models performed better compared to MLANN and empirical models in all scenarios.
- Statistical inferences based on paired *t* test and AIC also suggest that the Deep-LSTM models are superior to the MLANN and empirical models for different input combinations.
- Depending on the availability of the climatic data appropriate Deep-LSTM model can be adopted for estimating daily EP for the stations where direct measurement of evaporation is not done. In future other deep learning-based neural network structures may be applied to predict the nonlinear processes such as evaporation and reference evapotranspiration.

Compliance with ethical standards

Conflict of interest We declare that we have no conflict of interests.

References

1. Abtew W, Melesse A (2013) Evaporation and evapotranspiration: measurements and estimations. Springer, Netherlands, pp 1–206. <https://doi.org/10.1007/978-94-007-4737-1>
2. Yao H (2009) Long-term study of lake evaporation and evaluation of seven estimation methods: results from Dickie Lake, South-Central Ontario, Canada. *J Water Resour Prot* 01(02):59–77. <https://doi.org/10.4236/jwarp.2009.12010>
3. Martí P, González-Altozano P, López-Urrea R, Mancha LA, Shiri J (2015) Modeling reference evapotranspiration with calculated targets. Assessment and implications. *Agric Water Manag* 149:81–90. <https://doi.org/10.1016/j.agwat.2014.10.028>
4. Singh VP, Xu C-Y (1997) Evaluation and generalization of 13 mass transfer equations for determining free water evaporation. *Hydrol Process* 11(3):311–323. [https://doi.org/10.1002/\(SICI\)1099-1085\(19970315\)11:3%3c311:AID-HYP446%3e3.3.CO;2-P](https://doi.org/10.1002/(SICI)1099-1085(19970315)11:3%3c311:AID-HYP446%3e3.3.CO;2-P)
5. Xu CY, Singh VP (2000) Evaluation and generalization of radiation-based methods for calculating evaporation. *Hydrol Process* 14:339–349
6. Xu CY, Singh VP (2001) Evaluation and generalization of temperature-based methods for calculating evaporation. *Hydrol Process* 15:305–319
7. Gianniou SK, Antonopoulos VZ (2007) Evaporation and energy budget in Lake Vegoritis, Greece. *J Hydrol* 345:212–223
8. Rosenberry DO, Winter TC, Buso DC, Likens GE (2007) Comparison of 15 evaporation methods applied to a small mountain lake in the northeastern USA. *J Hydrol* 340:149–166. <https://doi.org/10.1016/j.jhydrol.2007.03.018>
9. Ali S, Ghosh NC, Singh R (2008) Evaluating best evaporation estimate model for water surface evaporation in semi-arid region, India. *Hydrol Processes* 22:1093–1106
10. Allen RG, Pereira LS, Raes D, Smith M (1998) Crop evapotranspiration. Guidelines for computing crop water requirements. Irrigation and drainage. Paper 56, FAO, Rome, p 300
11. Abudu S, Cui C, King JP, Moreno J, Bawazir AS (2011) Modeling of daily pan evaporation using partial least squares regression. *Sci China Technol Sci* 54(1):163–174. <https://doi.org/10.1007/s11431-010-4205-z>
12. Benzaghta MA, Mohammed TA, Ghazali AH, Soom MAM (2012) Validation of selected models for evaporation estimation from reservoirs located in arid and semi-arid regions. *Arab J Sci Eng* 37(3):521–534. <https://doi.org/10.1007/s13369-012-0194-5>
13. Guven A, Kisi O (2011) Daily pan evaporation modeling using linear genetic programming technique. *IrrigSci* 29:135–145
14. Guven A, Kisi O (2013) Monthly pan evaporation modeling using linear genetic programming. *J Hydrol* 503:178–185. <https://doi.org/10.1016/j.jhydrol.2013.08.043>
15. Keskin ME, Terzi Ö, Taylan D (2009) Estimating daily pan evaporation using adaptive neural-based fuzzy inference system. *Theor Appl Climatol* 98(1–2):79–87. <https://doi.org/10.1007/s00704-008-0092-7>
16. Kim S, Shiri J, Kisi O (2012) Pan evaporation modeling using neural computing approach for different climatic zones. *Water Resour Manag* 26(11):3231–3249. <https://doi.org/10.1007/s11269-012-0069-2>
17. Kim S, Singh VP, Seo Y (2014) Evaluation of pan evaporation modeling with two different neural networks and weather station data. *Theor Appl Climatol* 117(1):1–13. <https://doi.org/10.1007/s00704-013-0985-y>
18. Kişi Ö (2009) Modeling monthly evaporation using two different neural computing techniques. *Irrig Sci* 27(5):417–430. <https://doi.org/10.1007/s00271-009-0158-z>
19. Malik A, Kumar A (2015) Pan evaporation simulation based on daily meteorological data using soft computing techniques and multiple linear regression. *Water Resour Manag* 29(6):1859–1872. <https://doi.org/10.1007/s11269-015-0915-0>
20. Shirsath PB, Singh AK (2010) A comparative study of daily pan evaporation estimation using ANN, regression and climate based models. *Water Resour Manag* 24(8):1571–1581. <https://doi.org/10.1007/s11269-009-9514-2>
21. Sanikhani H, Kisi O, Nikpour MR, Dinpashoh Y (2012) Estimation of daily pan evaporation using two different adaptive neuro-fuzzy computing techniques. *Water Resour Manag* 26(15):4347–4365. <https://doi.org/10.1007/s11269-012-0148-4>

22. Tabari H, Marofi S, Sabziparvar AA (2010) Estimation of daily pan evaporation using artificial neural network and multivariate non-linear regression. *Irrig Sci* 28(5):399–406. <https://doi.org/10.1007/s00271-009-0201-0>
23. Deo RC, Samui P, Kim D (2016) Estimation of monthly evaporative loss using relevance vector machine, extreme learning machine and multivariate adaptive regression spline models. *Stoch Environ Res Risk Assess* 30(6):1769–1784. <https://doi.org/10.1007/s00477-015-1153-y>
24. Wang L, Kisi O, Zounemat-Kermani M, Li H (2017) Pan evaporation modeling using six different heuristic computing methods in different climates of China. *J Hydrol* 544:407–427. <https://doi.org/10.1016/j.jhydrol.2016.11.059>
25. Wang L, Niu Z, Kisi O, Li C, Yu D (2017) Pan evaporation modeling using four different heuristic approaches. *Comput Electron Agric* 140:203–213. <https://doi.org/10.1016/j.compag.2017.05.036>
26. Malik A, Kumar A, Kisi O (2017) Monthly pan-evaporation estimation in Indian central Himalayas using different heuristic approaches and climate based models. *Comput Electron Agric* 143:302–313. <https://doi.org/10.1016/j.compag.2017.11.008>
27. Tezel G, Buyukyildiz M (2016) Monthly evaporation forecasting using artificial neural networks and support vector machines. *Theor Appl Climatol* 124(1–2):69–80. <https://doi.org/10.1007/s00704-015-1392-3>
28. Kisi O, Genc O, Dinc S, Zounemat-Kermani M (2016) Daily pan evaporation modeling using Chi squared automatic interaction detector, neural networks, classification and regression tree. *Comput Electron Agric* 122:112–117. <https://doi.org/10.1016/j.compag.2016.01.026>
29. Keshtegar B, Piri J, Kisi O (2016) A nonlinear mathematical modeling of daily pan evaporation based on conjugate gradient method. *Comput Electron Agric* 127:120–130. <https://doi.org/10.1016/j.compag.2016.05.018>
30. Martí P, González-Altozano P, López-Urrea R, Mancha LA, Shiri J (2015) Modeling reference evapotranspiration with calculated targets. Assessment and implications. *Agric Water Manag* 149:81–90. <https://doi.org/10.1016/j.agwat.2014.10.028>
31. Panda B, Majhi B (2018) A novel improved prediction of protein structural class using deep recurrent neural network. *Evol Intell*. <https://doi.org/10.1007/s12065-018-0171-3>
32. Abd-Elazim SM, Ali ES (2016) Load frequency controller design of a two-area system composing of PV grid and thermal generator via firefly algorithm. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-016-2668-y>
33. Abd-Elazim SM, Ali ES (2016) Load frequency controller design via BAT algorithm for nonlinear interconnected power system. *Electr Power Energy Syst* 77:166–177. <https://doi.org/10.1016/j.ijepes.2015.11.029>
34. Ali ES, AbdElazim SM, Abdelaziz AY (2016) Ant lion optimization algorithm for renewable distributed generations. *Energy* 116:445–458. <https://doi.org/10.1016/j.energy.2016.09.104>
35. Arqub OA, Abo-Hammour Z (2014) Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. *Inf Sci* 279:396–415. <https://doi.org/10.1016/j.ins.2014.03.128>
36. Arqub OA, Al-Smadi M, Momani S, Hayat T (2017) Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems. *Soft Comput* 21(23):7191–7206. <https://doi.org/10.1007/s00500-016-2262-3>
37. Arqub OA, Mohamed A-S, Momani S, Hayat T (2016) Numerical solutions of fuzzy differential equations using reproducing kernel Hilbert space method. *Soft Comput* 20(8):3283–3302. <https://doi.org/10.1007/s00500-015-1707-4>
38. Abu Arqub O (2017) Adaptation of reproducing kernel algorithm for solving fuzzy Fredholm-Volterra integro-differential equations. *Neural Comput Appl* 28(7):1591–1610. <https://doi.org/10.1007/s00521-015-2110-x>
39. Zhao Z, Chen W, Wu X, Chen PCY, Liu J (2017) LSTM network: a deep learning approach for short-term traffic forecast. *IET Intell Transp Syst* 11(2):68–75. <https://doi.org/10.1049/iet-its.2016.0208>
40. Greff K, Srivastava RK, Koutnik J, Steunebrink BR, Schmidhuber J (2017) LSTM: a search space odyssey. *IEEE Trans Neural Netw Learn Syst* 28(10):2222–2232. <https://doi.org/10.1109/TNNLS.2016.2582924>
41. Xie Y, Le L, Zhou Y, Raghavan VV (2018) Deep learning for natural language processing. In: *Handbook of statistics* (vol 38, pp 317–328). Elsevier B.V. <https://doi.org/10.1016/bs.host.2018.05.001>
42. Alipanahi B, Delong A, Weirauch MT, Frey BJ (2015) Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat Biotechnol* 33(8):831–838. <https://doi.org/10.1038/nbt.3300>
43. Dalto M (2015) Deep neural networks for time series prediction with applications in ultra-short-term wind forecasting. In: *Proceedings of IEEE international conference on industrial technology (ICIT)*, 2015, 1657–1663. <https://doi.org/10.1109/ICIT.2015.7125335>
44. Fischer T, Krauss C (2018) Deep learning with long short-term memory networks for financial market predictions. *Eur J Oper Res* 270(2):654–669. <https://doi.org/10.1016/j.ejor.2017.11.054>
45. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
46. Haykin S (1998) *Neural networks: a comprehensive foundation*, 2nd edn. Prentice-Hall, Upper Saddle River, pp 26–32
47. Hargreaves GL, Hargreaves GH, Riley JP (1985) Irrigation water requirements for Senegal River Basin. *J Irrig Drain Eng* 111(3):265–275. [https://doi.org/10.1061/\(ASCE\)0733-9437\(1985\)111:3\(265\)](https://doi.org/10.1061/(ASCE)0733-9437(1985)111:3(265))
48. Doorenbos J, Pruitt WO (1977) Guidelines for predicting crop water requirements. FAO irrigation and drainage paper no. 24. Food and Agriculture Organization of the United Nations, Rome, 15–29, 112–115. <https://doi.org/10.1161/CIRCULATIONAHA.105.601930>
49. Nash JE, Sutcliffe JV (1970) River flow forecasting through conceptual models part I: a discussion of principles. *J Hydrol* 10(3):282–290. [https://doi.org/10.1016/0022-1694\(70\)90255-6](https://doi.org/10.1016/0022-1694(70)90255-6)
50. Kingma DP, Ba JL (2014) Adam optimizer. *ArXiv Preprint arXiv:1412.6980*, 1–15. <https://doi.org/10.1145/1830483.1830503>
51. Gers FA, Schraudolph NN, Schmidhuber J (2003) Learning precise timing with LSTM recurrent networks. *J Mach Learn Res* 3(1):115–143. <https://doi.org/10.1162/153244303768966139>
52. Graves A, Fernández S, Schmidhuber J (2005) Bidirectional LSTM networks for improved phoneme classification and recognition. In: *Proceedings of international conference on artificial neural networks*, pp 799–804. https://doi.org/10.1007/11550907_126
53. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Zheng X (2016) TensorFlow: a system for large-scale machine learning. In: *Proceedings of 12th USENIX conference on operating systems design and implementation*, 272–283. <https://doi.org/10.1126/science.aab4113.4>
54. Chollet F (2015) Keras: deep Learning library for Theano and TensorFlow. GitHub Repository, 1–21

55. Pedregosa F, Weiss R, Brucher M (2011) Scikit-learn : machine learning in python. *J Mach Learn Res* 12(October):2825–2830. <https://doi.org/10.1016/j.molcel.2012.08.019>
56. Symonds MRE, Moussalli A (2011) A brief guide to model selection, multimodel inference and model averaging in behavioural ecology using Akaike's information criterion. *Behav Ecol Sociobiol.* <https://doi.org/10.1007/s00265-010-1037-6>
57. Penny WD (2012) Comparing dynamic causal models using AIC. *BIC Free Energy NeuroImage* 59(1):319–330. <https://doi.org/10.1016/j.neuroimage.2011.07.039>

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