Alexandria Engineering Journal 86 (2024) 425–442

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Boruta extra tree-bidirectional long short-term memory model development for Pan evaporation forecasting: Investigation of arid climate condition

Masoud Karbasi [a,b,*], Mumtaz Ali [c,j], Sayed M. Bateni [d], Changhyun Jun [e,**], Mehdi Jamei [a,f,g], Zaher Mundher Yaseen [h,i]

[a] Canadian Centre for Climate Change and Adaptation, University of Prince Edward Island, St Peters, PE, Canada
[b] Water Engineering Department, Faculty of Agriculture, University of Zanjan, Zanjan, Iran
[c] UniSQ College, University of Southern Queensland, QLD 4305, Australia
[d] Dept. of Civil and Environmental Engineering and Water Resources Research Center, Univ. of Hawaii at Manoa, 2540 Dole St., Holmes 342, Honolulu, HI 96822, USA
[e] Department of Civil and Environmental Engineering, College of Engineering, Chung-Ang University, Seoul, South Korea
[f] Department of Civil Engineering, Faculty of Civil Engineering and Architecture, Shahid Chamran University of Ahvaz, Ahvaz, Iran
[g] New Era and Development in Civil Engineering Research Group, Scientific Research Center, Al-Ayen University, Thi-Qar, Nasiriyah 64001, Iraq
[h] Civil and Environmental Engineering Department, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia
[i] Interdisciplinary Research Center for Membranes and Water Security, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia
[j] Faculty of Sustainable Design Engineering, University of Prince Edward Island, Charlottetown, PE C1A4P3, Canada

ARTICLE INFO

Keywords:
Pan Evaporation
Forecasting
Machine Learning
Deep Learning
Boruta Feature Selection

ABSTRACT

In this study, two deep learning approaches, bidirectional long short-term memory (BiLSTM) and long short-term memory (LSTM), were used along with adaptive boosting and general regression neural network to forecast multi-step-ahead pan evaporation in two arid climate stations in Iran (Ahvaz and Yazd). Lagged time series of meteorological data and pan evaporation data were input to the machine learning models. Two feature selection methods, i.e., the Boruta extra tree and XGBoost, were used to select significant inputs to reduce the number of inputs and model complexity. Different statistical metrics were used to investigate the model performance. The results demonstrated that Boruta-extra-tree-based models were more accurate than XGBoost-based models. Compared with the machine learning techniques, the combination of Boruta extra tree and BiLSTM enabled more accurate one-day-ahead forecasting of pan evaporation for both sites (Root Mean Square Error (RMSE) = 1.6857, for the Ahvaz station, and RMSE = 1.3996 for the Yazd station). The proposed model was used to forecast up to 30 days ahead of pan evaporation in both stations. The results showed that the Boruta-BiLSTM model could accurately forecast the pan evaporation for up to 30 days in both stations.

1. Introduction

Evaporation is a unique natural phenomenon that is a key component of the hydrological cycle and influences the agricultural output and availability of water for human use [1,2]. Pan evaporation is a measurement that integrates the impacts of several climatic factors, including temperature, humidity, precipitation, drought dispersion, sun radiation, and wind. Measurements of pan evaporation allow farmers and ranchers to determine how much water their crops will require [3]. Pan evaporation (PE_m) has been highlighted as an intricate aspect in the water cycle [4–6]. Therefore, many researchers have attempted to identify precise and reliable techniques for modeling PE_m rates to ensure the efficient management of water resources and accurately reflect the water balance in natural and artificial water bodies [7]. The different approaches for forecasting/predicting PE_m, have been reviewed [8–10]. Furthermore, numerous scholars have predicted the rate of evaporation using empirical approaches [11]. Although these models are frequently employed for geographical location-specific PE_m prediction and calculation [12,13], they require a significant amount of meteorological data as input. Consequently, these strategies are challenging to implement

* Corresponding author at: Canadian Centre for Climate Change and Adaptation, University of Prince Edward Island, St Peters, PE, Canada.
** Corresponding author.
E-mail addresses: m.karbasi@znu.ac.ir (M. Karbasi), cjun@cau.ac.kr (C. Jun).

https://doi.org/10.1016/j.aej.2023.11.061
Received 3 July 2023; Received in revised form 20 September 2023; Accepted 22 November 2023
Available online 7 December 2023

Table 1
Descriptive statistics of meteorological variables for Ahvaz and Yazd stations.

			Ahvaz			
Variable	Tmax (°C)	Tmin (°C)	Wind Speed (m/s)	RH (%)	Solar Radiation (MJ/m^2)	Pan Evaporation (mm/d)
Mean	34.19	19.79	2.22	42.13	22.01	8.12
Standard deviation	10.68	8.59	1.14	19.79	8.19	5.21
Max	53.70	36.40	8.13	96.00	34.37	20.90
Min	9.80	-1.20	0.00	6.00	4.72	0.60
First quartile Q1	24.33	12.01	1.38	25.50	15.59	3.30
Second quartile Q2	35.05	20.10	2.00	38.13	22.85	7.50
Third quartile Q3	44.60	28.00	2.88	55.88	29.64	12.10
Skewness	-0.18	-0.13	0.83	0.61	-0.33	0.43
Kurtosis	-1.35	-1.27	0.91	-0.55	-1.02	-0.89
		Yazd				
Variable	Tmax (°C)	Tmin (°C)	Wind Speed (m/s)	RH (%)	Solar Radiation (MJ/m^2)	Pan Evaporation (mm/d)
Mean	27.92	14.20	2.53	25.61	20.96	9.27
Standard deviation	9.77	9.10	0.96	17.67	6.68	4.80
Max	45.60	33.10	8.25	97.00	31.77	19.80
Min	-3.60	-10.10	0.25	4.00	4.65	1.60
First quartile Q1	19.80	6.30	1.88	12.50	15.22	5.15
Second quartile Q2	28.60	14.90	2.38	20.13	21.38	8.60
Third quartile Q3	36.60	22.10	3.13	33.13	27.25	13.50
Skewness	-0.27	-0.15	0.85	1.45	-0.26	0.24
Kurtosis	-1.00	-1.13	1.66	1.93	-0.99	-1.15

because meteorological data for certain regions may not be easily accessible [14]. This is because PE_m process depends on several types of climate "hydrometeorological data" and ground observations data. Furthermore, the derivation of a comprehensive yet simple formula that incorporates all of the physical processes is challenging owing to the complex, nonlinear, and unstable nature of evaporation [15]. Therefore, it is necessary to establish flexible and user-friendly evaporation prediction methodologies; despite many researchers have attempted to model PE_m using empirical and semi-empirical methods based on diverse meteorological data [16–18].

In comparison with traditional parametric methods, soft computing algorithms based on machine learning (ML), which do not require prior knowledge of the fundamental physical processes behind evaporation or a significant amount of data, are promising alternatives for real-time decision support systems [6,19,20]. In recent decades, ML models have been extensively employed in the domain of water resources research and noted to be effective for forecasting evaporation [21–24]. The primary benefit of ML models is that they are entirely non-parametric and do not require any prior understanding of the associations between the input variables and output data [25,26]. ML models have been widely used for predicting/forecasting evaporation processes, especially in ecological and water engineering applications [27–31], because these models are easy to use, reliable, and adept at handling challenging nonlinear processes [32].

Despite the satisfactory predictive performance of conventional ML algorithms [6], certain limitations exist when advanced techniques such as data pre-processing algorithms, feature input selection, and boosted learning mechanisms are to be integrated. Furthermore, the existing techniques cannot consider the influence of parameter interactions on the model performance. To address these problems, evolutionary algorithms have been used to modify the internal parameters of ML models. Specifically, numerous researchers have tried to resolve the artificial neural network training problem using a combination of different model optimization strategies. For more accurate evaporation prediction, several researchers have merged nature-inspired algorithms with cutting-edge neural techniques [33,34], shifting the research trend toward highly reliable and robust artificial intelligence (AI) models.

Given the potential of neural networks, deep learning (DL) techniques have attracted significant attention [35]. The effectiveness of these approaches has been enhanced by developments in computer technologies, particularly graphic processing units [36], and the ease of accessibility of enormous data sets [36–38]. The main merit of those techniques is the ability to configure the learning process in a more robust manner due to the multiple layers of the network. Worth to mention, to the authors knowledge, the application of DL algorithms for PE_m prediction has been rarely reported in the literature. DL frameworks have a remarkable capacity to learn, especially time-series data, because of the presence of loops that allow information from the previous time slice to be sent to the subsequent time slice [39]. Nevertheless, recurrent neural networks (RNNs) as a representative DL technique encounter gradient dissipation and explosion, which deteriorates their capacity to learn long-distance information [40]. To address this problem, long short-term memory (LSTM) networks were developed, which can learn long-term dependencies in the data series and address the issue of vanishing gradients [41]. LSTM methods have been widely used in the hydrology and water management fields [42–44]. For example, streamflow modeling [45], precipitation-runoff [46], rainfall modeling [47], air temperature monitoring [48], and solar prediction [49]. According to research on the robustness of different neural network topologies for simulating and predicting the water levels of combined structures, in the case of multi-step-ahead predictions, LSTM (in conjunction with a cell memory recurrent neural network design) is preferred over conventional architectures that lack explicit cell memories [50]. Zhang et al. [51] used the LSTM to forecast water tables in agricultural regions. Moreover, the effectiveness of AI models such as LSTM for river basin flow prediction in India was compared to those of traditional and naive methods [43]. An LSTM model was used to forecast the discharge at the Hoa Binn station in Vietnam [52]. The authors assessed the model's effectiveness in forecasting flood flows and flood peaks and clarified the effect of dataset attributes on the model performance. Nevertheless, to the best of our knowledge, none of the existing studies have used long short-term approaches for forecasting PE_m. Given the widespread use of the LSTM model to explore hydrological processes [42–44], these models were adopted in this study for forecasting PE_m.

Specifically, in this study, a unique DL technique, i.e., bidirectional LSTM (BiLSTM), was developed to forecast daily evaporation in conjunction with two robust feature selection algorithms: Boruta extra tree and extreme gradient boosting (XGBoost). The main inspiration for using the BiLSTM over the other models is the capability to separate memory cells that can store long-term information without being affected by the current input or output. In contrast, the main feasibility of the Boruta feature selection algorithm is working based on statistical ground and works extremely well even without any specific input by the user. The Ahvaz and Yazd stations (with arid climates) in Iran were selected as test sites to evaluate the proposed technique. Four ML algorithms, BiLSTM, LSTM, adaptive boosting (AdaBoost), and general

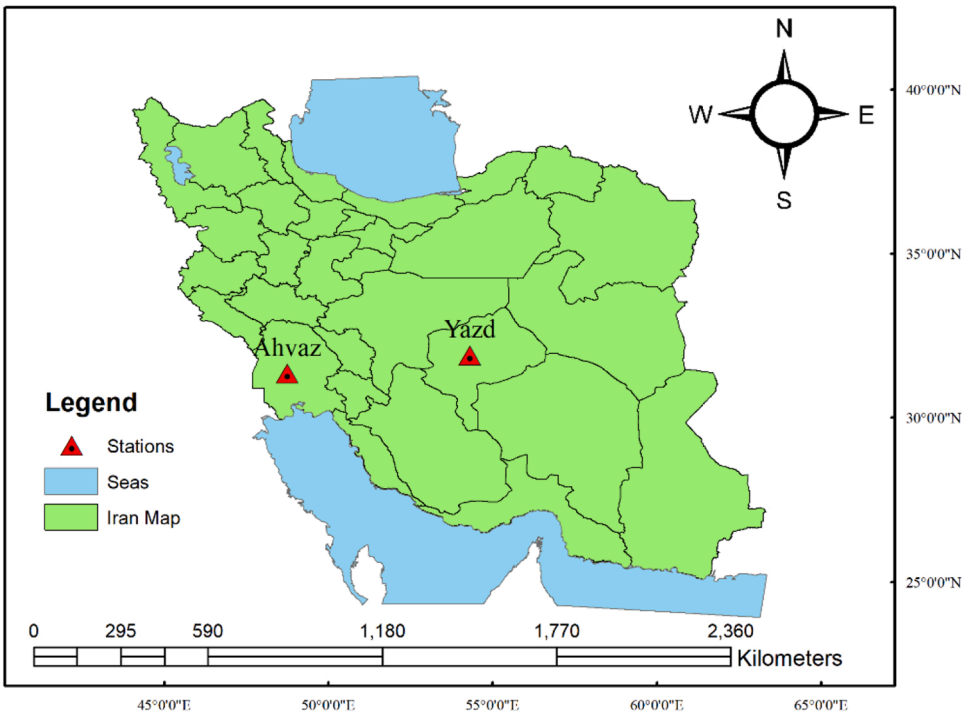

Fig. 1. Locations of Ahvaz and Yazd stations.

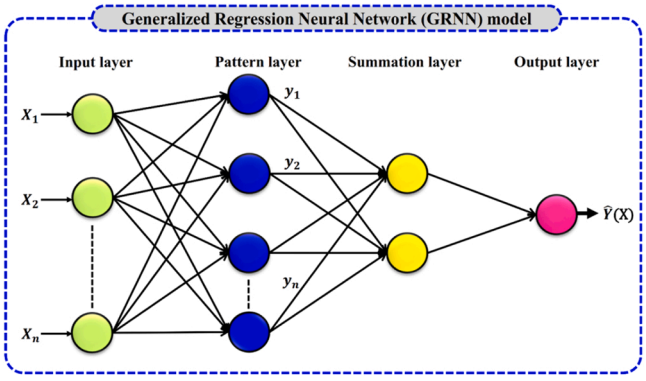

Fig. 2. Topology of the architecture of the GRNN model.

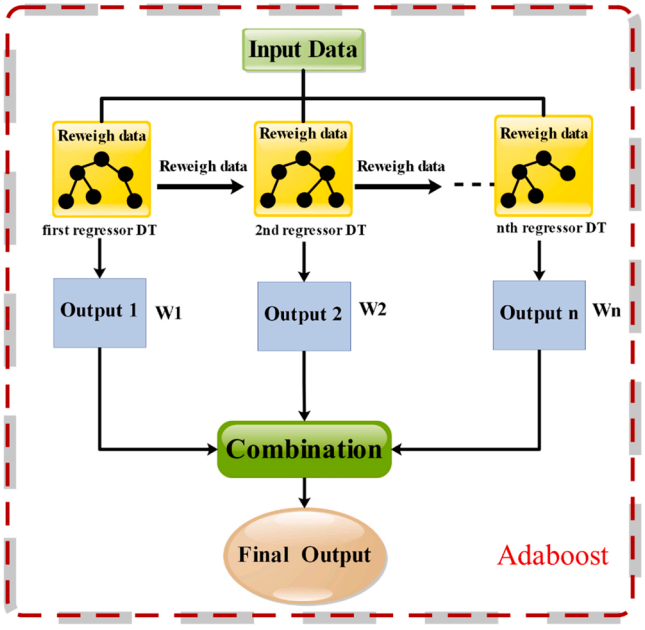

Fig. 3. Schematic of AdaBoost approach.

regression neural network (GRNN), were applied to evaluate the performance of the proposed model. Thus, the objectives of this study were to (1) forecast one-day-ahead evaporation rates using feature selection techniques and the BiLSTM model, (2) compare the effectiveness of the ML models, and (3) forecast multi-step-ahead daily evaporation with the optimal model.

2. Material and methods

2.1. Study area and data

The Ahvaz and Yazd stations were selected to forecast multi-step-ahead daily pan evaporation. The climate in Ahvaz pertains to a subtropical hot desert (Köppen climatic classification BWh), with long, hot summers and short, chilly winters. Summer temperatures are typically at least 45 °C and can occasionally approach 50 °C, with frequent sandstorms and dust storms. During the winter, the lowest temperature might decrease to approximately 5 °C. Ahvaz experiences snowless winters. The average yearly precipitation is approximately 230 mm. Similar to Ahvaz, Yazd has a hot desert climate (Köppen climatic classification BWh). With a yearly average of only 49 millimeters and only 23 days of rain, it is the driest significant city in Iran. Summer temperatures regularly exceed 40 °C, and there is no humidity.

The daily meteorological data, including the minimum and maximum temperature (Tmin, Tmax), wind speed, relative humidity (RH%), sun radiation, and pan evaporation, were provided by the Iran Meteorology Organization. Table 1 presents the descriptive statistics of these meteorological variables and pan evaporation for the Ahvaz and Yazd stations. The mean daily pan evaporation values for the Ahvaz and Yazd stations can be noted to be 8.12 mm and 9.27 mm, respectively. Maximum values of pan evaporation for Ahvaz (20.9 mm/day) and Yazd

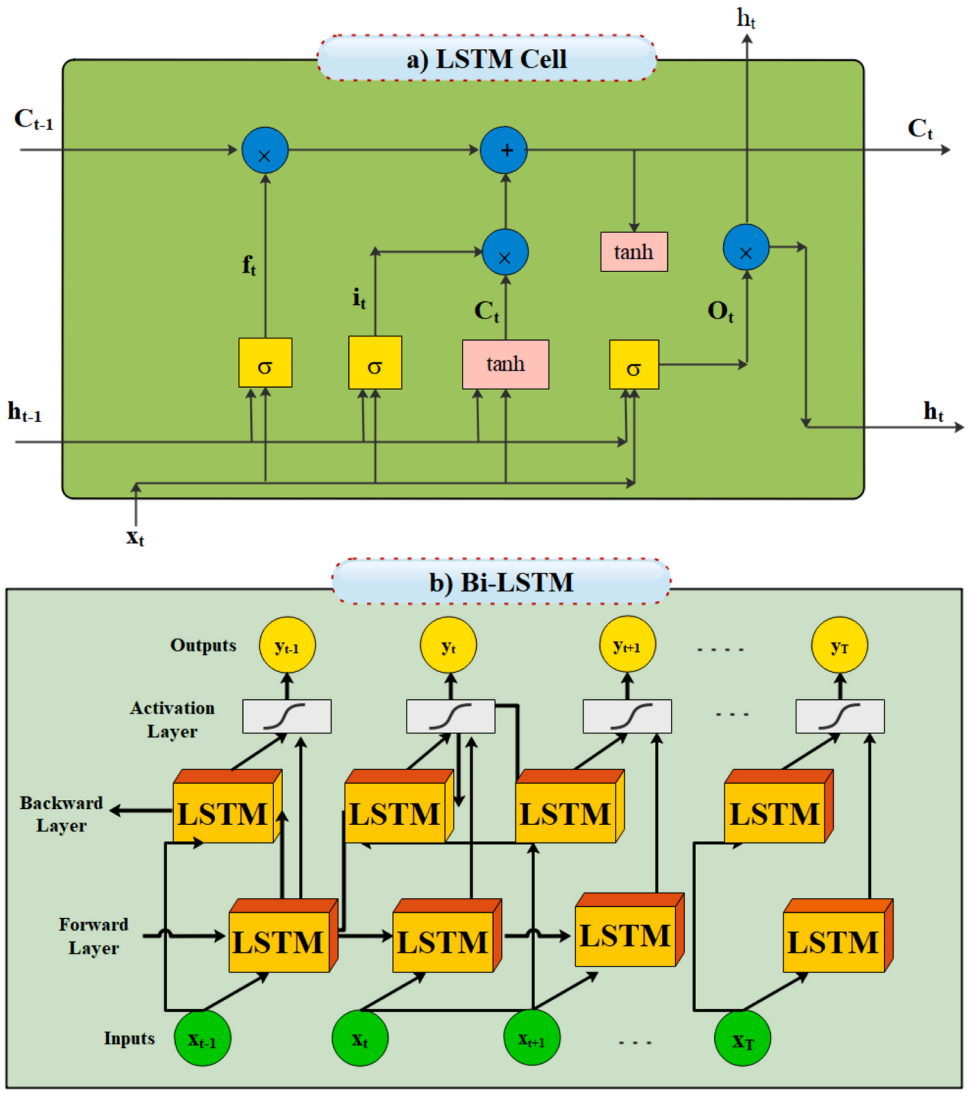

Fig. 4. Structure of LSTM cell (a) and BiLSTM model (b).

(19.8 mm/day) stations occurred in July and June, respectively. Minimum values of pan evaporation occurred in February at both stations.

2.2. Feature selection algorithms

2.2.1. Boruta extra tree

Feature selection is frequently a crucial step in the execution of ML algorithms to optimally select the input features to feed the ML models. The Boruta package, an extension of the random forest classification algorithm, is a popular wrapper feature selection method that has recently been used in hybrid ML-based models applied for engineering problems. The random forest classification method numerically evaluates the feature significance and can be executed with the default settings [53]. A benchmark criterion, the Z-score, is implemented in this scheme, which refers to the ratio of the average loss to the standard deviation. Using Python libraries, advanced stochastic decision tree methods can be used in the Boruta package [53]. One of these methods is the extra tree ensemble, which, combined with the Boruta package, yields the Boruta extra tree feature selection technique.

Extra tree is a group-learning method that uses the averaged predictions (forecasts) of several decision trees to increase the accuracy and reduce processing complexity [54]. First, the algorithm creates a random ensemble of trees, which are then combined using an appropriate technique, such as mathematical averaging for regression problems or voting majority for classification problems [55]. Instead of using a bootstrap replica, the extra trees build trees from the full learning sample, and node splitting is performed through entirely random cut-point selection. In this study, the Boruta extra tree scheme was established using Python's scikit-learn library [56] and the Boruta library.

2.2.2. XGBoost feature selection

The XGBoost technique is a variant of the gradient boosting machine approach [57] that implements the "boosting" notion by aggregating all the estimates from a group of weak learners to generate a strong learner (Fan et al., 2018). XGBoost divides the boosted trees into regression and classification trees [58], and the trees are grown as follows:

I) The decision tree is fitted to a subset of training data.
II) The residuals of the preceding tree are fitted with a loss function.
III) The following tree is formed by merging the loss function with the tree that came before it.
IV) The weighted cumulative contribution of all employed decision trees is output.

Recently, ensemble tree-based ML methods such as XGBoost have attracted considerable attention for feature selection as they can effectively estimate the profit and benefit related to each input feature during

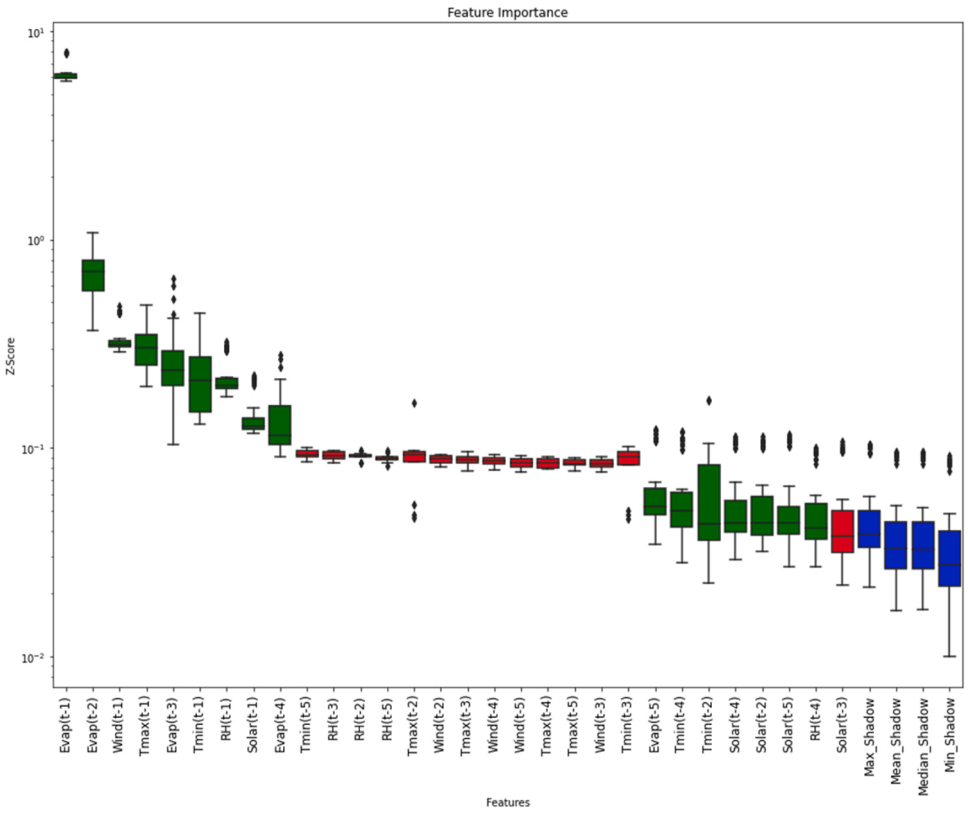

Fig. 5. Boruta extra tree feature selection for Ahvaz station.

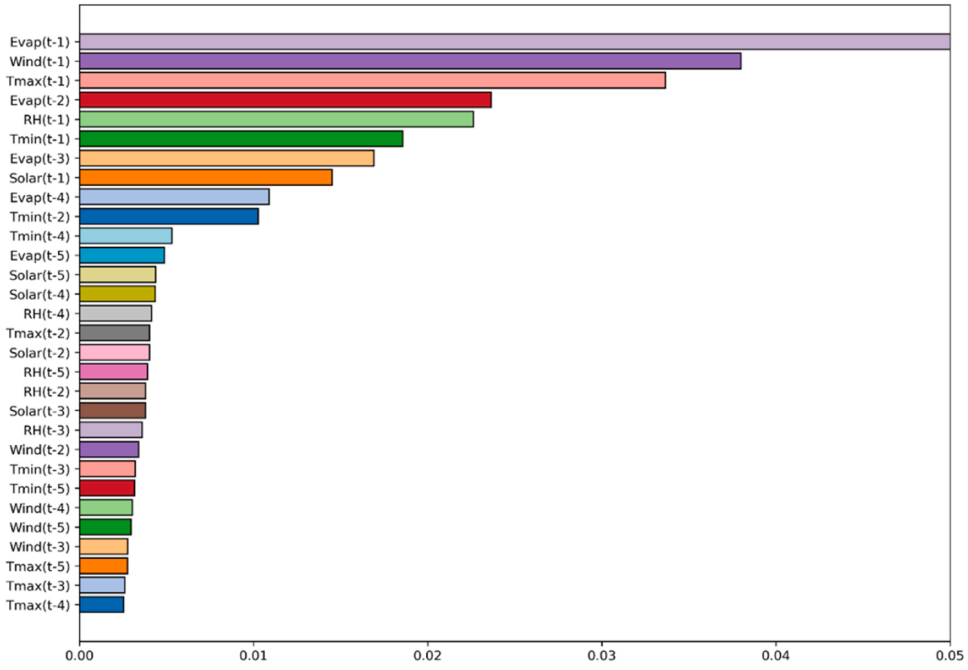

Fig. 6. XGBoost feature selection for Ahvaz station.

the training process. To reduce the dimensionality, XGBoost feature selection [57] can be performed to compute the feature significance score through the information gain as follows [59]:

$$\text{Gain} = \frac{1}{2}\left[\frac{\left(\sum_{\epsilon I_L}\widehat{g}_i\right)^2}{\sum_{\epsilon I_L}\widehat{h}_i+\lambda}+\frac{\left(\sum_{\epsilon I_R}\widehat{g}_i\right)^2}{\sum_{\epsilon I_R}\widehat{h}_i+\lambda}-\frac{\left(\sum_{\epsilon I}\widehat{g}_i\right)^2}{\sum_{\epsilon I}\widehat{h}_i+\lambda}\right] \quad (1)$$

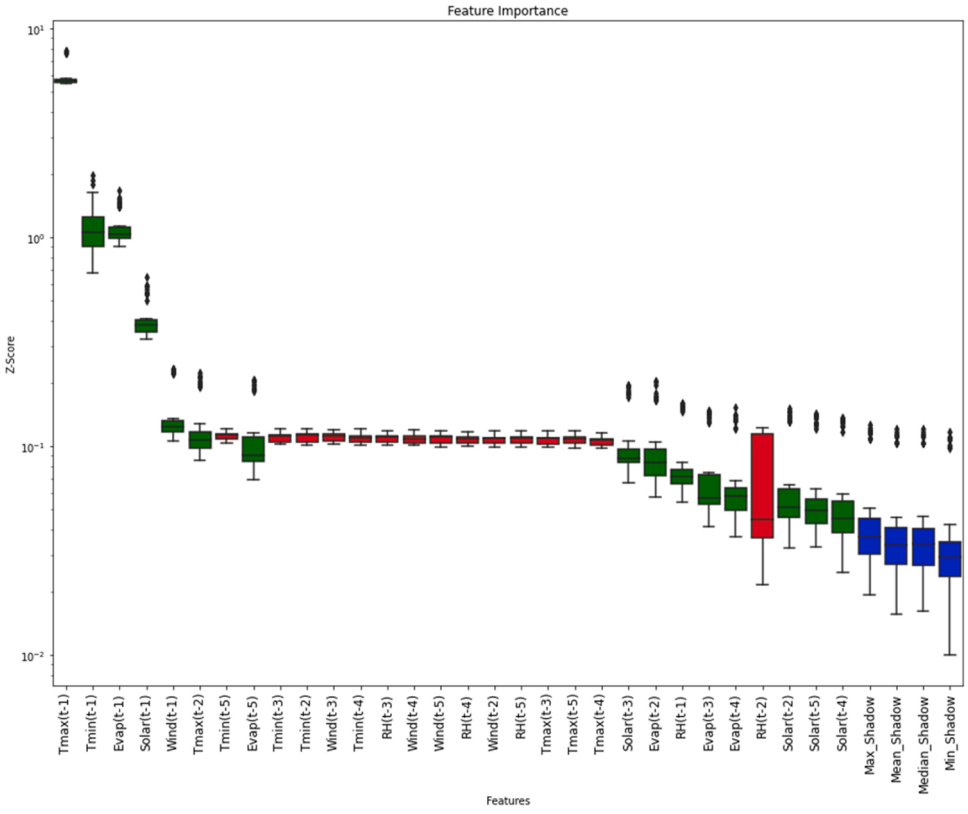

Fig. 7. Boruta extra tree feature selection for Yazd station.

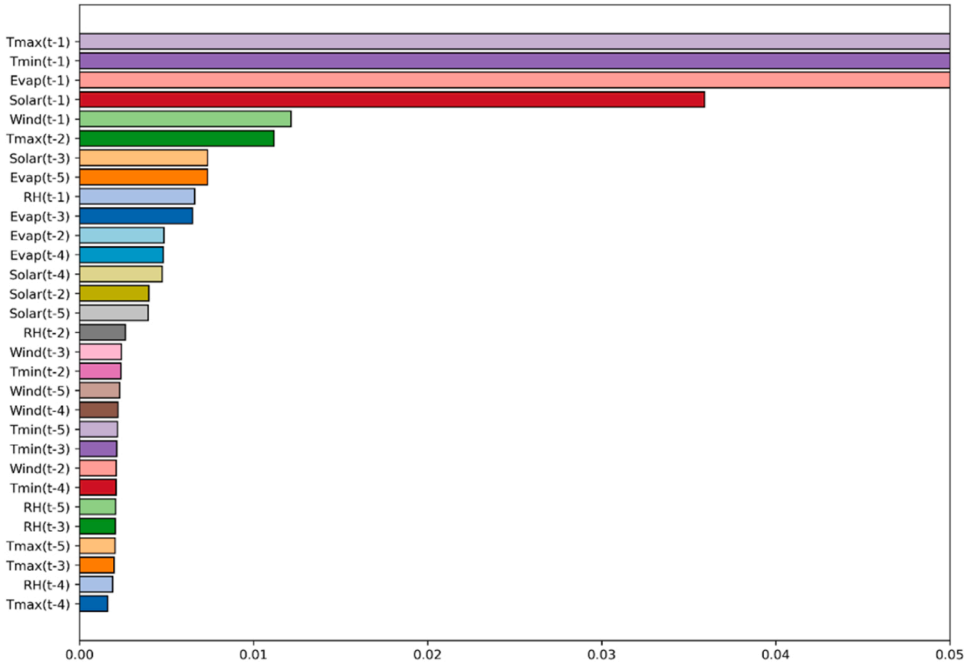

Fig. 8. XGBoost feature selection for Yazd station.

where $\widehat{g}_i$ and $\widehat{h}_i$ are the 1st and 2nd order gradients, respectively; λ and β are the penalty (regularization) coefficients; I_L and I_R denote the left and right nodes after segmentation, respectively, and i = 1,2,…,k.

Gain is the gain score for each split of a tree, and the average gain is used to determine the final feature importance value. The average gain is derived by dividing the overall gain of all trees by their total number of splits. A higher feature importance factor corresponds to a higher effectiveness and significance of the factor [59,60].

Table 2
Model settings for step-ahead forecasting of pan evaporation.

Study site	Models	Tuned parameters
Ahvaz-Boruta ET	AdaBoost	N_Estimators: 60, Learning Rate: 0.3
	GRNN	Spread: 0.225
	LSTM	Layers: 2, Number of Neurons: 100–150, Epochs: 50, Dropout: 0.1, Training Algorithm: Adam, Learning Rate: 0.0001, Batch Size: 128
	BiLSTM	Layers: 2, Number of Neurons: 50–200, Epochs: 50 Dropout: 0.1, Training Algorithm: Adam, Learning Rate: 0.0001, Batch Size: 64
Ahvaz-XGBoost	AdaBoost	N_Estimators: 60, Learning Rate: 0.3
	GRNN	Spread: 0.4
	LSTM	Layers: 2, Number of Neurons: 150–150, Epochs: 50, Dropout: 0.1, Training Algorithm: Adam, Learning Rate: 0.0002, Batch Size: 128
	BiLSTM	Layers: 2, Number of Neurons: 100–200, Epochs: 50 Dropout: 0.1, Training Algorithm: Adam, Learning Rate: 0.0001, Batch Size: 64
Study site	**Models**	**Tuned parameter**
Yazd-Boruta ET	AdaBoost	N_Estimators: 120, Learning Rate: 0.5
	GRNN	Spread: 0.35
	LSTM	Layers: 2, Number of Neurons: 200–200, Epochs: 50 Dropout: 0.1, Training Algorithm: Adam, Learning Rate: 0.0002, Batch Size: 64
	BiLSTM	Layers: 2, Number of Neurons: 210–210, Epochs: 50 Dropout: 0.08, Training Algorithm: Adam, Learning Rate: 0.0001, Batch Size: 64
Yazd-XGBoost	AdaBoost	N_Estimators: 80, Learning Rate: 0.6
	GRNN	Spread: 035
	LSTM	Layers: 2, Number of Neurons: 180–180, Epochs: 50 Dropout: 0.1, Training Algorithm: Adam, Learning Rate: 0.00011, Batch Size: 64
	BiLSTM	Layers: 2, Number of Neurons: 220–100, Epochs: 50 Dropout: 0.1, Training Algorithm: Adam, Learning Rate: 0.00011, Batch Size: 64

2.3. ML algorithms

2.3.1. GRNN

Specht (1991) introduced the GRNN as an extension of the radial basis function (RBF) for solving nonlinear engineering problems. The primary difference between the GRNN and conventional feed-forward neural network is that the former requires the determination of an ideal number of hidden layers and hidden nodes, whereas the latter does not. A GRNN model with forward propagation can estimate any arbitrary function among input–output vectors [61] and is classified as a probabilistic neural network [62]. The learning process of the GRNN seldom falls into local minima. The fundamental architecture of the GRNN model is shown in Fig. 2, which consists of four units, including the input layer, pattern layer, summation layer, and output layer. The RBF kernel spread parameter, which is used to figure out how similar the input parameters are, needs to be tuned in the GRNN. The forecasted outcome $\widehat{Y} \in \Re$ for the input vector $x_t \in \Re^d, t = 1, 2, \ldots, N$ forms an N-training dataset [63] and can be defined as

$$\widehat{Y} = \frac{\sum_{i=1}^{N} Y^i \exp\left(-\frac{D_i^2}{2\sigma_*^2}\right)}{\sum_{i=1}^{N} \exp\left(-\frac{D_i^2}{2\sigma_*^2}\right)} \tag{2}$$

$$D_i^2 = \left(X - X^i\right)^T \left(X - X^i\right) \tag{3}$$

where $\widehat{Y}$ and Y^i denote the forecasted output and output of the input layer, respectively; σ_* denotes the spread hyperparameter or smoothing parameter; D represents the distance function related to the RBF kernel; and X^i is the training input vector related to N variables. T is the transpose matrix [64].

2.3.2. AdaBoost

AdaBoost [65] was designed to increase the efficiency of ML learning. Regression problems are solved with the AdaBoost regressor, which is a special kind of AdaBoost. The main concept underlying the AdaBoost regressor is to regularly reweight the training examples based on the prediction error at each boosting iteration to learn a succession of weak regressors with a large bias error but a low variance error [66]. In this method, the subsequent regressor assigns greater weight to the cases that were mistakenly predicted in the preceding step. The coefficients of all weak regressors are combined to produce the final prediction and generate a model with minimum bias and variance error [56,67]. This method is not vulnerable to overtraining. The mechanism of the AdaBoost algorithm can be described as follows [68]:

The training dataset is $\Psi = \left\{(x_1, y_1), \ldots, \left(x_j, y_j\right)\right\}$

where $x_j$denotes the input vector, y_j denotes the output (target), and m is number of training data points.

By initiating the weight vector, the AdaBoost procedure can be expressed as

$$\omega_1(x_i) = \frac{1}{j} for\ i = \{1, 2, \ldots, j\} \tag{4}$$

The error rate based on T weak learners, $h_t(t = 1, 2, \ldots, T)$, is described as [69]

$$\pi_t = \sum P_t(x_i)[h_t(x_i) - y_i] \tag{5}$$

Here, $P_t(x_i)$ can be obtained using an arithmetic average function $\frac{\omega_t(x_i)}{\sum_{i=1}^{j} \omega_t(x_i)}$, with $\pi_t > 0.5$ and (T=t-1) for terminating the loop. To calculate each weight, the weight confidence is defined as

$\alpha_t = \log\left(\frac{\pi_t}{1-\pi_t}\right)$.

To updated the weight during training, $\omega_{t+1}(x_i) = \omega_t(x_i) \times e^{-y_i h_t(x_i)\alpha_t}$ can be applied in a new stage, where $t < Tt$ determines $P_t(x_i)$ and $\pi_t > 0.001$. This process is continued until the requirements are satisfied. Finally, the aggregated learner $H(x)$ with the highest precision is obtained by aggregating all the weak weight learners, as follows [66]:

$$H(x) = sign\left[\sum_j^T \alpha_t[h_t(x) = y]\right] \tag{6}$$

The CART decision tree is employed as a base regressor in this study. The three key hyperparameters that define the AdaBoost model are the learning rate (to adjust the contribution of weak learners to the final strong learner), maximum tree depth, and maximum number of trees. Fig. 3 illustrates the process flow of the AdaBoost algorithm.

2.3.3. LSTM

The LSTM is a sophisticated RNN model that can handle time-series data [41]. The RNN can record sequences; however, the disappearing or inflating gradients prevent it from remembering long-term dependencies in extended sequences [70]. The LSTM is designed to prevent or resolve this problem. Specifically, this problem is resolve using a gating mechanism to regulate information, as shown in Fig. 4.a. The LSTM uses an input gate, a forgetting gate, and an output gate to exclude irrelevant data from the present situation [71]. This mechanism extends the duration of information storage and preserves several earlier data points. The input of the LSTM gate is the state that is hidden between the current time step input x_t and prior time step h_{t-1} [72]. Full connection layer computations determine the output. The LSTM model can be mathematically described as follows [73]:

$$\text{Input gate}: i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i \tag{7}$$

$$\text{Forgetting gate}: f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{8}$$

$$\text{Gated unit}: \widetilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_c\right) \tag{9}$$

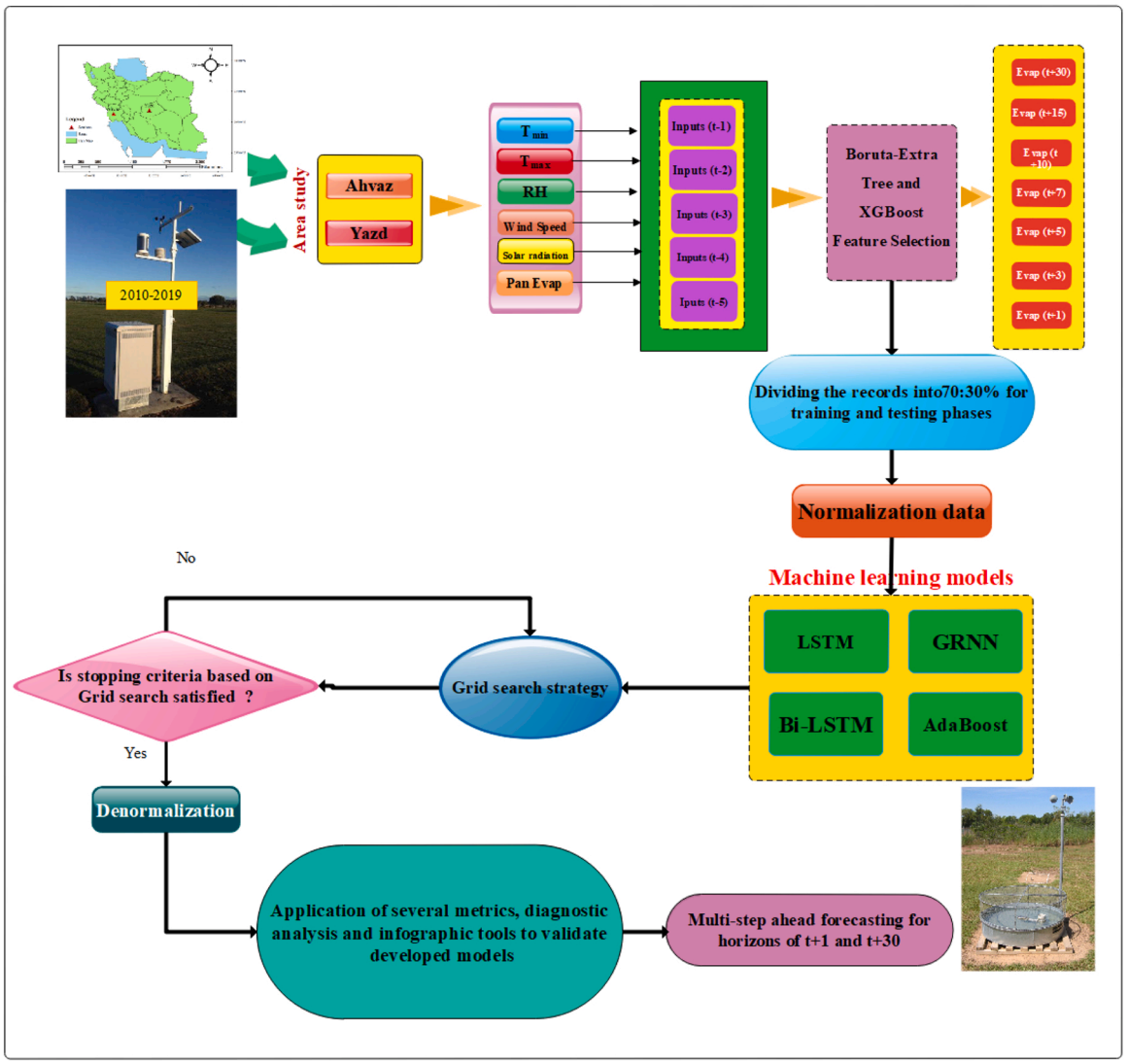

Fig. 9. Methodology for pan evaporation forecasting.

Table 3
Results of pan evaporation forecasting for one step ahead in Ahvaz Station. The boldfaced text indicates the best model.

Model	Set	R	MAPE	RMSE	U95%	KGE	IA
Boruta-BiLSTM	**Train**	**0.9520**	**22.0334**	**1.8800**	**5.0319**	**0.8314**	**0.9653**
	Test	**0.9396**	**25.3541**	**1.6857**	**4.6684**	**0.8972**	**0.9670**
Boruta-LSTM	Train	0.9495	22.4752	1.7859	4.8888	0.8749	0.9697
	Test	0.9350	26.5599	1.7835	4.8936	0.9123	0.9648
Boruta-GRNN	Train	0.9499	23.4915	1.7203	3.3697	0.9039	0.9725
	Test	0.9283	32.1696	2.0051	6.7607	0.8749	0.9570
Boruta-AdaBoost	Train	0.9481	32.5784	1.7732	4.8863	0.8915	0.9705
	Test	0.9268	42.0809	2.2129	5.7005	0.8270	0.9487
XGBoost-BiLSTM	Train	0.9515	21.8933	1.7418	4.7797	0.8819	0.9713
	Test	0.9366	27.2928	1.8254	4.9476	0.9029	0.9643
XGBoost-LSTM	Train	0.9506	23.6593	1.7094	4.7373	0.9025	0.9728
	Test	0.9351	30.1372	1.9188	5.1230	0.8815	0.9611
XGBoost-GRNN	Train	0.9317	33.5736	2.0892	4.0939	0.8019	0.9543
	Test	0.9236	40.7774	2.0375	7.0261	0.8253	0.9499
XGBoost-AdaBoost	Train	0.9476	32.5935	1.7776	4.9024	0.8915	0.9704
	Test	0.9284	41.4599	2.1641	5.5948	0.8342	0.9506

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \tag{10}$$

$$\text{Output gate}: O_t = \sigma\left(w_o[h_{t-1}, x_t] + b_o\right) \tag{11}$$

where W_f, W_i, W_c, and W_o represent the weight matrices of the forget gate, input gate, cell gate, and output gate units, respectively, with the corresponding bias vectors being b_f, b_i, b_c, and b_o. σ is the conventional activation function of the forget gate and input gate components. tanh represents the activation functions of the cell gate unit. $\odot$ is the element-

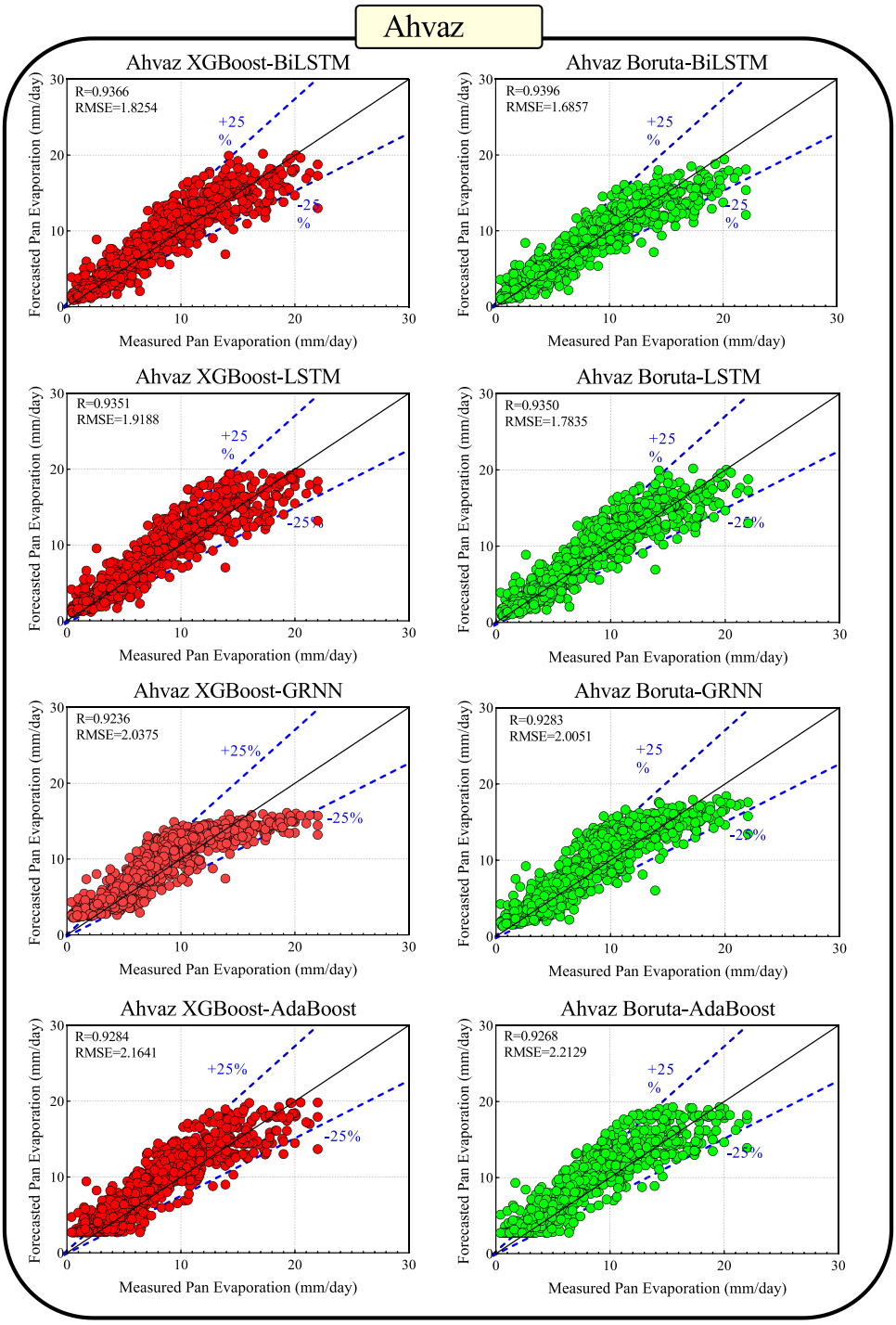

Fig. 10. Scatter plots of the pan evaporation forecasting for Ahvaz station.

wise product. Fig. 4.a shows the typical structure of LSTM cell.

2.3.4. BiLSTM

In the operation of a one-directional LSTM, the input is processed only in the forward direction, or, in other words, only the existing information is used. In contrast, BiLSTM consists of two layers of LSTM, one with forward motion and the other with reverse motion [74]. The layer with forward motion receives the past information for input sequence data, whereas that with backward motion (also known as reverse LSTM) receives future information for the input sequence data. After the data have passed through these layers at least once, the output received from both layers is merged and forwarded for further processing. The hidden layer state of each level (h_t) in the BiLSTM network consists of three components: the forward hidden vector h_t, backward hidden vector h_i, and current input x_t. The hidden layer state of each level can be expressed as [75].

$$h_t = LSTM(x_t, h_{t-1}) \tag{12}$$

$$h_i = LSTM(x_t, h_{i-1}) \tag{13}$$

$$y_t = \sigma(a_t h_t + b_t h_i + C_t) \tag{14}$$

where LSTM() represents the standard working mechanism of an LSTM network, and a_t and b_t represent the forward and backward output

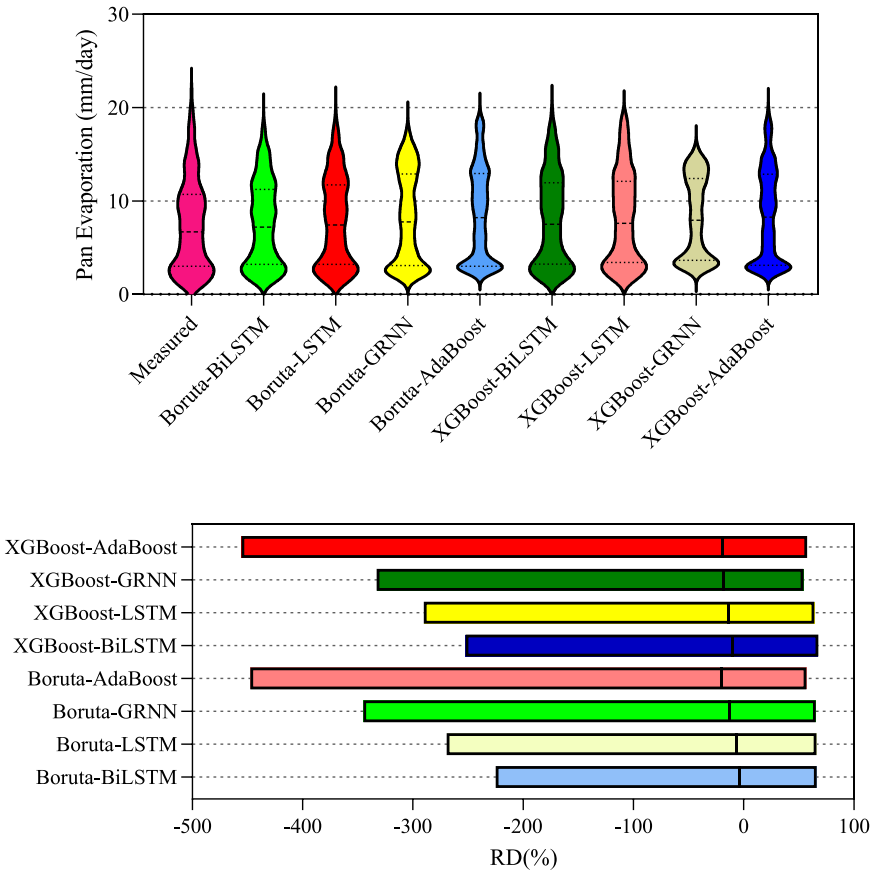

Fig. 11. Violin plots of measured and forecasted values (Top) and relative deviation (RD, %) variation (Bottom) for different models for Ahvaz station.

Table 4
Results of pan evaporation forecasting for one step ahead in Yazd Station. The boldfaced text indicates the best model.

Model	Set	R	MAPE	RMSE	U95%	KGE	IA
Boruta-BiLSTM	**Train**	**0.9631**	**15.1255**	**1.3160**	**3.6426**	**0.9334**	**0.9804**
	Test	**0.9535**	**15.2988**	**1.3996**	**3.8804**	**0.9408**	**0.9759**
Boruta-LSTM	Train	0.9618	14.4430	1.3760	3.7807	0.9026	0.9778
	Test	0.9521	15.0656	1.4779	4.0215	0.9085	0.9724
Boruta-GRNN	Train	0.9433	19.8349	1.6248	3.1852	0.8903	0.9686
	Test	0.9368	18.7226	1.6352	5.1771	0.9051	0.9659
Boruta-AdaBoost	Train	0.9559	21.5598	1.4860	4.0576	0.8986	0.9742
	Test	0.9440	18.9884	1.5435	4.2624	0.9186	0.9701
XGBoost-BiLSTM	Train	0.9517	15.5046	1.5248	4.2009	0.8974	0.9726
	Test	0.9490	15.3590	1.4908	4.0971	0.9138	0.9719
XGBoost-LSTM	Train	0.9504	16.0163	1.6306	4.4108	0.8600	0.9677
	Test	0.9491	15.8161	1.5771	4.2353	0.8779	0.9677
XGBoost-GRNN	Train	0.9384	24.6401	1.8285	3.5845	0.7819	0.9549
	Test	0.9332	20.7898	1.7557	5.9973	0.8080	0.9556
XGBoost-AdaBoost	Train	0.9478	21.6732	1.6112	4.4148	0.8764	0.9688
	Test	0.9357	19.5082	1.6628	4.5787	0.8921	0.9642

weights of the hidden layer, respectively. C_t is the offset optimization parameter for the hidden layer at time t. Fig. 4.b shows the typical structure of BiLSTM model.

2.4. Model development

In this study, daily meteorological pan evaporation time-series data were used to forecast pan evaporation in Ahvaz and Yazd stations for the next 1–30 d. GRNN, AdaBoost, LSTM, and BiLSTM were used to forecast time series in Python and MATLAB (2022b) environments. Python was used to create the AdaBoost model (scikit-learn package version 1.0.2), the LSTM and BiLSTM models were implemented the Keras package (version 2.11.0), and the GRNN model was created using MATLAB. Significant time delays in forecasting were determined using the Boruta-Shap package (version 1.0.16) and the XGBoost package (version 1.7.2) in python was used for creating XGBoost feature selection model. The following subsections describes the stages of model development.

2.4.1. Model inputs and determination of important features

Feature selection is a crucial stage for ML-based models due to its capacity to decrease computation cost and input dimensionality while increasing accuracy. We used lags of meteorological data from 5 days ago to predict the pan evaporation for the next 30 days. Because not all five lags may not have the same level of significance, the Boruta and XGBoost methods were used to identify the best inputs. The random forest technique is the foundation of the conventional Boruta model.

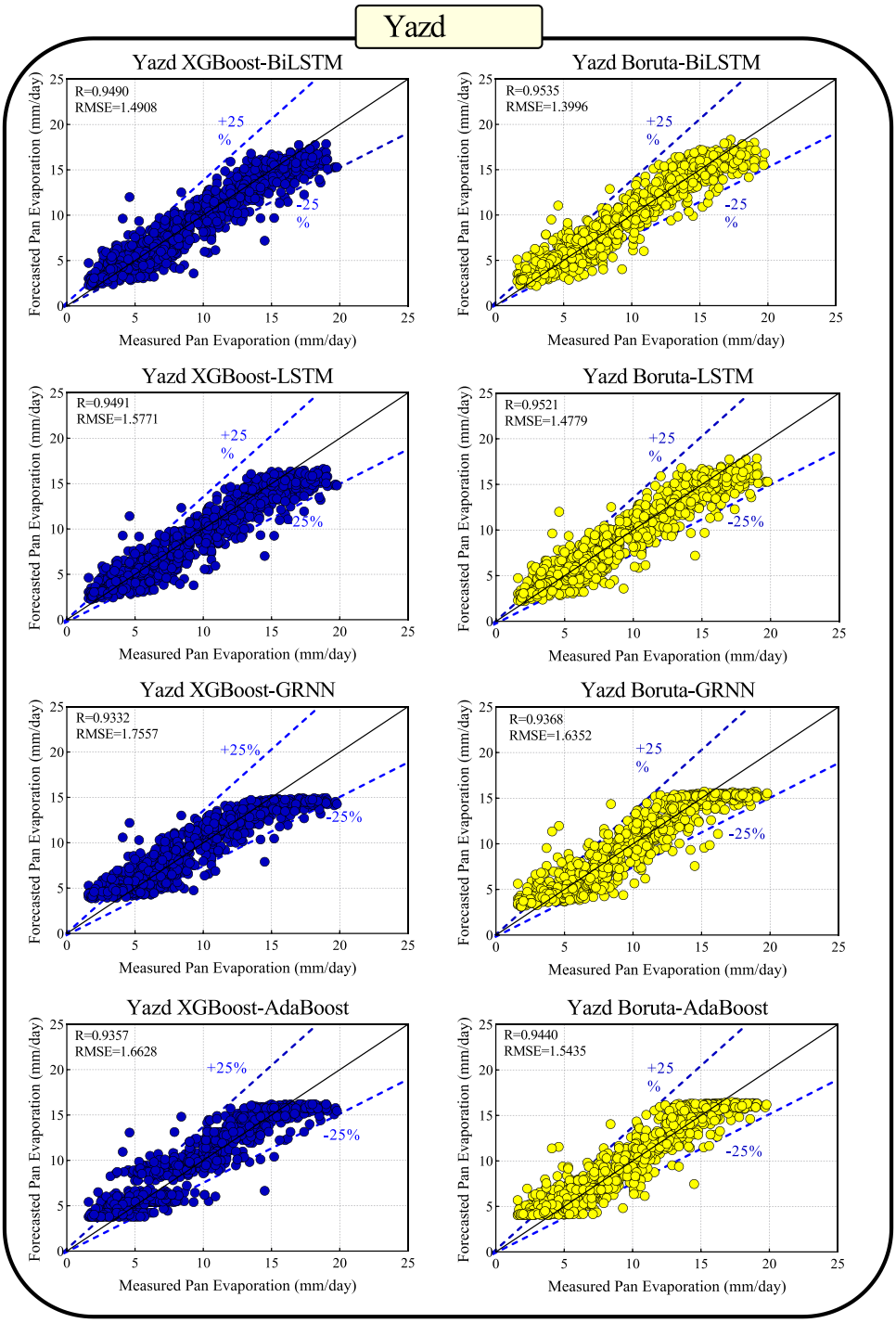

Fig. 12. Scatter plots of the pan evaporation forecasting for Yazd station.

However, the extra tree approach was used in this study to identify the optimal inputs. In the Boruta technique, the inputs that are less effective than the Max-Shadow benchmark value are eliminated. Figs. 5 and 7 show the Z-score values for lags up to 5 days used to forecast pan evaporation for the Ahvaz and Yazd stations, respectively. The ineffective and effective, inputs are shown in red and green, respectively. Figs. 6 and 8 show the importance of different meteorological lags to forecast pan evaporation using the XGBoost model. The threshold of importance was set as 0.01.

To create ML models for daily pan evaporation predictions, the collected data were divided into training and testing sets with ratios of 70% (2010–2016) and 30% (2017–2019), respectively. To ensure that all of the data in the training set were the same size and that the model rapidly converged, they were normalized between 0 and 1 and then denormalized after forecasting model.

2.4.2. *Hyperparameter tuning*

A key task in ML prediction is to adjust the model's hyperparameters to enhance the precision and efficiency. Several techniques, such as meta-heuristic optimization algorithms, grid search, and random search, have been developed to optimize the model parameters. A grid search method was used in this study. Notably, in these techniques, an effectiveness measure must be defined, typically through cross-validation on the training set or testing on a hold-out validation set. Table 2 represents

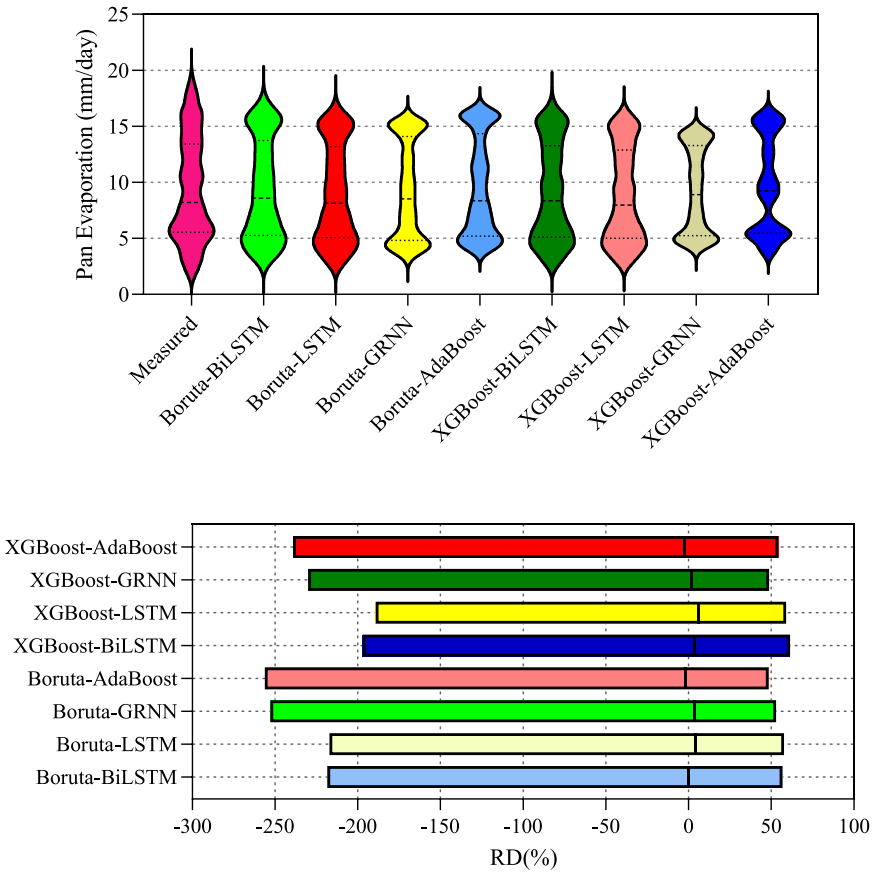

Fig. 13. Violin plots of measured and forecasted values (Top) and RD % variation (Bottom) for different models for Yazd station.

the optimized hyperparameters for different combinations of feature selection algorithms and ML models at both stations. The GRNN model requires the optimization of only one hyperparameter, i.e., the spread value. This parameter was searched in the range [0.01, 0.5]. The key parameters in the AdaBoost model, N_estimator and learn_rate, were sought in the ranges of [10,150] and [0.01, 0.9], respectively. The LSTM and BiLSTM models have similar hyperparameters, i.e., the numbers of layers and neurons in each layer, and their values were sought in ranges of 1–4 and 50–250, respectively. The ideal settings were established by minimizing the root mean square error (RMSE). The models were trained through Adam learning, and the rectified linear unit (Relu) function was used as the activation function. Dropout values in the range [0, 0.3] were employed to prevent model overfitting. In general, the dropout shows the number of training samples chosen at random from the total number of samples. The remaining parameters of the LSTM and BiLSTM models were the learning rate (learning effectiveness of the network), batch size (quantity of data in each batch), and epoch (number of iterations necessary to attain the optimal model). Table 2 displays the results of a grid search performed to obtain these values. Fig. 9 shows the process flow for forecasting the daily pan evaporation.

2.5. Statistical metrics

The robustness of the proposed ML-based forecasting models was evaluated using six statistical indicators: the coefficient of correlation (R), mean absolute percent error (MAPE), uncertainty coefficient with 95% confidence level ($U_{95\%}$), root mean square error (RMSE), Kling–Gupta efficiency (KGE) [76] and index of agreement (I_A). The mathematical expressions of these metrics are [77,78].

$$R = \frac{\sum_{i=1}^{N}\left(PE_{obs,i} - \overline{PE_{obs}}\right)\left(PE_{for,i} - \overline{PE_{for}}\right)}{\sqrt{\sum_{i=1}^{N}\left(PE_{obs,i} - \overline{PE_{obs}}\right)^2 \sum_{i=1}^{N}\left(PE_{for,i} - \overline{PE_{for}}\right)^2}} \tag{15}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(PE_{obs,i} - PE_{for,i}\right)^2} \tag{16}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{PE_{obs,i} - PE_{for,i}}{PE_{obs,i}}\right| \tag{17}$$

$$KGE = 1 - \sqrt{(R-1)^2 + (\alpha-1)^2 + (\beta-1)^2} \tag{18}$$

$$I_A = 1 - \left[\frac{\sum_{i=1}^{N}\left(PE_{for,i} - PE_{obs,i}\right)^2}{\sum_{i=1}^{N}\left(\left|PE_{for,i} - \overline{PE_{obs}}\right| + \left|PE_{obs,i} - \overline{PE_{obs}}\right|\right)^2}\right] \tag{19}$$

$$U_{95\%} = 1.96\sqrt{ST_e^{\,2} + RMSE^2} \tag{20}$$

where $PE_{obs,i}$ and $PE_{for,i}$ are the observed and forecasted values of PE, respectively; and $\overline{PE_{obs}}$ and $\overline{PE_{for}}$ denote the mean values of the observed and forecasted PE parameters, respectively. N is the length of the time series, α is the relative standard deviation of the forecast and observed pan evaporation, β is the ratio between the average forecasted and observed PE values, and ST_e is the standard deviation of error. A model is optimal when R, I_A, $U_{95\%}$, and KGE are one, and all the diagnostic metrics, such as RMSE and MAPE, are zero.

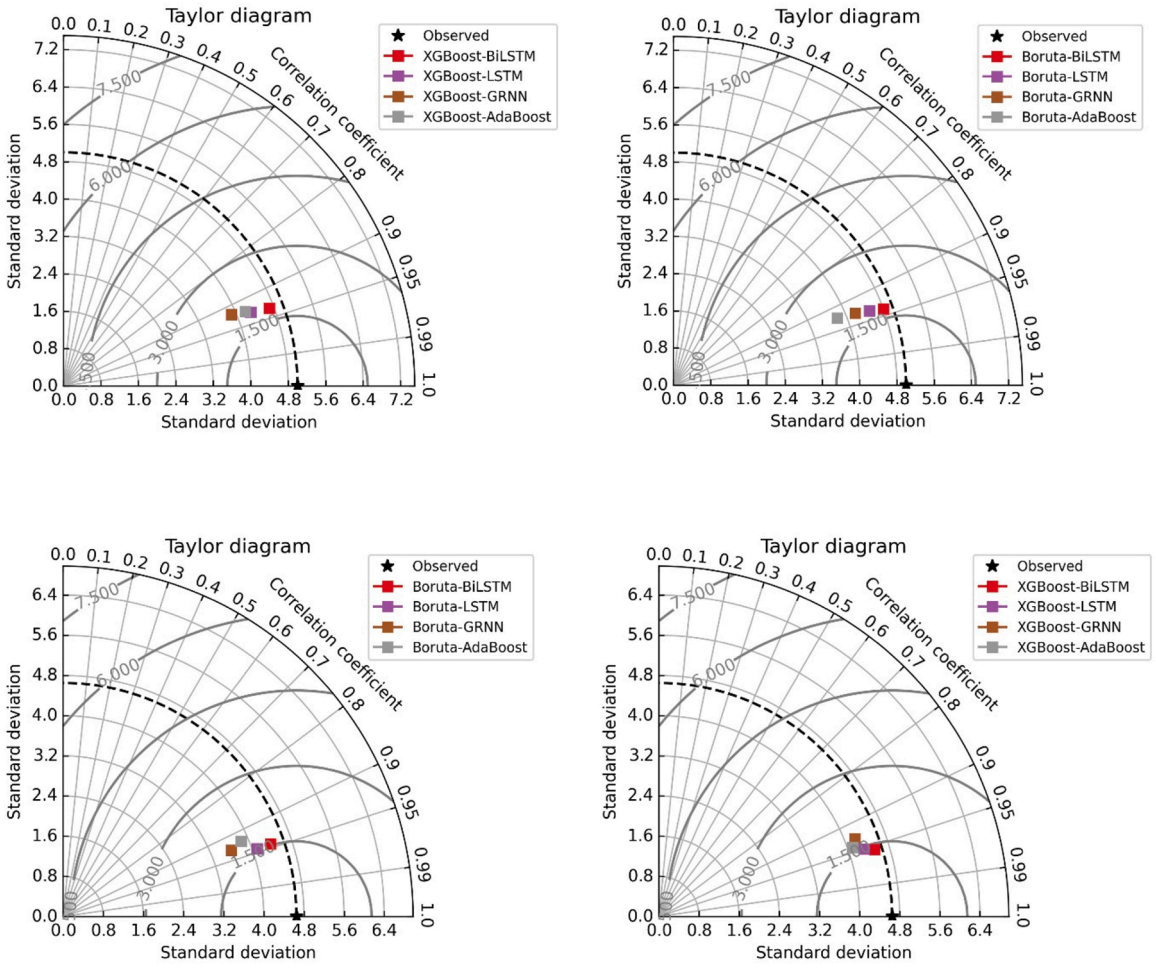

Fig. 14. Taylor diagrams for Ahvaz and Yazd stations.

3. Application results and analysis

The forecasting ability of the Boruta-BiLSTM was compared with the Boruta-LSTM, Boruta-GRNN, Boruta-AdaBoost, XGBoost-BiLSTM, XGBoost-LSTM, XGBoost-GRNN and XGBoost-AdaBoost models, in terms of the R, MAPE, KGE, RMSE, U95%, and IA. Moreover, diagnostic plots were obtained for the training and testing periods to forecast the pan evaporation for Ahvaz and Yazd stations in Iran.

Table 3 shows that the Boruta-BiLSTM model achieved the highest forecasting accuracy in the Ahvaz station (R = 0.9520, MAPE = 22.0334, RMSE = 1.8800, $U_{95\%}$ = 5.0319, KGE = 0.8314, IA = 0.9653 in training; R = 0.9396, MAPE = 25.3541, RMSE = 1.6857, $U_{95\%}$ = 4.6684, KGE = 0.8972, IA = 0.9670 in testing). The second-highest performing model was Boruta-LSTM (R = 0.9495, 0.9350; RMSE = 1.7835, 1.7859; MAPE = 22.4752, 26.5599; KGE = 0.8749, 0.9123, IA = 0.9697, 0.9648; $U_{95\%}$ = 4.8888, 4.8936 in the training and testing periods, respectively), followed by XGBoost-BiLSTM, and other comparing models (refer to the Table 3). While all the models exhibtes reasonably good performance for each assessment metric, but the best performance was attained by the Boruta-BiLSTM model as judged by these metrics in Table 3. Therefore, the Boruta-BiLSTM model forecasted values closer to the measured pan evaporation based on R, KGE, and IA metrics whease generating lower errors by judging the MAPE, and RMSE metrics for Ahvaz station. The values close to 1 of these R, KGE, and IA metrics report that the forecast are accurate. The $U_{95\%}$ metrics ascertain the uncertainty coefficient with 95% confidence level (U95%) of each model. Overall, the Boruta-based models outperformed the XGBoost-based models in forecasting the pan evaporation for Ahvaz station.

Scatter plots (Fig. 10) further highlight the forecasting ability of the Boruta-BiLSTM in comparison with the Boruta-LSTM, Boruta-GRNN, Boruta-AdaBoost, XGBoost-BiLSTM, XGBoost-LSTM, XGBoost-GRNN, and XGBoost-AdaBoost models for Ahvaz station. Fig. 10 shows both the R and RMSE values for each model. The Boruta-BiLSTM achieved a slightly higher accuracy than the XGBoost-BiLSTM with R (0.9396, 0.9366) and RMSE (1.6857, 1.8254), which was the second-highest performing model. The scatter plots showed that the hybrid versions of the Boruta and XGBoost achieved the highest precisions with the highest R and lowest RMSE values in forecasting pan evaporation for the Ahvaz station.

Fig. 11 shows the violin plots (top) and relative deviation (RD, %) (down) to further assess the prediction effectiveness of the Boruta-BiLSTM in comparison with the other comparing models for Ahvaz station. The Boruta-BiLSTM model generated the most accurate violin distribution. Moreover, the RD (%) error variations of the Boruta-BiLSTM model were the lowest, ranging between 100 and − 250, for the Ahvaz station. The violin and RD (%) plots showed that the Boruta-BiLSTM achieved the most accurate predictions for the Ahvaz station compared with the other models.

For the Yazd station (Table 4), the Boruta-BiLSTM model achieved the highest precision based on R = 0.9631, MAPE = 15.1255, RMSE = 1.3160, $U_{95\%}$ = 3.6426, KGE = 0.9334, IA = 0.9804 in training period and R = 0.9535, MAPE = 15.2988, RMSE = 1.3996, $U_{95\%}$ = 3.8804, KGE = 0.9408, IA = 0.9759 in testing period to forecast the pan evaporation. The second-highest performing model was Boruta-LSTM, followed by XGBoost-BiLSTM and Boruta-AdaBoost. Here again the Boruta-BiLSTM model forecasted values were much closer to the

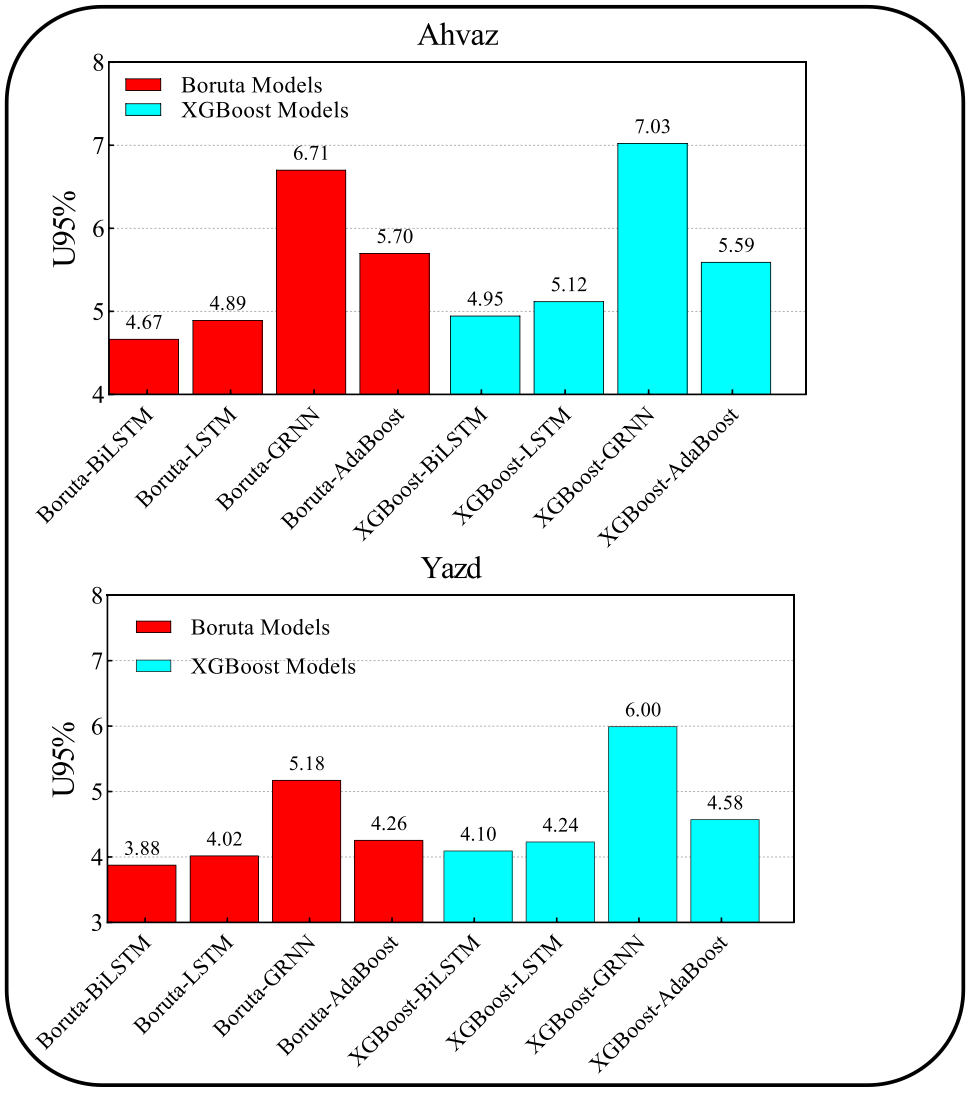

Fig. 15. Uncertainty variations in different models for Ahvaz and Yazd stations.

measured pan evaporation by analysing these metrics R, MAPE, KGE, RMSE, U95%, and IA metrics for Yazd station while the other comparing models forecasts were more scattered.

Fig. 12 shows the scatter diagram illustrating the relationship between the measured and forecasted pan evaporation generated by Boruta-BiLSTM and other benchmarking models for Yazd station, in terms of the R and RMSE values. The Boruta-BiLSTM (R = 0.9535; RMSE = 1.3996) achieved the highest precision, followed by Boruta-BiLSTM (R = 0.9521; RMSE = 1.4779), XGBoost-BiLSTM, Boruta-GRNN, Boruta-AdaBoost, XGBoost-BiLSTM, XGBoost-LSTM, XGBoost-GRNN, and XGBoost-AdaBoost models. Therefore, the hybrid versions of the Boruta framework outperformed the XGBoost model in terms of the accuracy for the Yazd station.

Fig. 13 shows the violin plots (top) and RD (%) (bottom) for the Yazd station produced by the Boruta-BiLSTM in comparison with the Boruta-LSTM, Boruta-GRNN, Boruta-AdaBoost, XGBoost-BiLSTM, XGBoost-LSTM, XGBoost-GRNN, and XGBoost-AdaBoost models. The measured pan evaporation is also plotted for a more comprehensive and detailed analysis. The Boruta-BiLSTM model achieved a more reasonable violin distribution against the observed pan evaporation, followed by the XGBoost-BiLSTM and Boruta-LSTM models. Moreover, the Boruta-BiLSTM generated lower RD (%) error variations ranging between 100 to − 250 compared with the benchmarking models for Yazd station. In other words. the Boruta-BiLSTM precisely forecasted the pan evaporation.

Fig. 14 shows a Taylor diagram of the Boruta-based hybrid (Right) and XGBoost-based hybrid (Left) models to assess the forecasting accuracy for pan evaporation in Ahvaz and Yazd stations. The Taylor diagram can comprehensively compare the model forecasts in relation to the observed/measured data based on the standard deviation and correlation coefficient. The Boruta-BiLSTM and XGBoost-BiLSTM models appeared to be closer to the observed pan evaporation, in comparison with the other counterpart models for both Ahvaz and Yazd stations. Overall, the Boruta-BiLSTM can most accurately forecast the pan evaporation, followed by XGBoost-BiLSTM.

Fig. 15 shows the U95% metrics in terms of the bar chart of the Boruta-BiLSTM against the comparing models for both Ahvaz and Yazd stations. The Boruta-BiLSTM model achieves the highest accuracy with the lowest values of U95% = 4.67 (Ahvaz station); and 3.88 (Yazd station) against the observed pan evaporation, followed by the Boruta-LSTM and XGBoost-BiLSTM models. Overall, the Boruta-BiLSTM is the most superior pan evaporation model for both stations.

3.1. *Multi-step-ahead pan evaporation forecasting*

Table 5 indicates the multi-step (i.e., 3-, 5-, 7-, 10-, 15-, and 30-d)

Table 5
Results of multi-step-ahead pan evaporation forecasting in Ahvaz and Yazd stations.

Station	Step ahead	Set	R	MAPE	RMSE	U95	KGE	IA
Ahvaz	3	Train	0.9219	29.4784	2.1205	4.1570	0.8797	0.9573
		Test	0.8993	34.7461	2.2787	9.4532	0.8685	0.9434
	5	Train	0.9272	28.4687	2.0544	4.0274	0.8769	0.9596
		Test	0.8882	36.7928	2.4156	10.5999	0.8571	0.9368
	7	Train	0.9777	16.6697	1.1844	2.3218	0.9206	0.9872
		Test	0.8822	39.2256	2.4500	11.0295	0.8543	0.9338
	10	Train	0.9308	29.1025	2.0062	3.9330	0.8756	0.9614
		Test	0.8863	38.9771	2.4177	10.6957	0.8574	0.9361
	15	Train	0.9474	26.2316	1.7688	3.4675	0.8858	0.9702
		Test	0.8796	40.0755	2.4683	11.2746	0.8543	0.9328
	30	Train	0.9315	30.2974	2.0105	3.9414	0.8560	0.9601
		Test	0.8411	49.0542	2.8745	14.9913	0.8083	0.9083
Yazd	3	Train	0.9522	16.5608	1.4872	2.9156	0.9222	0.9746
		Test	0.9310	18.8914	1.6993	5.6620	0.9184	0.9640
	5	Train	0.9457	17.6839	1.5822	3.1018	0.9121	0.9710
		Test	0.9216	20.3108	1.8078	6.4022	0.9054	0.9588
	7	Train	0.9506	16.9035	1.5137	2.9674	0.9139	0.9734
		Test	0.9209	20.1464	1.8156	6.4583	0.9052	0.9585
	10	Train	0.9503	17.1883	1.5190	2.9778	0.9109	0.9732
		Test	0.9171	20.9873	1.8584	6.7588	0.9019	0.9564
	15	Train	0.9616	15.3361	1.3466	2.6399	0.9173	0.9790
		Test	0.9088	22.7878	1.9473	7.4069	0.8911	0.9516
	30	Train	0.9710	13.6451	1.1859	2.3249	0.9172	0.9837
		Test	0.8767	27.0591	2.2555	9.9167	0.8577	0.9336

ahead pan evaporation forecasts for Ahvaz and Yazd stations based on the Boruta-BiLSTM model. The Boruta-BiLSTM model exhibits an excellent performance for multi-step-ahead (3-d) pan evaporation forecasting based on the assessment metrics [(R = 0.9219, MAPE = 29.4784, RMSE = 2.1205, U95% = 4.1570, KGE = 0.8797, IA = 0.9573,)-train; (R = 0.8993, MAPE = 34.7461, RMSE = 2.2783, U95% = 9.4532, KGE = 0.8685, IA = 0.9434,)-test] for Ahvaz station. The performance is satisfactory for the Yazd station as well [(R = 0.9522, MAPE = 16.5608, RMSE = 1.4872, U95% = 2.9156, KGE = 0.9222, IA = 0.9746,)-train; (R = 0.9310, MAPE = 18.8914, RMSE = 1.6993, U95% = 5.6620, KGE = 0.9184, IA = 0.9640,)-test]. Similarly, the Boruta-BiLSTM achieves reasonable forecasts for the 5-, 7-, 10-,15-, and 30-d-ahead pan evaporation for both Ahvaz and Yazd stations (Table 5) in both the training and testing periods. The outcomes highlight that the Boruta-BiLSTM is the optimal pan evaporation forecasting model for Ahvaz and Yazd stations.

Fig. 16 shows polar graphs that indicate the RMSEs incurred by the Boruta-BiLSTM model in forecasting multi-step-ahead pan evaporation for the Ahvaz and Yazd stations. The Boruta-BiLSTM model produced the smallest RMSE in the 3, 5, 7, 10, 15, and 30-d ahead pan evaporation forecasting. The highest accuracy corresponded to the 3 d problem, followed by the 5-, 7-,10-, and 30 d problems for both Ahvaz and Yazd stations.

4. Discussion and future direction

This paper aims to construct a novel Boruta-BiLSTM framework, which is the hybridization of bidirectional long short-term memory (BiLSTM) with Boruta extra tree (Boruta) to forecast multi-step-ahead pan evaporation in Ahvaz and Yazd stations in Iran. Moreover, the BiLSTM and XGBoost were also integrated to design the XGBoost-BiLSTM model to forecast pan evaporation. The Boruta and XGBoost models are the feature selection models to select significant inputs, which can help in optimization to enhance the forecasting accuracy of the BiLSTM model in the hybrid modeling paradigm. Moreover, Boruta and XGBoost models are used to reduce the number of inputs and model complexity. For comparison, the Boruta-LSTM, Boruta-GRNN, Boruta-AdaBoost, XGBoost-LSTM, XGBoost-GRNN, and XGBoost-AdaBoost models were also constructed to forecast multi-step-ahead pan evaporation in Ahvaz and Yazd stations in Iran.

The forecasting result confirms that the Boruta-BiLSTM models outperformed all comparing the hybrid and standalone versions of the models for both stations based on goodness-of-fit metrics and diagnostic plots. The novel Boruta-BiLSTM models model is an efficient method to forecast multi-step-ahead pan evaporation. It is also noted that Boruta-extra-tree-based models show more accurate results as compared to the XGBoost-based models, thus encouraging that the Boruta-BiLSTM model is a sophisticated, intelligent model. But for future work, some further recommendations and suggestions can be considered.

Deep learning models have proved to be very effective and powerful; however, due to their black-box nature, some limitations restrict their competence and cannot provide model's prediction explainability and interpretability. Thus, the integration of explainable AI models such as Local Interpretable Model-Agnostic Explanations (LIME) [79] and Shapley Additive explanations (SHAP) [80] models can be an interesting direction for future research. Moreover, the hybridization of Boruta-BiLSTM with physical models can be considered an alternative area that can explain the model's physics-based perspective and viewpoint. Besides, the Boruta-BiLSTM model can be mixed with Bayesian Model Averaging [81] and bootstrapping [82] techniques to enhance the forecasting skills as well as tackle the model uncertainties arising in the learning process.

Another possible approach can be signal-based decomposition methods, which can offer novelty in overcoming noise, non-stationarity, and non-linearity. Consequently, MVMD [83] can be a better approach to upsurge the accuracy of the Boruta-BiLSTM model as it simultaneously captures the non-stationary and non-linearity within the data by undertaking the mode mixing issues [84]. The results approved that the Boruta-BiLSTM model achieves higher precision to forecast multi-step-ahead pan evaporation against other comparing models. The Boruta-BiLSTM model can be potentially applied in renewable and sustainable energy, hydrology, agriculture, and other climate change areas of interest.

5. Conclusion

This paper proposes a novel Boruta-BiLSTM model for multi-step-ahead (i.e., 1-, 3-, 5-, 7-, 10-, 15-, and 30-d) pan evaporation forecasting for Ahvaz and Yazd stations. The proposed approach uses the Boruta model for feature selection of the BiLSTM. The model was compared with the XGBoost-BiLSTM model that uses XGBoost for feature selection and Boruta-LSTM, Boruta-GRNN, Boruta-AdaBoost,

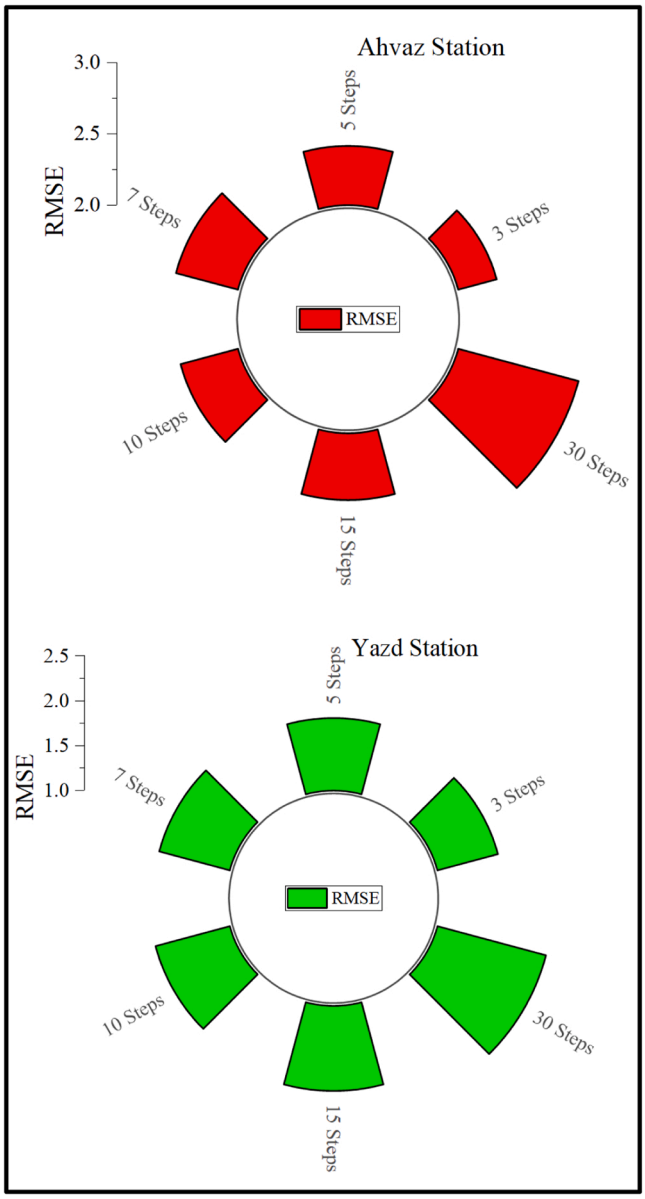

Fig. 16. RMSE for multi-step-ahead pan evaporation forecasting in Ahvaz (Top) and Yazd (Bottom) stations.

XGBoost-LSTM, XGBoost-GRNN, and XGBoost-AdaBoost models as benchmarking models. Evaluations were performed using standard goodness-of-fit metrics. The results revealed that the Boruta-BiLSTM achieved the highest accuracy for multi-step-ahead pan evaporation forecasting at both Ahvaz and Yazd stations: (R = 0.9396, MAPE = 25.3541, RMSE = 1.6857, $U_{95\%}$ = 4.6684, KGE = 0.8972, IA = 0.9670) for Ahvaz station, and (R = 0.9535, MAPE = 15.2988, RMSE = 1.3996, $U_{95\%}$ = 3.8804, KGE = 0.9408, IA = 0.9759) for Yazd station. Notably, the proposed Boruta-BiLSTM outperforms the XGBoost-BiLSTM and other XGBoost-based models. The Boruta-BiLSTM model can be extended to other sectors, such as environment studies, hydrology, agriculture, and renewable energy, to facilitate enhanced decision-making.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Research for this paper was carried out under the KICT Research Program (project no. 20230115–001, Development of IWRM-Korea Technical Convergence Platform Based on Digital New Deal) funded by the Ministry of Science and ICT and in part supported by Korea Environmental Industry & Technology Institute (KEITI), funded by Korea Ministry of Environment (MOE) (No. 2022003610001 & 2022003640001).

References

[1] J.L. Monteith, Evaporation and environment, *in*: Symp. Soc. Exp. Biol., Cambridge University Press (CUP) Cambridge, 1965: pp. 205–234.
[2] C. Jarmain, C.S. Everson, M.J. Savage, M.G. Mengistu, A.D. Clulow, S. Walker, M. B. Gush, Refining tools for evaporation monitoring in support of water resources management, Water Res. Comm. Rep. (2009) 8.
[3] A. Ertek, S. Şensoy, I. Gedik, C. Küçükyumuk, Irrigation scheduling based on pan evaporation values for cucumber (Cucumis sativus L.) grown under field conditions, Agric. Water Manag. 81 (2006) 159–172.
[4] A. Seifi, M. Ehteram, F. Soroush, A.T. Haghighi, Multi-model ensemble prediction of pan evaporation based on the Copula Bayesian Model Averaging approach, Eng. Appl. Artif. Intell. 114 (2022), 105124.
[5] P.S. Shirgure, G.S. Rajput, Evaporation modeling with neural networks – a research review, Int. J. Res. Rev. Soft Intell. Comput. (2011).
[6] W. Jing, Z.M. Yaseen, S. Shahid, M.K. Saggi, H. Tao, O. Kisi, S.Q. Salih, N. Al-Ansari, K.-W. Chau, Implementation of evolutionary computing models for reference evapotranspiration modeling: short review, assessment and possible future research directions, Eng. Appl. Comput. Fluid Mech. 13 (2019) 811–823.
[7] A. Ashrafzadeh, M.A. Ghorbani, S.M. Biazar, Z.M. Yaseen, Evaporation process modelling over northern Iran: application of an integrative data-intelligence model with the krill herd optimization algorithm, Hydrol. Sci. J. (2019).
[8] W. Brutsaert, Some exact solutions for nonlinear desorptive diffusion, Z. Für Angew. Math. Und Phys. 33 (1982) 540–546.
[9] R.K. Singh, L. Ganapathi, P. Tiwari, J. Narayan, Effect of processing geometry in oxygen incorporation and insitu formation of YBa2Cu3O7 superconducting thin films by pulsed laser evaporation technique, Appl. Phys. Lett. 55 (1989) 2351–2353.
[10] F.I. Morton, Studies in evaporation and their lessons for the environmental sciences, Can. Water Resour. J. 15 (1990) 261–286, https://doi.org/10.4296/cwrj1503261.
[11] E. Sartori, A critical review on equations employed for the calculation of the evaporation rate from free water surfaces, Sol. Energy (2000), https://doi.org/10.1016/S0038-092X(99)00054-7.
[12] F.I. Morton, Evaporation research—a critical review and its lessons for the environmental sciences, Crit. Rev. Environ. Sci. Technol. 24 (1994) 237–280.
[13] U.S. Panu, T. Nguyen, Estimation of mean areal evaporation in northwestern Ontario, Can. Water Resour. J. 19 (1994) 69–82.
[14] K. Khosravi, P. Daggupati, M.T. Alami, S.M. Awadh, M.I. Ghareb, M. Panahi, B. T. Pham, F. Rezaie, C. Qi, Z.M. Yaseen, Meteorological data mining and hybrid data-intelligence models for reference evaporation simulation: a case study in Iraq, Comput. Electron. Agric. 167 (2019), 105041.
[15] D. Han, A. Moghaddamnia, Reply to comments on "Evaporation estimation using artificial neural networks and adaptive neurofuzzy inference system techniques" by A. Moghaddamnia, M. Ghafari Gousheh, J. Piri, S. Amin and D. Han [Adv. Water Resour. 32 (2009) 88–97], Adv. Water Resour. 32 (2009) 967–968, https://doi.org/10.1016/j.advwatres.2009.02.012.
[16] C.H.B. Priestley, R.J. Taylor, On the assessment of the surface heat flux and evaporation using large-scale parameters, Mon. Weather Rev. 100 (1972) 81–92.
[17] J.F. Griffiths, Another evaporation formula, Agric. Meteorol. 3 (1966) 257–261.
[18] H.L. Penman, Natural evaporation from open water, bare soil and grass, Proc. R. Soc. A Math. Phys. Eng. Sci. 193 (1948) 120–145, https://doi.org/10.1098/rspa.1948.0037.
[19] R.C. Deo, M. Şahin, Application of the extreme learning machine algorithm for the prediction of monthly Effective Drought Index in eastern Australia, Atmos. Res. 153 (2015) 512–525.
[20] A.D. Mehr, O. Akdegirmen, Estimation of urban imperviousness and its impacts on flashfloods in Gazipaşa, Turkey, Knowl. -Based Eng. Sci. 2 (2021) 9–17.
[21] A. Malik, A. Kumar, S. Kim, M.H. Kashani, V. Karimi, A. Sharafati, M.A. Ghorbani, N. Al-Ansari, S.Q. Salih, Z.M. Yaseen, Modeling monthly pan evaporation process over the Indian central Himalayas: application of multiple learning artificial intelligence model, Eng. Appl. Comput. Fluid Mech. 14 (2020) 323–338.
[22] L. Diop, A. Bodian, K. Djaman, Z.M. Yaseen, R.C. Deo, A. El-shafie, L.C. Brown, The influence of climatic inputs on stream-flow pattern forecasting: case study of Upper Senegal River, Environ. Earth Sci. 77 (2018), https://doi.org/10.1007/s12665-018-7376-8.
[23] M. Ali, R. Prasad, Y. Xiang, Z.M. Yaseen, Complete ensemble empirical mode decomposition hybridized with random forest and kernel ridge regression model for monthly rainfall forecasts, J. Hydrol. 584 (2020), 124647.
[24] R. Tur, S. Yontem, A comparison of soft computing methods for the prediction of wave height parameters, Knowl. -Based Eng. Sci. 2 (2021) 31–46.

[25] O. Kisi, B. Choubin, R.C. Deo, Z.M. Yaseen, Incorporating synoptic-scale climate signals for streamflow modelling over the Mediterranean region using machine learning models, Hydrol. Sci. J. (2019).
[26] S.R. Naganna, B.H. Beyaztas, N. Bokde, A.M. Armanuos, On the evaluation of the gradient tree boosting model for groundwater level forecasting, Knowl. -Based Eng. Sci. 1 (2020) 48–57.
[27] M.A. Ghorbani, M.A. Jabehdar, Z.M. Yaseen, S. Inyurt, Solving the pan evaporation process complexity using the development of multiple mode of neurocomputing models, Theor. Appl. Climatol. (2021), https://doi.org/10.1007/s00704-021-03724-8.
[28] H. Moeeni, H. Bonakdari, Impact of normalization and input on ARMAX-ANN model performance in suspended sediment load prediction, Water Resour. Manag. 32 (2018) 845–863, https://doi.org/10.1007/s11269-017-1842-z.
[29] A. Sharafati, S.B.H.S. Asadollah, A. Neshat, A new artificial intelligence strategy for predicting the groundwater level over the Rafsanjan aquifer in Iran, J. Hydrol. (2020), https://doi.org/10.1016/j.jhydrol.2020.125468.
[30] A. Sharafati, R. Yasa, H.M. Azamathulla, Assessment of stochastic approaches in prediction of wave-induced pipeline scour depth, J. Pipeline Syst. Eng. Pract. 9 (2018), https://doi.org/10.1061/(ASCE)PS.1949-1204.0000347.
[31] D. Myronidis, K. Ioannou, D. Fotakis, G. Dörflinger, Streamflow and hydrological drought trend analysis and forecasting in cyprus, Water Resour. Manag. (2018), https://doi.org/10.1007/s11269-018-1902-z.
[32] K.P. Sudheer, A.K. Gosain, D. Mohana Rangan, S.M. Saheb, Modelling evaporation using an artificial neural network algorithm, Hydrol. Process. 16 (2002) 3189–3202.
[33] L. Wu, G. Huang, J. Fan, X. Ma, H. Zhou, W. Zeng, Hybrid extreme learning machine with meta-heuristic algorithms for monthly pan evaporation prediction, Comput. Electron. Agric. 168 (2020), 105115, https://doi.org/10.1016/j.compag.2019.105115.
[34] O. Kisi, A. Mirboluki, S.R. Naganna, A. Malik, A. Kuriqi, M. Mehraein, Comparative evaluation of deep learning and machine learning in modelling pan evaporation using limited inputs, Hydrol. Sci. J. (2022).
[35] M. Fu, T. Fan, Z.I.A.N.G. Ding, S.Q. Salih, N. Al-ansari, Z.M. Yaseen, Deep learning data-intelligence model based on adjusted forecasting window scale, Appl. Dly. Streamflow Simul. 8 (2020).
[36] J. Schmidhuber, Deep Learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117, https://doi.org/10.1016/j.neunet.2014.09.003.
[37] A. Halevy, P. Norvig, F. Pereira, The unreasonable effectiveness of data, IEEE Intell. Syst. 24 (2009) 8–12, https://doi.org/10.1109/mis.2009.36.
[38] T. Tiyasha, T.M. Tung, Z.M. Yaseen, Deep learning for prediction of water quality index classification: tropical catchment environmental assessment, Nat. Resour. Res. (2021) 1–20.
[39] M. Sit, B.Z. Demiray, Z. Xiang, G.J. Ewing, Y. Sermet, I. Demir, A comprehensive review of deep learning applications in hydrology and water resources, Water Sci. Technol. (2020).
[40] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Netw. 5 (1994) 157–166, https://doi.org/10.1109/72.279181.
[41] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.
[42] F. Kratzert, D. Klotz, C. Brenner, K. Schulz, M. Herrnegger, Rainfall-runoff modelling using Long Short-Term Memory (LSTM) networks, Hydrol. Earth Syst. Sci. (2018), https://doi.org/10.5194/hess-22-6005-2018.
[43] B.B. Sahoo, R. Jha, A. Singh, D. Kumar, Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting, Acta Geophys 67 (2019) 1471–1481.
[44] F. Kratzert, D. Klotz, G. Shalev, G. Klambauer, S. Hochreiter, G. Nearing, Benchmarking a catchment-aware long short-term memory network (LSTM) for large-scale hydrological modeling, Hydrol. Earth Syst. Sci. Discuss. (2019), https://doi.org/10.5194/hess-2019-368.
[45] S. Tang, F. Sun, W. Liu, H. Wang, Y. Feng, Z. Li, Optimal postprocessing strategies with LSTM for global streamflow prediction in ungauged basins, Water Resour. Res. (2023) e2022WR034352.
[46] B. Li, R. Li, T. Sun, A. Gong, F. Tian, M.Y.A. Khan, G. Ni, Improving LSTM hydrological modeling with spatiotemporal deep learning and multi-task learning: A case study of three mountainous areas on the Tibetan Plateau, J. Hydrol. (2023), https://doi.org/10.1016/j.jhydrol.2023.129401.
[47] P.K. Pareek, C. Srinivas, B.N. Jagadeesh, Prediction of Rainfall in Karnataka Region using optimised MVC-LSTM Model, in: 2023 IEEE Int. Conf. Integr. Circuits Commun. Syst., IEEE, 2023: pp. 1–8.
[48] J. Hou, Y. Wang, J. Zhou, Q. Tian, Prediction of hourly air temperature based on CNN–LSTM, Geomat., Nat. Hazards Risk. 13 (2022) 1962–1986.
[49] J. Liu, X. Huang, Q. Li, Z. Chen, G. Liu, Y. Tai, Hourly stepwise forecasting for solar irradiance using integrated hybrid models CNN-LSTM-MLP combined with error correction and VMD, Energy Convers, Manag 280 (2023), 116804.
[50] D. Zhang, G. Lindholm, H. Ratnaweera, Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring, J. Hydrol. 556 (2018) 409–418.
[51] J. Zhang, Y. Zhu, X. Zhang, M. Ye, J. Yang, Developing a long short-term memory (LSTM) based model for predicting water table depth in agricultural areas, J. Hydrol. (2018), https://doi.org/10.1016/j.jhydrol.2018.04.065.
[52] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Adv. Neural Inf. Process. Syst., 2014: pp. 3104–3112.
[53] M.B. Kursa, W.R. Rudnicki, Feature selection with the Boruta package, J. Stat. Softw. 36 (2010) 1–13.
[54] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (2006) 3–42.
[55] M. Seyyedattar, M.M. Ghiasi, S. Zendehboudi, S. Butt, Determination of bubble point pressure and oil formation volume factor: Extra trees compared with LSSVM-CSA hybrid and ANFIS models, Fuel 269 (2020), 116834, https://doi.org/10.1016/j.fuel.2019.116834.
[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn, Mach. Learn. Python, J. Mach. Learn. Res 12 (2011) 2825–2830.
[57] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proc. 22nd Acm Sigkdd Int. Conf. Knowl. Discov. Data Min., 2016: pp. 785–794.
[58] H. Zheng, J. Yuan, L. Chen, Short-Term Load Forecasting Using EMD-LSTM neural networks with a xgboost algorithm for feature importance evaluation, Energies 10 (2017), https://doi.org/10.3390/en10081168.
[59] C. Chen, H. Shi, Z. Jiang, A. Salhi, R. Chen, X. Cui, B. Yu, DNN-DTIs: Improved drug-target interactions prediction using XGBoost feature selection and deep neural network, Comput. Biol. Med. 136 (2021), 104676, https://doi.org/10.1016/j.compbiomed.2021.104676.
[60] C. Chen, Q. Zhang, B. Yu, Z. Yu, P.J. Lawrence, Q. Ma, Y. Zhang, Improving protein-protein interactions prediction accuracy using XGBoost feature selection and stacked ensemble classifier, Comput. Biol. Med. 123 (2020), 103899, https://doi.org/10.1016/j.compbiomed.2020.103899.
[61] H.K. Cigizoglu, Application of Generalized Regression Neural Networks to Intermittent Flow Forecasting and Estimation, J. Hydrol. Eng. 10 (2005) 336–341, https://doi.org/10.1061/(ASCE)1084-0699(2005)10:4(336).
[62] L. Ding, P. Rangaraju, A. Poursaee, Application of generalized regression neural network method for corrosion modeling of steel embedded in soil, Soils Found. 59 (2019) 474–483, https://doi.org/10.1016/j.sandf.2018.12.016.
[63] M. Firat, M.E. Turan, M.A. Yurdusev, Comparative analysis of neural network techniques for predicting water consumption time series, J. Hydrol. 384 (2010) 46–51, https://doi.org/10.1016/j.jhydrol.2010.01.005.
[64] M. Jamei, I. Ahmadianfar, I.A. Olumegbon, M. Karbasi, A. Asadi, On the assessment of specific heat capacity of nanofluids for solar energy applications: Application of Gaussian process regression (GPR) approach, J. Energy Storage (2021) 102067.
[65] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1997) 119–139.
[66] M. Jamei, M. Karbasi, O.A. Alawi, H.M. Kamar, K.M. Khedher, S.I. Abba, Z. M. Yaseen, Earth skin temperature long-term prediction using novel extended Kalman filter integrated with Artificial Intelligence models and information gain feature selection, Sustain. Comput. Inform. Syst. 35 (2022), 100721.
[67] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, API design for machine learning software: experiences from the scikit-learn project, ArXiv Prepr. ArXiv1309.0238. (2013).
[68] A. El Bilali, A. Taleb, Y. Brouziyne, Groundwater quality forecasting using machine learning algorithms for irrigation purposes, Agric. Water Manag. 245 (2021), 106625.
[69] U.K. Singh, M. Jamei, M. Karbasi, A. Malik, M. Pandey, Application of a modern multi-level ensemble approach for the estimation of critical shear stress in cohesive sediment mixture, J. Hydrol. 607 (2022) 127549.
[70] K.U. Jaseena, B.C. Kovoor, Decomposition-based hybrid wind speed forecasting model using deep bidirectional LSTM networks, Energy Convers. Manag. 234 (2021), 113944, https://doi.org/10.1016/j.enconman.2021.113944.
[71] A. Bhattarai, D. Qadir, A.M. Sunusi, B. Getachew, A.R. Mallah, Dynamic Sliding Window-Based Long Short-Term Memory Model Development for Pan Evaporation Forecasting, Knowl. -Based Eng. Sci. 4 (2023) 37–54.
[72] S. Elsayed, M. Gupta, G. Chaudhary, S. Taneja, H. Gaur, M. Gad, M. Hamdy Eid, A. Kovács, S. Péter, A. Gaagai, U. Schmidhalter, Interpretation the Influence of Hydrometeorological Variables on Soil Temperature Prediction Using the Potential of Deep Learning Model, Knowl. -Based Eng. Sci. 4 (2023) 55–77, https://doi.org/10.51526/kbes.2023.4.1.55-77.
[73] Y. Dai, Q. Zhou, M. Leng, X. Yang, Y. Wang, Improving the Bi-LSTM model with XGBoost and attention mechanism: A combined approach for short-term power load prediction, Appl. Soft Comput. (2022), 109632.
[74] M. Schuster, K.K. Paliwal, Bidirectional Recurrent Neural Networks, IEEE Trans. SIGNAL Process. 45 (1997) 2673–2681.
[75] R.V. Joseph, A. Mohanty, S. Tyagi, S. Mishra, S.K. Satapathy, S.N. Mohanty, A hybrid deep learning framework with CNN and Bi-directional LSTM for store item demand forecasting, Comput. Electr. Eng. 103 (2022), 108358.
[76] H.V. Gupta, H. Kling, K.K. Yilmaz, G.F. Martinez, Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling, J. Hydrol. 377 (2009) 80–91.
[77] M. Karbasi, M. Jamei, M. Ali, A. Malik, Z.M. Yaseen, Forecasting weekly reference evapotranspiration using Auto Encoder Decoder Bidirectional LSTM model hybridized with a Boruta-CatBoost input optimizer, Comput. Electron. Agric. 198 (2022), 107121, https://doi.org/10.1016/j.compag.2022.107121.
[78] M. Jamei, M. Ali, A. Malik, M. Karbasi, E. Sharma, Z.M. Yaseen, Air quality monitoring based on chemical and meteorological drivers: Application of a novel data filtering-based hybridized deep learning model, J. Clean. Prod. 374 (2022), 134011, https://doi.org/10.1016/j.jclepro.2022.134011.
[79] S. Mishra, B.L. Sturm, S. Dixon, Local interpretable model-agnostic explanations for music content analysis., in: ISMIR, 2017: pp. 537–543.
[80] F. Ghafarian, R. Wieland, D. Lüttschwager, C. Nendel, Application of extreme gradient boosting and Shapley Additive explanations to predict temperature regimes inside forests from standard open-field meteorological data, Environ. Model. Softw. 156 (2022), 105466.

[81] J.M. Sloughter, T. Gneiting, A.E. Raftery, Probabilistic wind speed forecasting using ensembles and Bayesian model averaging, J. Am. Stat. Assoc. 105 (2010) 25–35.
[82] S. Mouatadid, J.F. Adamowski, M.K. Tiwari, J.M. Quilty, Coupling the maximum overlap discrete wavelet transform and long short-term memory networks for irrigation flow forecasting, Agric. Water Manag. 219 (2019) 72–85, https://doi.org/10.1016/j.agwat.2019.03.045.
[83] N. ur Rehman, H. Aftab, Multivariate variational mode decomposition, IEEE Trans. Signal Process. 67 (2019) 6039–6052.
[84] F. Gao, X. Shao, A novel interval decomposition ensemble model for interval carbon price forecasting, Energy 243 (2022), 123006.