Environmental Science

https://www.aimspress.com/journal/environmental

AIMS Environmental Science, 12(5): 770–794.
DOI: 10.3934/environsci.2025034
Received: 11 December 2024
Revised: 03 July 2025
Accepted: 28 July 2025
Published: 14 August 2025

Research article

Improving potential evapotranspiration prediction accuracy using advanced machine learning models: a case study in Ahvaz, Iran

Jamshid Bani Feri[1], Aslan Egdernezhad[2,*], Ali Mokhtaran[3], Mahdi Asadilour[4] and Davoud Khodadadi Dehkordi[5]

[1] Ph.D. student, Department of Water Sciences and Engineering, Ahv.C., Islamic Azad University, Ahvaz, Iran; jamshidbaniferi@iau.ac.ir

[2] Assistant Professor, Department of Water Sciences and Engineering, Ahv.C., Islamic Azad University, Ahvaz, Iran; a_eigder@iau.ac.ir

[3] Assistant Professor., Agricultural Engineering Research Department, Khuzestan Agricultural and Natural Resources Research and Education Center, Agricultural Research Education and Extension Organization (AREEO), Ahwaz, Iran; alimokhtaran@gmail.com

[4] Assistant Professor, Department of Water Sciences and Engineering, Ahv.C., Islamic Azad University, Ahvaz, Iran; mehdi.asadilour@iau.ac.ir

[5] Assistant Professor, Department of Water Sciences and Engineering, Ahv.C., Islamic Azad University, Ahvaz, Iran; korkori@iau.ac.ir

* **Correspondence:** Email: a_eigder@iau.ac.ir.

Abstract: Accurate forecasting of daily potential evapotranspiration (ET_o) is vital for efficient irrigation scheduling, sustainable water resource management, and optimal crop yield, especially in arid regions. In this study, we present a novel, large-scale comparison of three advanced machine learning models, Support Vector Machine (SVM), Artificial Neural Network (ANN), and Gene Expression Programming (GEP), for ET_o prediction at the Ahvaz synoptic station in Iran. The work is distinctive in using one of the longest ET_o datasets available (1979–2023; 16,084 records), incorporating a sensitivity analysis for input selection, and applying the Developed Discrepancy Ratio (DDR) as an advanced performance metric. The ANN model (MLP 4-9-1 architecture) demonstrated the highest prediction accuracy, achieving an R^2 of 0.9806, RMSE of 0.4122, and DDRmax of 3.27 in the training phase, and an R^2 of 0.9779, RMSE of 0.4327, and DDR_{max} of 3.22 during validation. In comparison, SVM and GEP models showed lower accuracy across all phases. These results highlight the superior capability of the ANN model for ET_o forecasting and its potential as a reliable tool for irrigation planning and water resource management in arid regions like Ahvaz.

Keywords: climate; artificial intelligence; dry region; water cycle; performance assessment

1. Introduction

Water is the lifeblood of our planet, and its efficient management is crucial for sustainable agriculture, ecosystem health, and human consumption. One of the key components of the water cycle is evapotranspiration (ET), the process by which plants release water vapor into the atmosphere. Accurate ET prediction is essential to optimize irrigation systems, predict crop yields, and mitigate the impacts of droughts and floods. However, traditional methods of estimating ET are often labor-intensive, costly, and prone to errors. This is where machine learning models come in, offering a powerful tool for predicting ET with unprecedented accuracy and precision. By leveraging the power of MLMs, it can unlock new insights into the complex relationships between climate, soil, and plant variables, and develop more effective strategies for managing our precious water resources. In this paper, we delve into the exciting world of daily potential ET (ET_o) prediction using MLMs, and explore the innovative approaches and techniques that are revolutionizing the field of hydrology. Given the critical importance of accurately forecasting changes in weather conditions and the significant role of models in simulating and predicting key parameters within the hydrological cycle, numerous studies have been undertaken to evaluate and compare the performance of various predictive models across diverse research domains [1–3].

Reference	Involved model(s)	Predicted variable	Country	Findings
[4]	RF, MARS	Dew point temperature	Iran	The MARS model showed precise outputs.
[5]	ANN-AR, ANN-RF, ANN-REPtree, ANN-M5P, ANN-Bagging	ET_o	India	ANN-M5P showed the highest accuracy
[6]	ANN	ET_o	Greece	Accurate output was obtained.
[7]	MVMD-RR, MVMD-KELM, MVMD-RR-KELM	ET_o	India	Their hybrid MVMD-RR-KELM model was superior.
[8]	AR, MA, ARMA, ARIMA, LSSVM, ANFIS, GRNN	ET_o	Iran	The ARMA model was more accurate.
[9]	FFNN, RBFNN, GEP	ET_o		The FFNN model emerged as the best
[10]	GRNN	T	India	The model had the best output.
[11]	MLP, Empirical equations	Solar radiation		The model provided accurate outcome.
[12]	ANN, ANFIS, SVM, Empirical equations	Air temperature	Turkey	ANFIS outperformed other the others.
[13]	MGGP, Empirical equations	Solar radiations	Turkey	The MGGP was more successful.

Reference	Involved model(s)	Predicted variable	Country	Findings
[14]	ANNs, SVMs, ANN-PA, SVM-PA	Daily solar radiation		The hybrid SVM-PA exhibited more accurate.
[15]	SVM, MLP, GRNN, CCNN, GMDH	ET_o	Pakistan	SVM was the most efficient.
[16]	XGBoost, MLP, M5 model tree, XGBoost-GWO	ET_o	China	XGBOOST-GWO outperformed the others.
[17]	ANN, ANFIS-GP, ANFIS-SC, GRNN, GEP, MARS	Wind speed	Iran	GRNN showed the highest predictive performance
[18]	ANN, ANN-GP, Empirical equations	Solar radiation	Iran	ANN-GP model had accurate output.
[19]	GEP, empirical equations	ET_o	Iran, Spain	GEP was the best.
[20]	SVM, GEP, MARS, and empirical models	ET_o	Iran	MARS and SVM-RBF models showing superior performance
[21]	ANN, SVR	ET_o	India	SVR had more accurate outputs.
[22]	ANN, GEP, WR, and empirical models	Solar radiation		ANNs outperformed other models
[23]	GEP, Empirical equations	ET_o	Spain	GEP yielded the best results
[24]	ANN, ANFIS, GEP	ET_o	Iran	GEP showed precise outcome.
[25]	GEP, ANFIS	ET_o	Iran	GEP was more accurate.
[26]	GEP	ET_o	Burkina Faso	The most efficient outputs were obtained using GEP.
[27]	HYDRUS-1D, ANN, GP		Canada	GP model was the accurate model.
[28]	LGP, GEP	ET_o	Iran	LGP was more precise than GEP.
[29]	GA, GRNN, GRNN-GA	ET_o	Korea	Hybrid GRNN-GA achieved the highest performance.

Notes: Adaptive Neuro-Fuzzy Inference System (ANFIS); Random Forest (RF); Multivariate Adaptive Regression Splines (MARS); Artificial Neural Network (ANN); Multivariate Variational Mode Decomposition (MVMD) With Ridge Regression (RR) And Kernel Extreme Learning Machine (KELM); Feed Forward Neural Network (FFNN), Radial Basis Function Neural Network (RBFNN), Gene Expression Programing (GEP); Generalized Regression Neural Network (GRNN); Multi-Gene Genetic Programming (MGGP); Procrustes Analysis (PA); Extreme Gradient Boosting (Xgboost); Grey Wolf Optimizer (GWO) Algorithm; Linear Genetic Programming (LGP); Group Method Of Data Handling (GMDH); Cascade Correlation Neural Network (CCNN); Subtractive Clustering (SC)

As highlighted, ET_o, as an integral component of the hydrological water cycle, plays a pivotal role in water resources management. A comprehensive review of literature underscores the significance of precise estimation and prediction of ET_o, garnering considerable attention from researchers. Consequently, diverse methodologies have been evaluated to achieve accurate predictions and estimations. Notably, recent advancements in climatology have witnessed the emergence of MLMs as viable tools for this purpose. These models, capable of discerning intricate and latent relationships between input and output variables without necessitating specialized expertise, offer promising avenues for accurate predictions. In this context, the Ahvaz synoptic station, in southwest Iran, represents a key site for studying ET_o estimation due to its hot, arid climate and critical dependence on efficient water management for agriculture. Here, we explore the efficacy of three MLMs, including SVM, GEP, and ANN, in estimating ET_o at this representative location.

2. Materials and methods

2.1. Case study

In this study, data from the Ahvaz synoptic station in Iran, spanning the statistical years 1979 to 2023, were utilized. The Ahvaz station is at a latitude of 31.3442° and a longitude of 48.7442°. The elevation of this station is 22.5 meters above sea level, and it is characterized by a dry climate. The station's location is depicted in Figure 1. The daily recorded statistics at this station include minimum, maximum, and average temperatures (T_{min}, T_{max}, and T_{ave}, respectively), minimum, maximum, and average humidity levels (RH_{min}, RH_{max}, and RH_{ave}, respectively), wind speed (WS), the number of sunny hours (SHs), precipitation amounts (P), and the reference values of ET_o. A summary of the statistical properties of each measured parameter is provided in Table 1. Table 2 presents the correlation coefficients among the variables measured at the station. Evidently, there is a positive correlation between ET_o and temperature, a negative correlation between ET_o and humidity, a positive correlation between ET_o and wind speed, and a negative correlation between ET_o and precipitation.

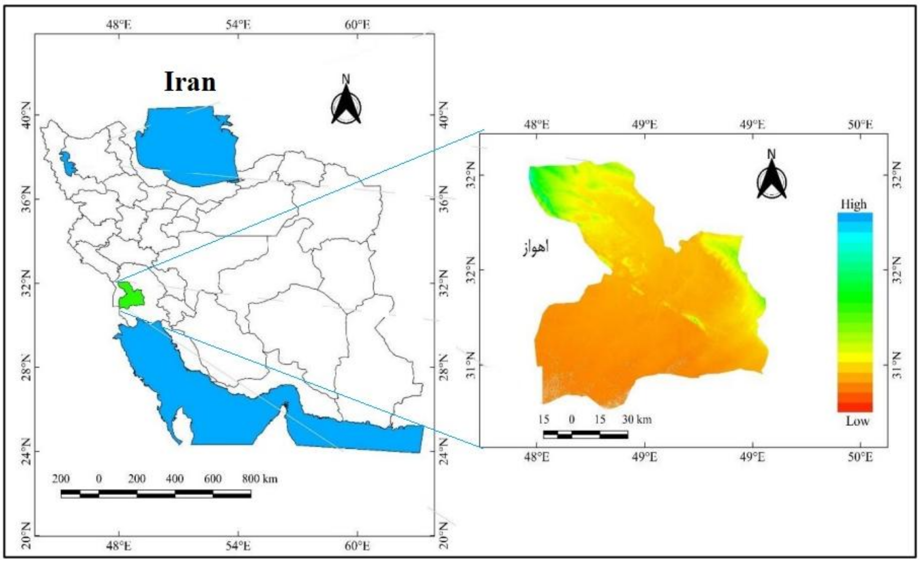

Figure 1. The geographical coordinates and elevation of the station under study.

Table 1. A synopsis of the statistical properties for each measured parameter.

Data	Property						
	N total	Mean	Standard Deviation	Sum	Minimum	Median	Maximum
T(min)	16084	18.95	8.31	304816.70	-1.20	19.50	37.00
T(max)		33.52	10.92	539168.42	0.00	34.60	53.70
T(ave)		26.24	9.47	421992.31	0.00	27.00	43.75
RH(min)		61.64	22.78	991456.47	0.00	61.63	100.00
RH(max)		25.51	18.05	410271.29	0.00	19.00	98.00
RH(ave)		43.58	19.36	700863.88	0.00	40.50	99.00
WS		1.51	0.85	24306.74	0.00	1.40	6.54
SH		8.56	3.43	137628.83	0.00	9.60	14.01
P		0.64	3.87	10299.10	0.00	0.00	100.00
ETo		5.11	2.95	82233.12	0.56	4.80	16.06

Table 2. The correlation coefficient matrix of measured values.

		T_{min}	T_{max}	T_{ave}	$Rh_{(min)}$	$Rh_{(max)}$	$Rh_{(ave)}$	WS	SH	P	ET_o
T_{min}	Pearson Corr.	1.0000	0.9369	0.9794	-0.7191	-0.6343	-0.7187	0.3289	0.4220	-0.1056	0.8509
T_{max}	Pearson Corr.	0.9369	1.0000	0.9881	-0.7795	-0.7810	-0.8226	0.2327	0.5841	-0.1908	0.8641
T_{ave}	Pearson Corr.	0.9794	0.9881	1.0000	-0.7653	-0.7290	-0.7900	0.2786	0.5222	-0.1564	0.8720
$Rh_{(min)}$	Pearson Corr.	-0.7191	-0.7795	-0.7653	1.0000	0.7963	0.9594	-0.3019	-0.5100	0.2194	-0.7767
$Rh_{(max)}$	Pearson Corr.	-0.6343	-0.7810	-0.7290	0.7963	1.0000	0.9346	-0.1866	-0.6594	0.3398	-0.7269
$Rh_{(ave)}$	Pearson Corr.	-0.7187	-0.8226	-0.7900	0.9594	0.9346	1.0000	-0.2646	-0.6073	0.2875	-0.7957
WS	Pearson Corr.	0.3289	0.2327	0.2786	-0.3019	-0.1866	-0.2646	1.0000	0.0513	0.0407	0.5860
SH	Pearson Corr.	0.4220	0.5841	0.5222	-0.5100	-0.6594	-0.6073	0.0513	1.0000	-0.2548	0.5896
P	Pearson Corr.	-0.1056	-0.1908	-0.1564	0.2194	0.3398	0.2875	0.0407	-0.2548	1.0000	-0.1712
ET_o	Pearson Corr.	0.8509	0.8641	0.8720	-0.7767	-0.7269	-0.7957	0.5860	0.5896	-0.1712	1.0000

2.2. An introduction to MLMs involved

2.2.1. Overview of ANNs

ANNs represent a class of computational models inspired by the structure and function of biological neural networks in the human brain. ANNs are widely used in various fields, including pattern recognition, classification, regression, and time-series prediction. The structure of an ANN consists of interconnected processing units called neurons or nodes, organized in layers. The three major types of layers in an ANN are the input layer, hidden layer(s), and output layer. The mathematical representation of an ANN involves the propagation of signals through the network and the adjustment of weights to minimize errors during training. The forward propagation is a simple feedforward neural network, where signals propagate from the input layer through one or more hidden layers to the output layer. During forward propagation, the input signals x1, x2, …, xn are passed through the network, and the weighted sum of inputs is computed for each neuron in the hidden layers and output layer. This process is mathematically represented as:

$$z_j=\sum \omega_{ij}x_i+b_j \tag{1}$$

where z_j is the weighted sum at neuron j, ω_{ij} is the weight connecting input neuron i to hidden neuron j, x_i is the input signal at neuron i, and b_j is the bias term for neuron j. The weighted sum is then passed through an activation function f, which introduces non-linearity into the network. Common activation functions include sigmoid, tanh, and ReLU (Rectified Linear Unit):

$$a_j=f(z_j) \tag{2}$$

The outputs a_j of the activation functions become the input signals for the next layer. After forward propagation, the output of the network is compared to the true target values, and the error is calculated using a loss function such as mean squared error (MSE):

$$E=\frac{1}{N}\sum (y_i-\hat{y}_i)^2 \tag{3}$$

where y_i is the true target value and $\hat{y}_i$ is the predicted value. Backpropagation is then used to update the weights of the network to minimize this error. The gradients of the loss function with respect to the weights are computed using the chain rule of calculus and are used to update the weights through optimization algorithms like stochastic gradient descent (SGD) or Adam. The following are the simulation steps of the ANN: (i) Initialization: Initialize the weights and biases of the network randomly or using predefined strategies; (ii) Forward Propagation: Pass the input data through the network to obtain predictions; (iii) Compute the error between the predicted and actual values using a suitable loss function; (iv) Compute the gradients of the loss function with respect to the weights and biases using backpropagation; (v) Weight Update: Update the weights and biases of the network using optimization algorithms to minimize the error; and (vi) Iteration: Repeat steps 2–5 for a specified number of iterations or until convergence criteria are met. By iteratively adjusting the weights and biases based on the error signal, ANNs can learn complex patterns and relationships in data, enabling them to make accurate predictions and classifications in various applications (Fuladipanah et al., [30]; Sajindra et al. [31]; Jayathilake et al. [32]; Azamathulla et al. [33]).

2.2.2. Overview of GEP

Gene Expression Programming (GEP) is a powerful evolutionary algorithm used for symbolic regression and function optimization tasks. It combines principles from genetics and computational biology to evolve computer programs that solve complex problems. GEP operates by evolving populations of symbolic expressions represented as linear chromosomes, where each chromosome consists of multiple genes encoding sub-expressions or terminals. A GEP chromosome is composed of a fixed-length string of genes, with each gene encoding a segment of the program. These genes are then translated into hierarchical structures known as expression trees, representing the computer programs. Expression trees consist of nodes representing functions and terminals, with functions performing operations and terminals representing variables, constants, or other input data. Each gene within the chromosome encodes a segment of the expression, which can be a function or a terminal symbol. Mathematically, this can be represented as:

$$G_i=(s_{i1},s_{i2},\cdots, s_{in}) \tag{4}$$

where G_i is the ith gene, and s_{ij} represents the jth symbol in the gene. The genes are translated into expression trees, where the hierarchical structure is determined by the sequence of genes. Mathematically, this process involves parsing the genes according to a predefined grammar to generate the expression tree. The fitness of each individual in the population is evaluated based on its ability to solve the problem at hand. This typically involves executing the evolved program and comparing its output to a target value using an appropriate fitness function. The simulation steps are as following: (i) Initialization: Generate an initial population of individuals, each represented by a chromosome encoding a symbolic expression; (ii) Evaluation: Evaluate the fitness of each individual in the population by executing its corresponding expression and comparing the output to the target values using a fitness function; (iii) Selection: Select individuals from the population to serve as parents for the next generation based on their fitness. Common selection methods include tournament selection and roulette wheel selection; (iv) Recombination: Apply genetic operators such as crossover and mutation to the selected parents to produce offspring for the next generation. Crossover involves exchanging genetic material between parents, while mutation introduces random changes to the offspring's genes; (v) Replacement: Replace some individuals in the current population with the offspring to maintain the population size; and (vi) Termination: Repeat steps 2–5 for a specified number of generations or until a termination condition is met, such as reaching a satisfactory fitness level or a predefined number of generations. By iteratively evolving the population through selection, recombination, and mutation, GEP can efficiently search the solution space and discover symbolic expressions that accurately model the underlying relationship in the data (Leon et al. [34]; Fuladipanah et al. [35]; Gharehbaghi et al. [36]; Azamathulla et al. [37]).

2.2.3. Overview of the SVM

Developed by Cortes and Vapnik [38], the Support Vector Machine (SVM) is a robust supervised learning algorithm employed for classification and regression tasks. It functions by identifying the optimal hyperplane that separates data points into distinct classes while maximizing the margin between them. Given a set of training data ($\mathbf{X}_i$, y_i) where $\mathbf{X}_i$ denotes the input features and y_i represents the corresponding class label (for classification) or target value (for regression), SVM strives to find the optimal hyperplane described by the equation:

$$W^{T}X+b=0 \tag{5}$$

Here, **W** is the weight vector orthogonal to the hyperplane, b is the bias term, and **X** denotes the input features. For linearly separable data, the optimization problem can be expressed as:

$$\min_{W,b}\frac{1}{2}||\mathbf{W}||^{2} \tag{6}$$

subject to:

$$y_{i}(\mathbf{W}^{\mathbf{T}}\mathbf{X}_{\mathbf{i}}+b)\geq 1 \tag{7}$$

For data that are not linearly separable, SVM employs the kernel trick to map the input features into a higher-dimensional space where the data points become linearly separable. The revised optimization problem is:

$$\min_{W,b}\frac{1}{2}||\mathbf{W}||^{2}+C\sum_{i=1}^{N}\xi_{i} \tag{8}$$

subject to:

$$y_{i}(\mathbf{W}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{X}_{\mathbf{i}})+b)\geq 1-\xi_{i} \tag{9}$$

$$\xi_{i}\geq 0 \tag{10}$$

where $\mathbf{\Phi}(\mathbf{X}_{\mathbf{i}})$ is the feature mapping function, C is the regularization parameter, and ξ_{i} are the slack variables. The simulation steps of the SVM are as following: (i) Data Preprocessing: Prepare the training dataset by scaling the features and encoding the class labels or target values; (ii) Model Training: Choose a kernel function (e.g., linear, polynomial, radial basis function) and tune the hyperparameters (e.g., regularization parameter C, kernel parameters) using cross-validation, and train the SVM model on the training data; (iii) Model Evaluation: Assess the performance of the trained model on a separate validation dataset using appropriate metrics; (iv) Model Tuning: Adjust the model hyperparameters based on the validation results to enhance performance; and (v) Model Deployment: Once satisfactory performance is attained, deploy the trained SVM model to make predictions on new, unseen data. By following these simulation steps, SVM can efficiently classify or regress data by finding the optimal hyperplane or decision boundary that maximizes the margin between classes or minimizes the error for regression tasks (Kumar et al. [39]; Fuladipanah et al. [35]; Rathnayake et al. [40]).

2.3. Evaluation metrics

In the realm of predictive modeling and statistical analysis, quantifying the accuracy and reliability of models is paramount. Several metrics are widely used to evaluate the performance of models, each providing different insights into the model's strengths and weaknesses. Among these, RMSE, MAE, and R^2 are commonly utilized. Additionally, DDR is an advanced metric offering a nuanced assessment of model performance. The RMSE is a measure of the differences between predicted and observed values. It is particularly sensitive to large errors, making it useful for detecting models with occasional large deviations. The RMSE is defined mathematically as:

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i\text{-}\hat{y}_i)^2} \tag{11}$$

where y_i represents the observed value for the ith data point, $\hat{y}_i$ denotes the predicted value for the ith data point, and n signifies the total number of data points. RMSE is expressed in the same units as the observed values, which aids in the interpretability of the results. The MAE quantifies the average magnitude of errors in a set of predictions without considering their direction. It provides a straightforward measure of prediction accuracy. The MAE is given by:

$$\mathrm{MAE}=\frac{1}{n}\sum_{i=1}^{n}\left|y_i\text{-}\hat{y}_i\right| \tag{12}$$

where | | denotes the absolute value function. The MAE is also expressed in the same units as the data, making it directly interpretable. The R^2 assesses the proportion of variance in the dependent variable that is predictable from the independent variables. It provides an indication of the goodness-of-fit of a model. The R^2 value ranges from 0 to 1, where 1 indicates perfect prediction and 0 indicates no explanatory power. Mathematically, R^2 is defined as:

$$R^2=1\text{-}\frac{\sum_{i=1}^{n}(y_i\text{-}\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i\text{-}\bar{y})^2} \tag{13}$$

where $\bar{y}$ is the mean of the observed values. A higher R^2 value indicates a better fit of the model to the data. In addition to traditional evaluation metrics such as RMSE, MAE, and R^2, the Developed Discrepancy Ratio (DDR) index was used to provide a more nuanced assessment of model performance. The DDR index, proposed by Noori et al. [41], is a statistical tool designed to evaluate the degree of agreement between observed and predicted values by comparing the distribution of absolute errors to a theoretical distribution derived from the Gaussian (normal) function. The key advantage of the DDR lies in its ability to highlight both central tendency and spread of prediction errors, capturing not only average performance but also the frequency and magnitude of larger deviations. The DDR is calculated based on the probability density function (PDF) of standardized residuals, enabling a visual and quantitative comparison of prediction error distributions. By transforming the residuals into a standardized normal space, the DDR enables the identification of how well the prediction error pattern aligns with an ideal normal distribution centered around zero. This is particularly helpful for assessing the robustness and consistency of machine learning models across operational conditions. DDR_{max}, or the maximum value of the discrepancy ratio, represents the highest concentration of errors at a particular standardized value. It serves as an indicator of the peak alignment (or misalignment) between the predicted and observed datasets. A higher DDR_{max} value typically reflects better model performance, indicating that a larger proportion of prediction errors are concentrated near the zero-error line. Conversely, a lower DDR_{max} suggests that errors are more widely scattered, indicating weaker model reliability. In this study, the DDR and DDR_{max} metrics were used alongside RMSE, MAE, and R^2 to evaluate the predictive strength of the machine learning models. This approach ensures a comprehensive understanding of not only average error magnitude but also the distribution and extremity of prediction errors, which is particularly important in hydrological modeling, where localized errors can have significant implications for decision-making. The DDR index is calculated as follows:

$$DDR=\frac{\text{Predicted value}}{\text{Observed value}}-1 \tag{14}$$

As mentioned, in order to enhance interpretation and visualization, it is essential to convert the Gaussian distribution of DDR values into a standard normal distribution. This process involves a two-step approach. Initially, the DDR values (referred to as variable x) are standardized, resulting in the calculation of the normalized DDR value ($x_{(DDR)}$) using a Gaussian function. Following this, a visual representation is generated, where $x_{(DDR)}$ values are compared with their standardized counterparts (Z_{DDR}). In the graphical representation of Z_{DDR} versus $x_{(DDR)}$, a greater alignment of error distribution towards the central tendency and higher $x_{(DDR)}$ values signify improved accuracy.

2.4. Sensitivity analysis

According to Koncar [42], the Gamma test constitutes a non-parametric statistical technique utilized for output estimation by identifying the optimal input-output dataset configurations based on minimal mean square error values. This technique is proposed as an appropriate method for determining the most efficacious combination of various input variables to precisely characterize the output. In this methodology, the dataset is represented as $\{(x_i,y_i), 1\leq i\leq M)\}$, where the input vectors $x_i\in R^m$ are m dimensional vectors, and the corresponding outputs $y_i\in R$ are scalars. The input vector x influences the output y. The relationship between the input and output variables is defined by the following equation:

$$y=Gx+\Gamma \tag{15}$$

Here, G and Γ denote the gradient and the intercept of the regression line where x=0, respectively, and y is the output. Lower values of G and Γ suggest that the respective input variables are more appropriate. In addition to these two criteria, an indicator denoted as $\text{V-Ratio}=\frac{\Gamma}{\sigma^2(y)}$, where Γ represents the gamma function and $\sigma^2(y)$ is the output variance, is utilized to identify the optimal input parameters. The values of the V-Ratio range from 0 to 1. A V-Ratio value closer to zero for each input parameter indicates the effectiveness of that particular input. The various combinations of input variables have been delineated following the format introduced by Mask (Malik et al. [43]). Table 2 encompasses nine parameters, with the mask representation employing an eight-digit binary sequence corresponding to these variables: T_{min}, T_{max}, T_{ave}, RH_{min}, RH_{max}, RH_{ave}, WS, SHs, and P, respectively. Within this representation, the binary digits '1' and '0' denote the inclusion or exclusion of a parameter. For instance, '11111111' denotes the inclusion of all parameters, while '01111111' indicates the exclusion of the first parameter, T_{min}, from the sensitivity analysis. As previously indicated, the optimal model is distinguished by the lowest values of Γ, G, and V-Ratio. Several permutations were conducted to assess the sensitivity of the parameters involved in the ET_o simulation. An illustrative instance of this analysis, comprising 25 alternatives, is delineated in Table 3. Based on the findings, the optimal and most efficacious input configuration for forecasting and simulating ET_o consists of the digit combination 001010110 (the raw of no. 19). In essence, in employing MLMs, the pertinent parameters encompass T_{ave}, RH_{min}, WS, and HS. While Figure 2-i illustrates the variation of Γ-σ for all combinations, a scatter plot of Γ and σ for the opted and optimized 001010110 combination is presented in Figure 2-ii.

Table 3. Examining Sensitivity in Model Input Parameters.

No.	Γ	G	Standard Error	V-Ratio	Mask
1	0.297427	0.053459	0.015277	0.034097	111010111
2	0.254336	0.06074	0.012875	0.029157	011010111
3	0.258858	0.060585	0.013181	0.029676	101010111
4	0.218563	0.11027	0.008443	0.025056	011010110
5	0.263779	0.071748	0.01422	0.03024	101001111
6	0.221845	0.110694	0.009479	0.025432	101010110
7	0.266067	0.068812	0.016094	0.030502	011001111
8	0.223812	0.069941	0.010285	0.025658	001010111
9	0.225671	0.083949	0.011749	0.025871	001001111
10	0.272215	0.089098	0.012347	0.031207	111000111
11	0.272409	0.084766	0.011072	0.031229	111010110
12	0.321378	0.059071	0.016096	0.036843	111001111
13	0.234125	0.115905	0.010518	0.02684	101001110
14	0.235387	0.111817	0.011329	0.026985	011001110
15	0.235348	0.08894	0.014931	0.02698	010001111
16	0.287649	0.055306	0.017668	0.032976	110010111
17	0.245963	0.190428	0.008102	0.028197	111000110
18	0.242758	0.074345	0.012789	0.02783	010010111
19	0.198678	0.146447	0.006912	0.022777	001010110
20	0.29955	0.086101	0.012207	0.034341	111001110
21	0.350276	0.050867	0.016257	0.040156	011101111
22	0.201923	0.155379	0.006354	0.023149	001001110
23	0.252361	0.083384	0.010932	0.028931	110001110
24	0.306033	0.061923	0.016555	0.035084	110001111
25	0.255086	0.092136	0.009796	0.029243	101000111

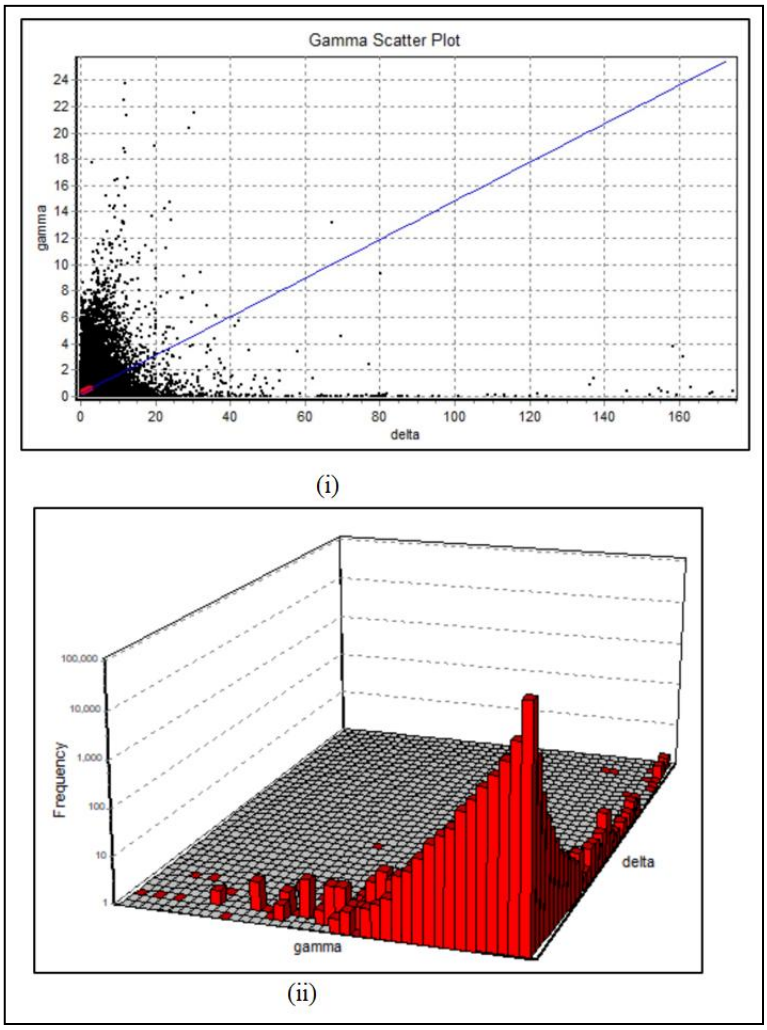

Figure 2. Outcomes of Γ-test, (i) Γ vs. σ; (ii) a 3D histogram of Γ-σ-Frequency for the 001010110.

3. Results and discussion

3.1. Models' performance metrics

Upon determining the effective input parameters, the ANN, SVM, and GEP models were each implemented using a data partitioning strategy comprising 70% for training (11,260 data points), 15% for testing (2,412 data points), and 15% for validation (2,412 data points). The performance evaluation of these three models was conducted utilizing the performance metrics outlined in the preceding section. The overall results of the simulation, based on the optimal parameter configurations for each of the three machine learning models, are presented in Table 4.

Table 4. A summary of the statistical performance of the MLMs involved.

Model name	Training phase				
	RMSE	MAE	R^2	DDR max	Bias
SVM	0.9026	0.6604	0.9138	2.5773	-0.0513
GEP	0.7048	0.5591	0.9444	1.5953	0.0647
MLP (4-9-1)	0.4122	0.3207	0.9806	3.2694	0.0004
Model name	Testing phase				
SVM	1.0654	0.7882	0.8824	2.19	-0.1265
GEP	0.7048	0.5640	0.9443	1.59	0.0516
MLP (4-9-1)	0.4308	0.3333	0.9789	3.29	-0.0120
Model name	Validation phase				
SVM	0.9213	0.6790	0.9216	2.64	0.2153
GEP	1.0501	0.7918	0.8863	2.25	-0.0260
MLP (4-9-1)	0.4327	0.3342	0.9779	3.22	0.0083

3.2. SVM's outputs

The SVM model, optimized with an RBF kernel (C=19, ε=0.1, γ=0.25), demonstrated moderate performance across all phases. During training, it achieved reasonable accuracy but exhibited a slight tendency toward underestimation. This underestimation became more pronounced in the testing phase, where the model showed increased error magnitudes and reduced stability compared to training. In validation, while predictive consistency partially recovered relative to testing, the model maintained higher errors than other phases and displayed a shift to systematic overestimation. Throughout all phases, SVM maintained moderate explanatory power for ET_o variability, though its error distribution indicated less reliability in extreme value prediction compared to ANN.

3.3. GEP's outputs

The GEP model, configured with the parameters detailed in Table 5, demonstrated reliable performance during the training and testing phases, showing consistent accuracy in modeling ET_o. The evolved expression tree incorporated mathematical functions such as cosine, arctangent, square root, cube root, and basic arithmetic operators, reflecting the model's flexibility in capturing complex relationships. However, the model's accuracy declined during the validation phase, indicating its reduced ability to generalize to unseen data. This outcome may be attributed to the complexity of the symbolic expression generated, which, while effective in fitting training data, led to overfitting and decreased predictive reliability when applied to new observations. The tree expression of the GEP model output is depicted in Figure 3, where the constants in the equation are G1C0=0.563538, G1C1=-0.381042, G2C0=-7.928467, G2C1=-0.861634, G3C0=4.249603, and G3C1= -7.235016.

Table 5. Tuning parameters of the GEP.

Parameters	Values
Head size	8
Chromosomes numbers	45
Number of genes	3
Mutation rate	0.044
Inversion rate	0.1
One-point recombination rate	0.3
Two-point recombination rate	0.3
Gene recombination rate	0.1
Gene transposition rate	0.1
IS transposition rate	0.1
RIS transposition rate	0.1
Fitness function error type	RMSE
Linking function	+

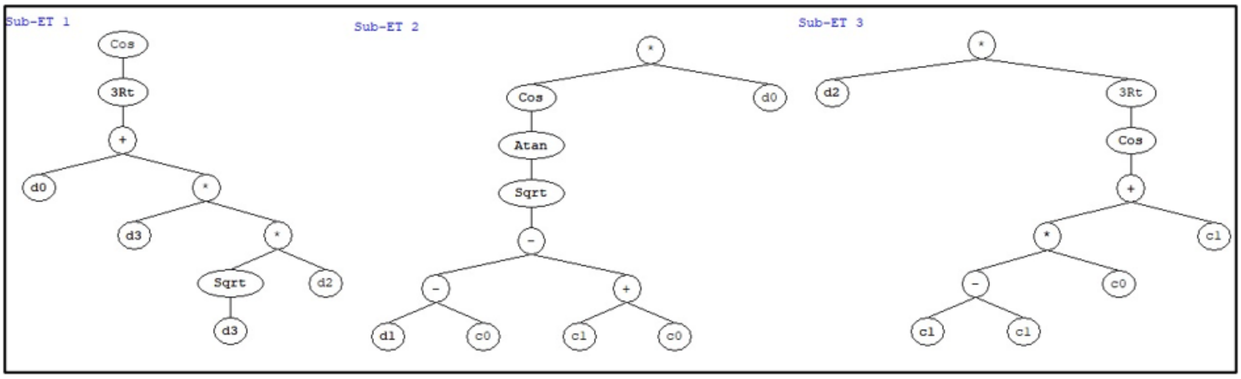

Figure 3. Expression tree of the GEP model.

3.4. ANN's outputs

The MLP model with a 4-9-1 architecture, utilizing Tanh and Identity activation functions for the hidden and output layers, respectively, and optimized with the BFGS 358 algorithm, achieved the best overall performance among the models tested. It provided highly accurate predictions and maintained strong agreement with observed ETo values throughout the training, testing, and validation phases. The model's consistent performance across these phases highlights its capacity to capture the complex nonlinear relationships between the climatic variables and ETo, and its ability to generalize effectively to new data. This robustness makes the MLP model a reliable tool for ETo prediction in arid climates like that of Ahvaz. Based on the performance indices RMSE, MAE, R^2, and DDR, the MLP 4-9-1 model achieved consistently better results across all phases. For example, in the validation phase, the MLP achieved an R^2 of 0.9779, compared to 0.9216 for SVM and 0.8863 for GEP. Similarly, RMSE values were 0.4327 for MLP, 0.9213 for SVM, and 1.0501 for GEP, indicating the MLP model provided more accurate predictions. These differences demonstrate the superior predictive capability of the MLP model over SVM and GEP.

3.5. Graphical performance criteria

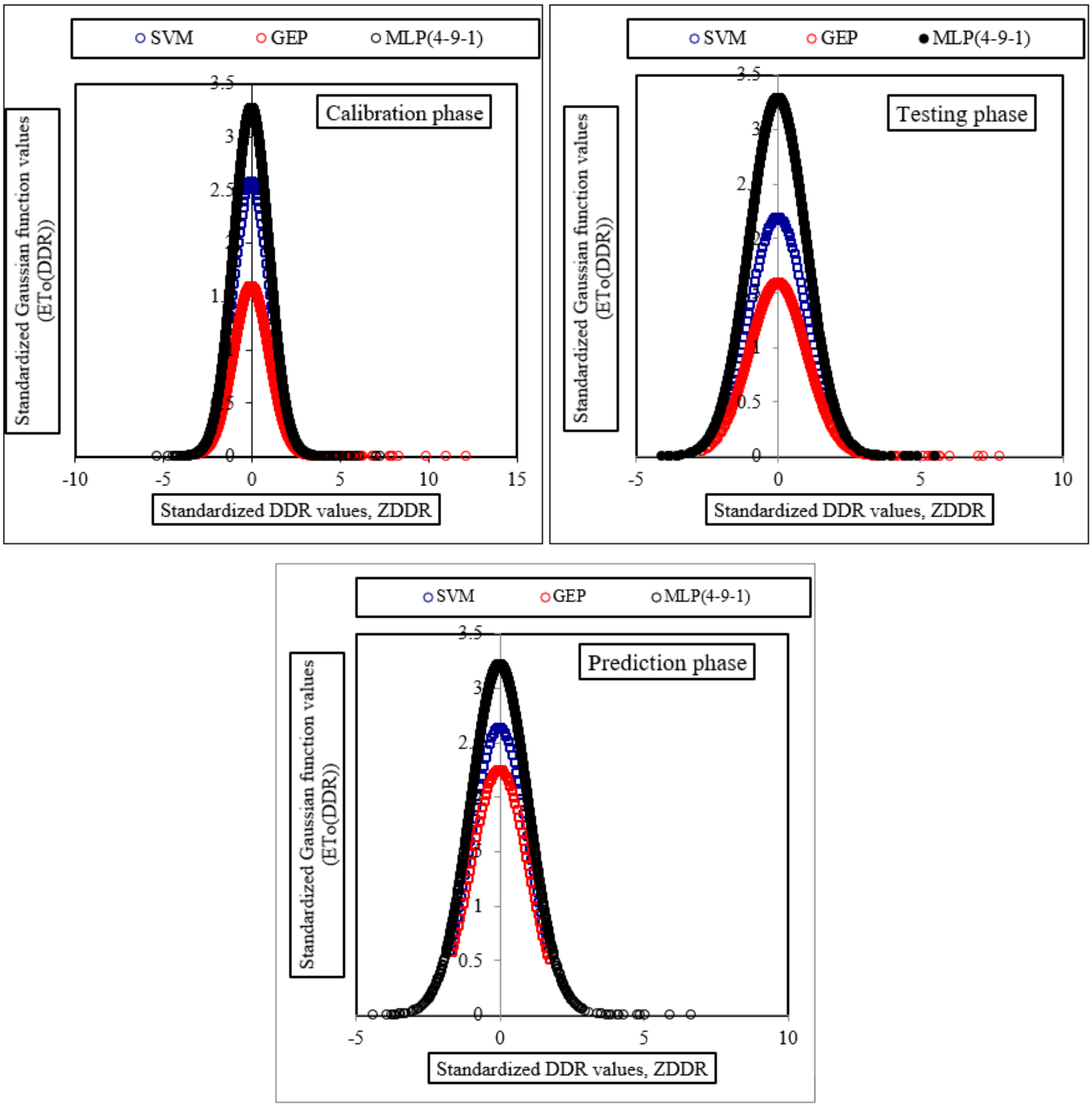

Figure 4. The distribution of the DDR index for involved MLMs.

To assess model performance through a graphical depiction, Figure 4 and Figure 5 were generated. Figure 4 illustrates the models' functionality based on the DDR index. Evidently, across all phases of training, testing, and prediction, the MLP 4-9-1 model exhibits notably superior performance compared to SVM and GEP, as indicated by its significantly higher peak point on the curve. The SVM model occupies a secondary or intermediate position in the DDR distribution curve, denoted by its blue coloration, while the GEP model is relegated to the lowest rank. These graphical findings align with the numerical results presented in Table 4. Figure 5 displays the scatter plot of observed versus predicted data, serving as a visual representation of the Coefficient of Determination (R^2) index. The inclusion of a 1:1 line (the bisector of the first quadrant of the coordinate plane) aids in assessing model performance; closer alignment of data points to this line signifies superior model performance. It is evident from the graphs that the MLP 4-9-1 model consistently outperforms the other two models. Figure 6 illustrates the distribution of the measured and the MLMs' output data. Box plots of measured and predicted values are also included.

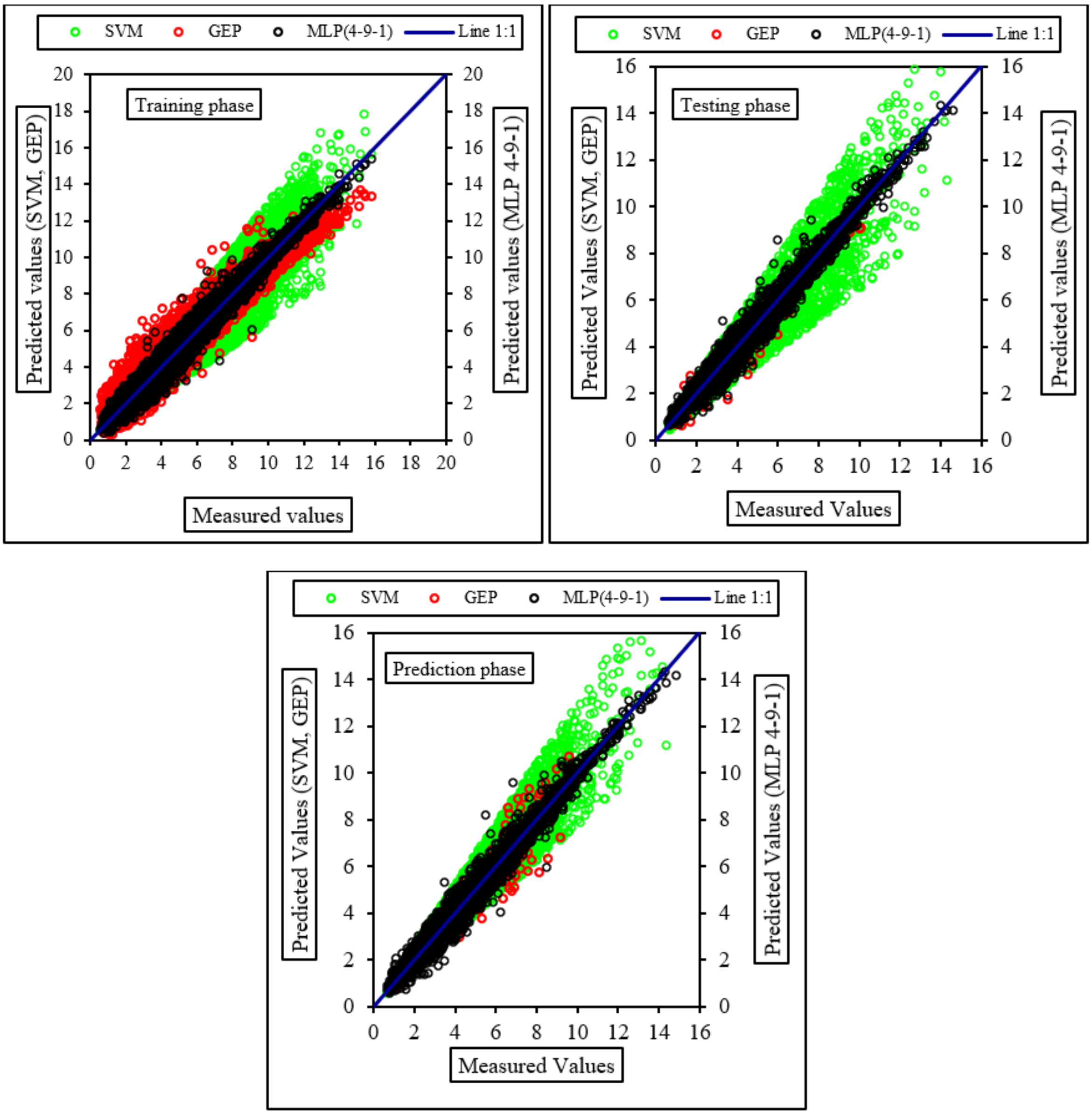

Figure 5. Scatter plot of the MLMs' outputs involved vs. measured data.

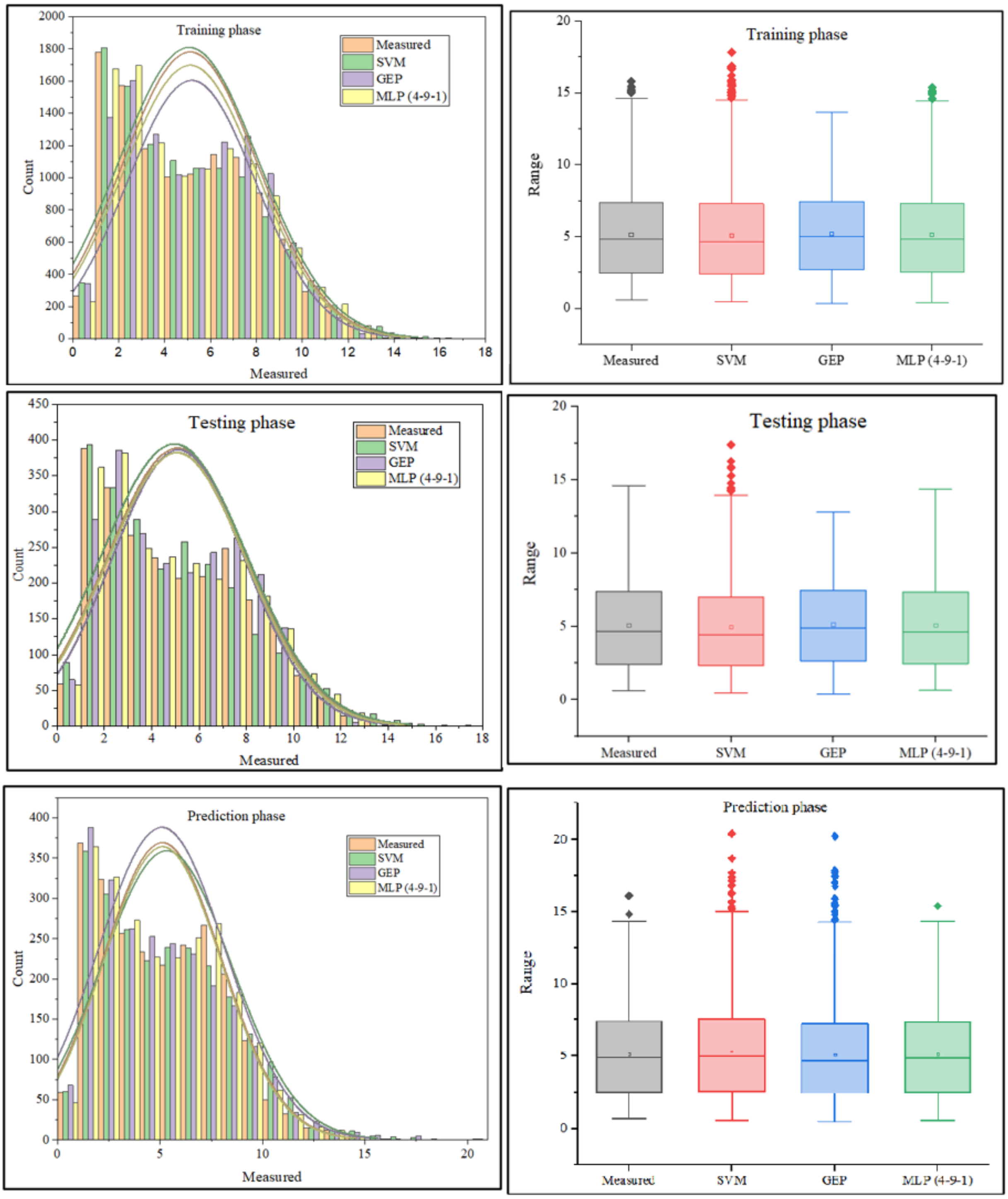

Figure 6. Distribution of the measured vs. outputs of the MLMs involved.

The performance of the MLMs are presented using the Taylor Diagram in Figure 7. The Taylor diagram visually assesses how well models or datasets compare to a reference so that (i) the radial distance from the origin represents its standard deviation, and (ii) the azimuthal angle represents its correlation coefficient with the reference. The diagram enables quick visual comparison and quantification of skill scores. In Figure 7, Taylor diagrams across training, testing, and prediction stages show the MLP 4-9-1 model consistently aligns closely with observational data. This proximity demonstrates the MLP model's superior performance in replicating the reference data compared to other models.

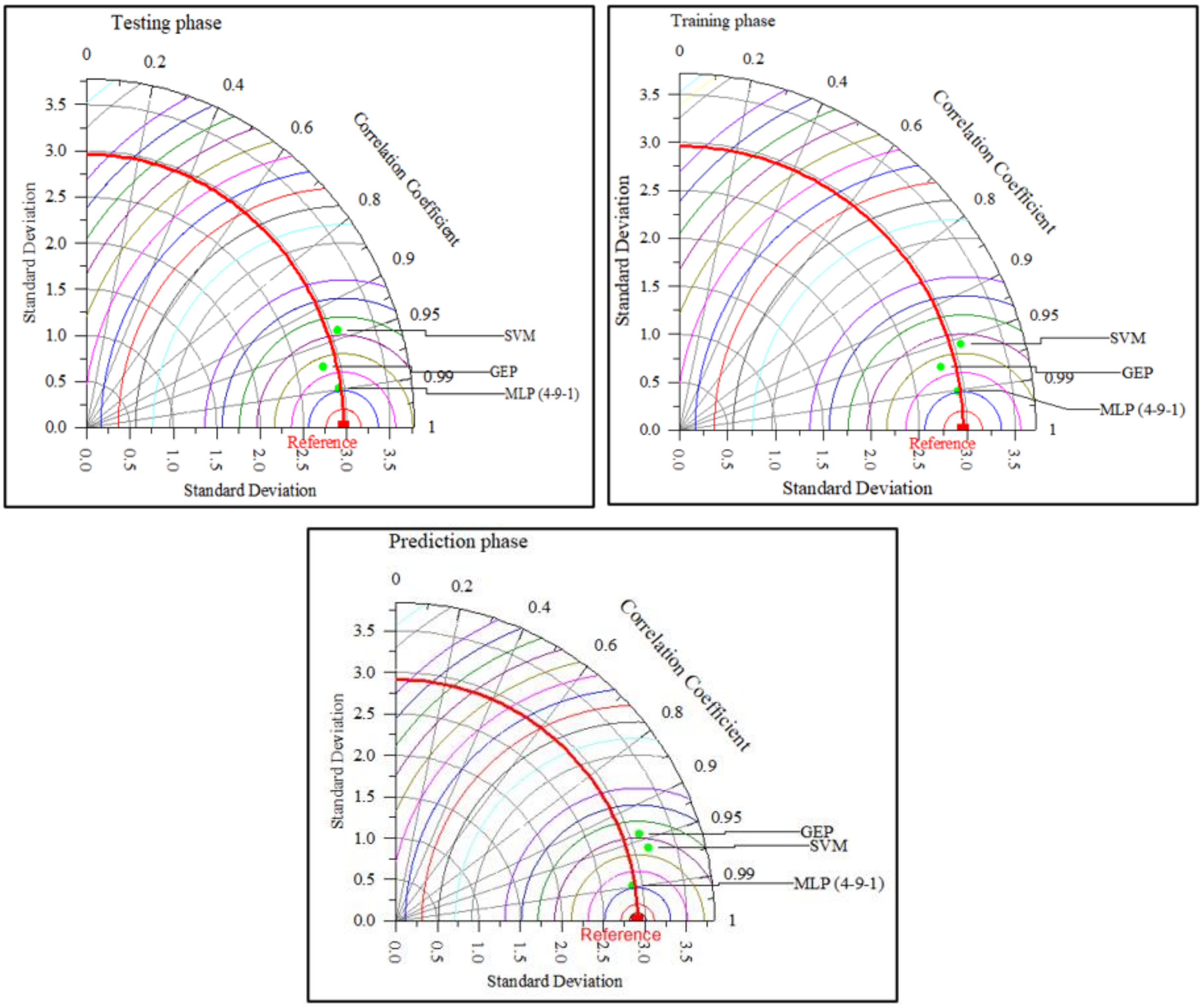

Figure 7. The Taylor diagram of the MLMs' outputs.

Figure 8 illustrates the alignment between the simulation results generated by the Multilayer Perceptron (MLP) during the prediction phase and the empirical data. It is evident that there is a high degree of concordance between the two time series, despite discrepancies that predominantly appear at the extremities.

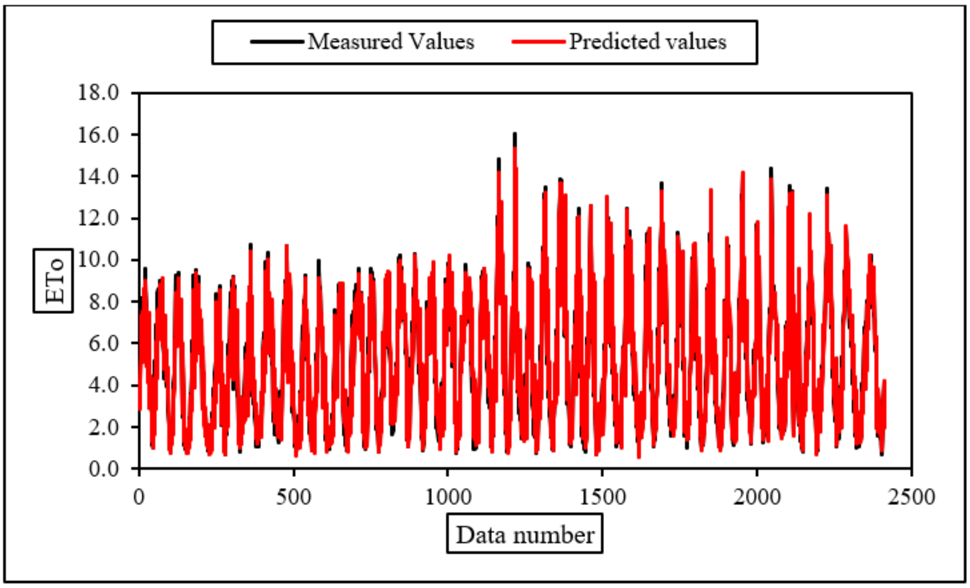

Figure 8. Measured vs. predicted values of ET_o during the validation phase.

It can be discussed that the superior performance of the MLP model in this study can be attributed to several interrelated factors that enhance its capability in modeling complex hydrological phenomena, ET_o. First and foremost, the MLP model is inherently well-suited for capturing highly nonlinear relationships between input variables (e.g., temperature, humidity, wind speed, and sunshine hours) and the target output (ET_o). Given the complex and dynamic interactions among climatic parameters in arid environments like Ahvaz, linear or weakly nonlinear models often fail to detect subtle dependencies. In contrast, the layered structure of MLP, combined with nonlinear activation functions (e.g., Tanh), enables it to approximate arbitrary nonlinear functions with high precision, as supported by the universal approximation theorem. Moreover, MLPs are generally more robust to noise in large datasets when appropriately regularized and trained. In this study, the performance of the MLP model remained consistent across training, testing, and validation datasets, suggesting strong generalization ability and limited overfitting. The use of the BFGS optimization algorithm further enhanced convergence and reduced the likelihood of becoming trapped in local minima, which can degrade performance in noisy or high-dimensional settings. On the other hand, the relatively lower performance of the GEP model may be partly due to overfitting during the training phase. As an evolutionary algorithm, GEP is highly flexible in fitting training data but is also prone to generating overly complex symbolic expressions, especially in the absence of strong constraints on expression tree size or depth. This can result in models that perform well on training data but generalize poorly to unseen data, as indicated by the drop in performance during the validation phase. Additionally, SVM-while effective at identifying optimal hyperplanes in transformed feature space- may be less adaptable to capturing highly nonlinear or intricate time-dependent patterns when compared to MLP, particularly in the presence of interacting climate variables with non-Gaussian distributions. In summary, the MLP model outperformed GEP and SVM due to its superior ability to model nonlinear relationships, robust optimization framework, and strong generalization capacity, making it particularly well-suited for forecasting ET_o in complex, data-rich, and environmentally variable regions such as Ahvaz.

4. Conclusion

We investigated the use of three MLMs, namely MLP, SVM, and GEP, to predict daily ET_o at the Ahvaz synoptic station. All models successfully captured the relationship between climatic variables and ET_o, demonstrating their potential for hydrological modeling in arid environments. Although all three MLMs demonstrated acceptable accuracy in predicting daily ET_o values, the MLP model consistently showed higher performance. For instance, in the prediction phase, the MLP achieved an R^2 of 0.9779, outperforming SVM (0.9216) and GEP (0.8863), with correspondingly lower RMSE (0.4327 for MLP versus 0.9213 for SVM and 1.0501 for GEP). These statistical comparisons highlight the advantage of using the MLP model for ETo prediction in this study. Overall, given the increasing demand for efficient water use in arid regions like Ahvaz, the superior performance of the MLP model highlights its value for supporting irrigation scheduling and water resource planning under variable climate conditions.

The superior performance of the ANN model, specifically the MLP (4-9-1) architecture, can be attributed to its inherent ability to capture complex nonlinear relationships between climatic inputs and ET_o. Unlike other MLMs, MLP's multi-layered structure, combined with nonlinear activation functions such as Tanh, enables it to approximate arbitrary nonlinear functions with high precision, as supported by the universal approximation theorem. This property is particularly valuable in hydrological modeling, where the relationships between temperature, humidity, wind speed, and solar radiation are often non-additive, seasonally variable, and highly interactive. Furthermore, the MLP model demonstrated strong generalization capability, evidenced by the consistent performance across training, testing, and validation phases, which suggests effective learning of the underlying data structure without overfitting. This robustness can be partially credited to the use of the BFGS optimization algorithm, which promotes smooth convergence and avoids entrapment in local minima during training. Additionally, the ANN's structure enables it to implicitly optimize the interactions among multiple input variables, something that models like GEP (which evolve symbolic expressions) or SVM (which map inputs to high-dimensional feature spaces) handle less flexibly. While SVM and GEP have their respective strengths, such as GEP's interpretability and SVM's margin maximization, neither offers the same level of adaptive learning and noise tolerance in high-volume, nonlinear datasets as the MLP. These factors collectively explain why the ANN model consistently outperformed the other two models in capturing the complex dynamics of ET_o under arid climate conditions.

While our results of demonstrate the high predictive performance of the MLP model compared to SVM and GEP for ET_o forecasting, several limitations must be acknowledged. First, like most machine learning models, MLPs require large volumes of high-quality, well-distributed data to ensure optimal performance. In regions with sparse or inconsistent climatic data, the model may underperform due to insufficient representation of local environmental variability. Second, although the MLP architecture captures complex nonlinear relationships, it operates as a black box, meaning the internal reasoning behind its predictions is not easily interpretable. This lack of transparency can be a drawback for decision-makers who prefer models with explicit functional forms or physically based formulations. Third, while the MLP model generalized well within the data from Ahvaz, its transferability to other geographic regions with different climatic regimes may be limited. Retraining and hyperparameter adjustment would be required for application in new locations, which can be computationally expensive and time-consuming. Moreover, model performance can be sensitive to the choice of architecture (e.g., number of hidden layers, neurons), activation functions, and optimization algorithms. Without careful tuning, MLPs may overfit or underfit the data. Similarly, evolutionary algorithms like GEP can be prone to overfitting if expression tree complexity is not adequately controlled, while SVMs may require kernel selection and parameter tuning that are not always straightforward. These limitations highlight the importance of combining MLMs with domain knowledge and using model ensembles or hybrid approaches in future research to improve accuracy, interpretability, and robustness under varying data conditions.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors of this research have no any conflict of interest.

References

1. Braddock RD, Kremmer ML, Sanzogni, L (1998) Feed-forward artificial neural network model for forecasting rainfall run-off. *Environmetrics* 9: 419–432. https://doi.org/10.1002/(SICI)1099-095X(199807/08)9:4<419::AID-ENV312>3.0.CO;2-D
2. Tripathi S, Srinivas VV, Nanjundiah RS (2006) Downscaling of precipitation for climatechange scenarios: a support vector machine approach. *J hydrol* 330: 621–640. https://doi.org/10.1016/j.jhydrol.2006.04.030
3. Nash JE, Sutcliffe JV (1970) River flow forecasting through conceptual models part I—A discussion of principles. *J hydrol* 10: 282–290. https://doi.org/10.1016/0022-1694(70)90255-6
4. Zhang G, Bateni SM, Jun C, Khoshkam H, Band SS, Mosavi A (2022) Feasibility of random forest and multivariate adaptive regression splines for predicting long-term mean monthly dew point temperature. *Front Environ Sci* 10: 826165. https://doi.org/10.3389/fenvs.2022.826165
5. Elbeltagi A, Kushwaha NL, Rajput J, et al. (2022) Modelling daily reference evapotranspiration based on stacking hybridization of ANN with meta-heuristic algorithms under diverse agro-climatic conditions. *Stoch Environ Res Risk Assess* 36: 3311–3334. https://doi.org/10.1007/s00477-022-02196-0
6. Dimitriadou S, Nikolakopoulos KG (2022) Artificial neural networks for the prediction of the reference evapotranspiration of the Peloponnese Peninsula, Greece. *Water* 14: 2027. https://doi.org/10.3390/w14132027

7. Malik A, Jamei M, Ali M, et al. (2022) Multi-step daily forecasting of reference evapotranspiration for different climates of India: A modern multivariate complementary technique reinforced with ridge regression feature selection. *Agr Water Manage* 272: 107812. https://doi.org/10.1016/j.agwat.2022.107812
8. Aghelpour P, Norooz-Valashedi R (2022) Predicting daily reference evapotranspiration rates in a humid region, comparison of seven various data-based predictor models. *Stoch Environ Res Risk A* 36: 4133–4155. https://doi.org/10.1007/s00477-022-02249-4
9. Achite M, Jehanzaib M, Sattari MT, et al. (2022) Modern techniques to modeling reference evapotranspiration in a semiarid area based on ANN and GEP models. *Water* 14: 1210. https://doi.org/10.3390/w14081210
10. Sanikhani H, Deo RC, Samui P, et al. (2018) Survey of different data-intelligent modeling strategies for forecasting air temperature using geographic information as model predictors. *Comput Electron Agr* 152: 242–260. https://doi.org/10.1016/j.compag.2018.07.008
11. Bellido-Jiménez JA, Gualda JE, García-Marín AP, (2021) Assessing new intra-daily temperature-based machine learning models to outperform solar radiation predictions in different conditions. *Appl Energ* 298: 117211. https://doi.org/10.1016/j.apenergy.2021.117211
12. Yakut E, Süzülmüş S (2020) Modelling monthly mean air temperature using artificial neural network, adaptive neuro-fuzzy inference system and support vector regression methods: A case of study for Turkey. *Network Comp Neural* 31: 1–36. https://doi.org/10.1080/0954898X.2020.1759833
13. Citakoglu H, Babayigit B, Haktanir NA, (2020) Solar radiation prediction using multi-gene genetic programming approach. *Theor Appl Climatol* 142: 885–897. DOI:10.1007/s00704-020-03356-4
14. Biazar SM, Rahmani V, Isazadeh M, et al. (2020) New input selection procedure for machine learning methods in estimating daily global solar radiation. *Arab J Geosci* 13: 1–17. https://doi.org/10.1007/s12517-020-05437-0
15. Raza A, Shoaib M, Faiz MA, et al. (2020). Comparative assessment of reference evapotranspiration estimation using conventional method and machine learning algorithms in four climatic regions. *Pure Appl Geophys* 177: 4479–4508. https://doi.org/10.1007/s00024-020-02473-5
16. Lu X, Fan J, Wu L, et al. (2020) Forecasting multi-step ahead monthly reference evapotranspiration using hybrid extreme gradient boosting with grey wolf optimization algorithm. *Comp Model Eng Sci* 125: 699–723. https://doi.org/10.32604/cmes.2020.011004
17. Maroufpoor S, Sanikhani H, Kisi O, et al. (2019) Long-term modelling of wind speeds using six different heuristic artificial intelligence approaches. *Int J Climatol* 39: 3543–3557. https://doi.org/10.1002/joc.6037
18. Jahani B, Mohammadi B, (2019) A comparison between the application of empirical and ANN methods for estimation of daily global solar radiation in Iran. *Theor Appl Climatol* 137: 1257–1269. https://doi.org/10.1007/s00704-018-2666-3

19. Kiafar H, Babazadeh H, Marti P, et al. (2017) Evaluating the generalizability of GEP models for estimating reference evapotranspiration in distant humid and arid locations. *Theor Appl Climatol* 130: 377–389. https://doi.org/10.1007/s00704-016-1888-5
20. Mehdizadeh S, Behmanesh J, Khalili K (2017) Using MARS, SVM, GEP and empirical equations for estimation of monthly mean reference evapotranspiration. *Comput Electron Agr* 139: 103–114. https://doi.org/10.1016/j.compag.2017.05.002
21. Tezel G, Buyukyildiz M (2016). Monthly evaporation forecasting using artificial neural networks and support vector machines. *Theor Appl Climatol* 124: 69–80. https://doi.org/10.1007/s00704-015-1392-3
22. Sharifi SS, Rezaverdinejad V, Nourani V (2016) Estimation of daily global solar radiation using wavelet regression, ANN, GEP and empirical models: A comparative study of selected temperature-based approaches. *J Atmos Sol-Terr Phys* 149: 131–145. https://doi.org/10.1016/j.jastp.2016.10.008
23. Martí P, González-Altozano P, López-Urrea R, et al. (2015) Modeling reference evapotranspiration with calculated targets. Assessment and implications. *Agr Water Manage* 149: 81–90. https://doi.org/10.1016/j.agwat.2014.10.028
24. Kisi O, Sanikhani H, Zounemat-Kermani M, et al. (2015) Long-term monthly evapotranspiration modeling by several data-driven methods without climatic data. *Comput Electron Agr* 115: 66–77. https://doi.org/10.1016/j.compag.2015.04.015
25. Shiri J, Marti P, Kisi O, et al. (2013) Evaluation of different data management scenarios for estimating daily reference evapotranspiration. *Hydrol Res* 44: 1058–1070. https://doi.org/10.2166/nh.2013.154
26. Traore S, Guven A, (2012) Regional-specific numerical models of evapotranspiration using gene-expression programming interface in sahel. *Water Resour Manag* 26: 4367–4380. https://doi.org/10.1007/s11269-012-0149-3
27. Izadifar Z, Elshorbagy A, (2010) Prediction of hourly actual evapotranspiration using neural networks, genetic programming, and statistical models. *Hydrol Process* 24: 3413–3425. https://doi.org/10.1002/hyp.7771
28. Kisi O, Guven A, (2010) Evapotranspiration modeling using linear genetic programming technique. *J Irrig Drain Eng* 136: 715–723. https://doi.org/10.1061/(ASCE)IR.1943-4774.0000244
29. Kim S, Kim HS, (2008) Neural networks and genetic algorithm approach for nonlinear evaporation and evapotranspiration modeling. *J Hydrol* 351. https://doi.org/10.1016/j.jhydrol.2007.12.014
30. Fuladipanah M, Azamathulla HM, Kisi O, et al. (2024) Quantitative forecasting of bed sediment load in river engineering: an investigation into machine learning methodologies for complex phenomena. *Water Supply* 24: 585-600. https://doi.org/10.2166/ws.2024.017

31. Sajindra H, Abekoon T, Wimalasiri EM, et al. (2023) An Artificial Neural Network for Predicting Groundnut Yield Using Climatic Data. *Agri Engineering* 5 1713-1736. https://doi.org/10.3390/agriengineering5040106
32. Jayathilake T, Sarukkalige R, Hoshino Y, et al. (2022) Wetland water level prediction using artificial neural networks-A case study in the Colombo flood detention area, Sri Lanka. *Climate* 11. https://doi.org/10.3390/cli11010001
33. Azamathulla HM, Rathnayake U, Shatnawi A (2018) Gene expression programming and artificial neural network to estimate atmospheric temperature in Tabuk, Saudi Arabia. *Appl Water Sci* 8: 1-7. https://doi.org/10.1007/s13201-018-0831-6
34. Leon LP, Azamathulla HM, Felix P, et al. (2023). Prediction of stiffness modulus of bituminous mixtures using the applications of multi-expression programming and gene expression programming. *Road Mater Pavement Des* 24: 2192-2208. https://doi.org/10.1080/14680629.2022.2126383
35. Fuladipanah M, Hazi MA, Kisi O, (2023) An in-depth comparative analysis of data-driven and classic regression models for scour depth prediction around cylindrical bridge piers. *Appl Water Sci* 13: 231. https://doi.org/10.1007/s13201-023-02022-0
36. Gharehbaghi A, Ghasemlounia R, Afaridegan E, et al. (2023) A comparison of artificial intelligence approaches in predicting discharge coefficient of streamlined weirs. *J Hydroinform* 25: 1513-1530. https://doi.org/10.2166/hydro.2023.063
37. Azamathulla HM (2012). Gene-expression programming to predict scour at a bridge abutment. *J Hydroinform* 14: 324-331. https://doi.org/10.2166/hydro.2011.135
38. Cortes C., Vapnik V. (1995). Support-vector networks. *Mach learn* 20: 273-297. https://doi.org/10.1007/BF00994018
39. Kumar V, Azamathulla HM, Sharma KV, et al. (2023) The state of the art in deep learning applications, challenges, and future prospects: A comprehensive review of flood forecasting and management. *Sustainability* 15: 10543. https://doi.org/10.3390/su151310543
40. Rathnayake N, Rathnayake U, Dang TL, et al. (2023) Water level prediction using soft computing techniques: A case study in the Malwathu Oya, Sri Lanka. *Plos one* 18: e0282847. https://doi.org/10.1371/journal.pone.0282847
41. Noori R, Khakpour A, Omidvar B, et al. (2010) Comparison of ANN and principal component analysis-multivariate linear regression models for predicting the river flow based on developed discrepancy ratio statistics. *Expert Syst Appl* 37: 5856-5862. https://doi.org/10.1016/j.eswa.2010.02.020
42. Koncar N, (1997). Optimization methodologies for direct inverse neurocontrol. (Publication No., SW72BZ) [Doctoral dissertation, London University, Englan].

43. Malik A, Tikhamarine Y, Al-Ansari N, et al. (2021) Daily pan-evaporation estimation in different agroclimatic zones using novel hybrid support vector regression optimized by Salp swarm algorithm in conjunction with gamma test. *Eng Appl Comp Fluid Mech* 15: 1075-1094. https://doi.org/10.1080/19942060.2021.1942990