

Article

Covariance Matrix Adaptation Evolution Strategy for Improving Machine Learning Approaches in Streamflow Prediction

Rana Muhammad Adnan Ikram ¹, Leonardo Goliatt ^{2,*} , Ozgur Kisi ^{3,*} , Slavisa Trajkovic ⁴ 
and Shamsuddin Shahid ⁵ 

¹ School of Economics and Statistics, Guangzhou University, Guangzhou 510006, China

² Computational Modeling, Federal University of Juiz de Fora, Juiz de Fora 36036-900, Brazil

³ Civil Engineering Department, Ilia State University, 0162 Tbilisi, Georgia

⁴ Faculty of Civil Engineering and Architecture, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia

⁵ School of Civil Engineering, Faculty of Engineering, University Teknologi Malaysia (UTM), Johor Bahru 81310, Malaysia

* Correspondence: leonardo.goliatt@uff.br (L.G.); ozgur.kisi@iliauni.edu.ge (O.K.)

Abstract: Precise streamflow estimation plays a key role in optimal water resource use, reservoirs operations, and designing and planning future hydropower projects. Machine learning models were successfully utilized to estimate streamflow in recent years. In this study, a new approach, covariance matrix adaptation evolution strategy (CMAES), was utilized to improve the accuracy of seven machine learning models, namely extreme learning machine (ELM), elastic net (EN), Gaussian processes regression (GPR), support vector regression (SVR), least square SVR (LSSVR), extreme gradient boosting (XGB), and radial basis function neural network (RBFNN), in predicting streamflow. The CMAES was used for proper tuning of control parameters of these selected machine learning models. Seven input combinations were decided to estimate streamflow based on previous lagged temperature and streamflow data values. For numerical prediction accuracy comparison of these machine learning models, six statistical indexes are used, i.e., relative root mean squared error (RRMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), Nash–Sutcliffe efficiency (NSE), and the Kling–Gupta efficiency agreement index (KGE). In contrast, this study uses scatter plots, radar charts, and Taylor diagrams for graphically predicted accuracy comparison. Results show that SVR provided more accurate results than the other methods, especially for the temperature input cases. In contrast, in some streamflow input cases, the LSSVR and GPR were better than the SVR. The SVR tuned by CMAES with temperature and streamflow inputs produced the least RRMSE (0.266), MAE (263.44), and MAPE (12.44) in streamflow estimation. The EN method was found to be the worst model in streamflow prediction. Uncertainty analysis also endorsed the superiority of the SVR over other machine learning methods by having low uncertainty values. Overall, the SVR model based on either temperature or streamflow as inputs, tuned by CMAES, is highly recommended for streamflow estimation.

Keywords: streamflow prediction; extreme learning machine; support vector regression; elastic net; covariance matrix adaptation evolution strategy

MSC: 68T20



Citation: Ikram, R.M.A.; Goliatt, L.; Kisi, O.; Trajkovic, S.; Shahid, S. Covariance Matrix Adaptation Evolution Strategy for Improving Machine Learning Approaches in Streamflow Prediction. *Mathematics* **2022**, *10*, 2971. <https://doi.org/10.3390/math10162971>

Academic Editors: Chaman Verma, Maria Simona Raboaca and Zoltán Illés

Received: 30 July 2022

Accepted: 15 August 2022

Published: 17 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Reliable streamflow prediction is vital for many hydrological and environmental studies, such as flood risk assessment, hydroelectric power, water allocations, irrigation scheduling, and water protection [1,2]. Many factors, such as temperature, precipitation, evapotranspiration, soil types, and urbanization affect the reliability of prediction [3–5]. In recent years, due to climate change in the whole world, water resources faced much change,

which resulted in extreme events such as frequent floods and drought. These extreme events not only cause the loss of many human lives but also create economic instability [6,7]. Therefore, precise estimation of streamflow is necessary to avoid and mitigate this loss by extreme events. The basic characteristic of the whole process is a nonlinear connection between these factors and streamflow.

There are two groups of streamflow prediction techniques: (1) physically based models and (2) data-driven techniques. Numerous previous studies concluded that nonlinear data-driven approaches often outperform physically based models and linear data-driven techniques [8–11]. Among the data-driven techniques, a series of existing studies indicated the potential of machine learning methods (MLMs) in streamflow prediction [12–14]. Kagoda et al. [15] successfully applied RBFNN to predict monthly streamflow in South Africa. Sun et al. [16] introduced the GPR model to predict monthly streamflow using the US MOPEX database. GPR produced more accurate predictions than artificial neural networks (ANN) and linear regression (LR). Mehr et al. [17] compared three ANN techniques. They concluded that RBFNN yielded the best results in monthly streamflow prediction in the Coruh Basin (Turkey). Kisi [18] reported that the LSSVR models outperformed the adaptive neuro-fuzzy embedded fuzzy c-means clustering (ANFIS-FCM) and autoregressive moving average (ARMA) models in monthly streamflow prediction in the Dicle Watershed (Turkey). Yaseen et al. [19] found that LSSVR yielded superior results to the M5 Tree and MARS models in predicting streamflow from observed precipitation.

Modaresi et al. [20] found that LSSVR performed better than the ANN model for monthly streamflow prediction nonlinear conditions in Iran. Worland et al. [21] compared EN, support vector machines (SVM), and the gradient boosting machine (GBM) with the null physically based model for weekly streamflow prediction. The results show that MLMs made more reliable predictions than physically based models. Hadi et al. [22] employed a hybrid XGB-ELM model. XGB was applied as an input selection technique, and ELM as a predicting technique. Climatic data predicted monthly streamflow, including temperature, precipitation, and evapotranspiration from the Goksu-Himmeti basin (Turkey). Overall results prove that the XGB-ELM model improved the streamflow prediction. The authors concluded that reliable streamflow prediction is possible using only temperature data; however, other meteorological variables enhance the model's capability. Li et al. [23] investigated EN, XGB, and SVR in monthly streamflow prediction using data from China. XGB outperformed EN, and SVR approaches. The authors concluded that MLMs could be used for monthly prediction.

Ni et al. [24] successfully predicted monthly streamflows by using a new hybrid technique based on XGB coupled with the Gaussian mixture model. The new technique performed better compared to SVM. Adnan et al. [25] applied multivariate adaptive regression splines (MARS), least square SVM (LSSVM), M5 model tree (M5 Tree), optimally pruned extreme learning machines (OP-ELM), and the seasonal autoregressive moving average (SARIMA) for predicting river flow using the temperature and precipitation data of two mountainous stations of Pakistan. The LSSVM and MARS models yielded the best results in Kalam station and Chakdara station, respectively. Only temperature data from Kalam station provided a successful streamflow prediction. LSSVM, OP-ELM, and MARS performed better than the SARIMA model. Thapa et al. [26] used SVR and GPR models for predicting snowmelt streamflow in the Langtang basin (Nepal). Jiang et al. [27] implemented the new ELM model with enhanced particle swarm optimization (ELM-IPSO) for simulating discharge from the Chaohe River basin (China). The ELM-IPSO made more reliable monthly streamflow predictions than ANN, the auto-regression method (AR), ELM with PSO algorithm (ELM-PSO), and ELM with genetic algorithm (ELM-GA). Malik et al. [28] applied SVR models to predict daily river flow in the Naula basin (India). They employed six meta-heuristic algorithms for optimizing SVR models. Authors reported that Harris hawks optimization (HHO) provided the best performance to the hybrid SVR-HHO model. Parisouj et al. [29] investigated ANN, ELM, and SVR in daily and monthly streamflow prediction using temperature and precipitation input data from four basins in the western

USA. The results show that SVR made more reliable daily and monthly predictions than ELM and ANN models. Niu and Feng [30] applied ANFIS, ANN, SVM, ELM, and GPR for daily streamflow prediction using data from south China. Obtained results show that ELM, GPR, and SVM models outperformed ANN and ANFIS models.

In recent years, hybrid machine learning models (MLMs) became preferred over standalone models and are successfully applied in the difference fields to model different variables; e.g., for load forecasting [31], to estimate the international airport freight volumes [32], to predict electricity prices [33], and for modeling of the tensile strength of the concrete [34]. Due to the nonlinear nature of hydrological variables' time series, researchers found more precise and accurate results by utilizing hybrid MLMs than standalone MLMs. Numerous meta-heuristic algorithms, such as Harris Hawks optimization (HHO), Particle Swarm Optimization (PSO), Ant Lion Optimization (ALO), Grey Wolf Optimizer (GWO), and the Fruit Fly Algorithm (FFA) were integrated with MLMs for streamflow prediction in recent studies [28,35,36]. However, in addition to the precision of meta-heuristics algorithms, the robustness and simplicity of meta-heuristic algorithms are also considered. Therefore, in this study, instead of complex metaheuristic algorithms, a simple robust meta-heuristic algorithm, the covariance matrix adaptation evolution strategy (CMAES), is introduced to build hybrid machine learning methods [37]. CMAES was successfully applied to solve many engineering problems [38–41]. However, CMAES was so far never used to predict streamflow. That fact was the key motivation for this study. This study demonstrates that CMAES is an alternative to perform the selection of different machine learning models in flow prediction. In this research, the hybrid explores the CMAES algorithm's search potential to generate benefits for machine learning models through the suitable selection of internal parameters. Furthermore, this study also performs an uncertainty analysis of the developed models, contributing to ensuring the robustness of the generated models and assisting in faster decision making. The computational framework is flexible and has the potential to embed different machine learning models that were not addressed in this study. The proposed strategy can be used by environmental monitoring and water management agencies, delivered as a web service and serving to issue short-term alerts, mitigating the effect of flash floods and assisting in managing water resources along the river studied in this research.

The objectives of this study are: (1) to predict monthly streamflow using antecedent streamflow observations and/or temperatures by different hybrid machine learning methods (MLMs), such as elastic net (EN), extreme learning machine (ELM), least squares support vector regression (LSSVR), support vector regression (SVR), radial basis function neural network (RBFNN), extreme gradient boosting (XGB) and Gaussian processes regression (GPR) (2) to implement CMAES as an optimization algorithm for these MLMs and (3) to evaluate the results of the applied hybrid machine learning methods.

2. Materials and Methods

2.1. Study Region and Datasets

The Upper Indus basin is selected as a case study in this study. The Upper Indus basin (UIB) is part of the world's largest transboundary river basin, i.e., the Indus Basin, with a catchment area of 289,000 km², almost covering 56% of the main Indus basin. UIB is located in the western area of Pakistan with a catchment elevation variation of 200 m to 8500 m. UIB is chosen due to the backbone role of this basin in the country's agriculture sector. UIB fulfills the 90% irrigation requirements of the agriculture sector of Pakistan. However, the agriculture sector was affected in recent years due to insufficient water supplies. Therefore, precise measurement in this key basin is important for proper water supply management. For streamflow estimation, the Bunji sub-basin in this key basin is selected in this study (Figure 1). Monthly streamflow data of the Bunji hydraulic station (latitude: 35.4, longitude: 74.4) and temperature data of the Bunji meteorological station from 1977 to 2008 is obtained from WAPDA. Thirty-two years of monthly data are partitioned into calibration and testing parts for streamflow analysis with training

and testing proportions of 75% and 25% training and testing proportions, respectively. Table 1 shows the statistical parameters of the streamflow and temperature data. Bunji basin is located at the confluence of the three biggest mountain ranges in the world, i.e., Hindu Kush, the Karakoram, and the Himalaya (HKH) mountains. This basin is selected as it directly drains to UIB. A key hydropower project is planned in this basin, i.e., Bunji Hydropower with 7100 MW generation capacity. Therefore, precise discharge simulation in this key basin can help the water manager optimize the scheduling of planned hydropower projects. Figure 2 shows the training (blue) and test set (orange) used in this study.

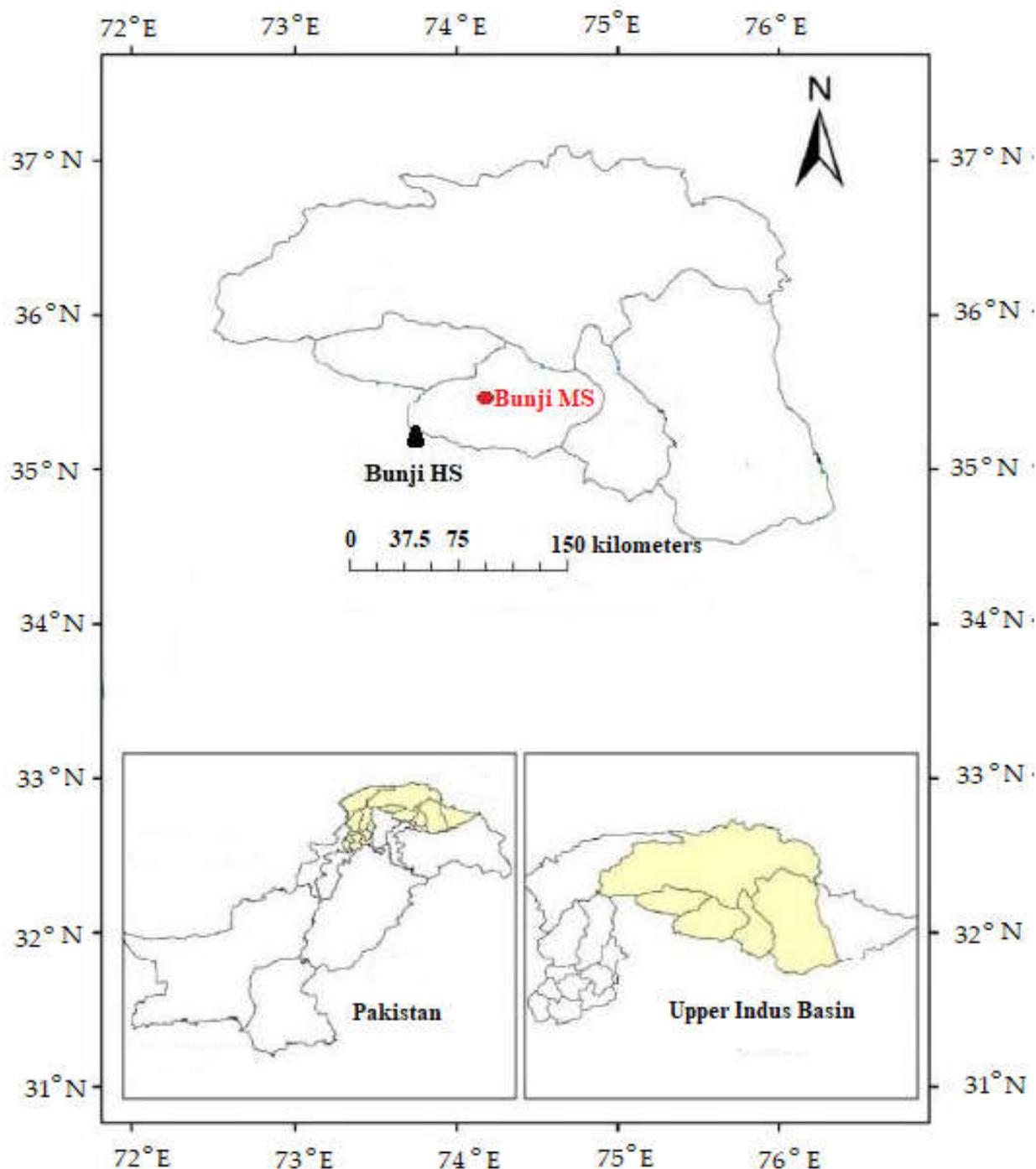


Figure 1. Location map of selected stations.

Table 1. The statistical parameters of the streamflow and temperature data.

Dataset	Parameter	Min	Mean	Std	Max
Training	T (C)	−3.33	13.80	8.72	29.45
	Q (m/s)	251.72	1784.68	1971.37	7799.4
Test	T (C)	−1.22	14.15	9.03	28.65
	Q (m/s)	272.50	1844.64	1941.29	1941.29

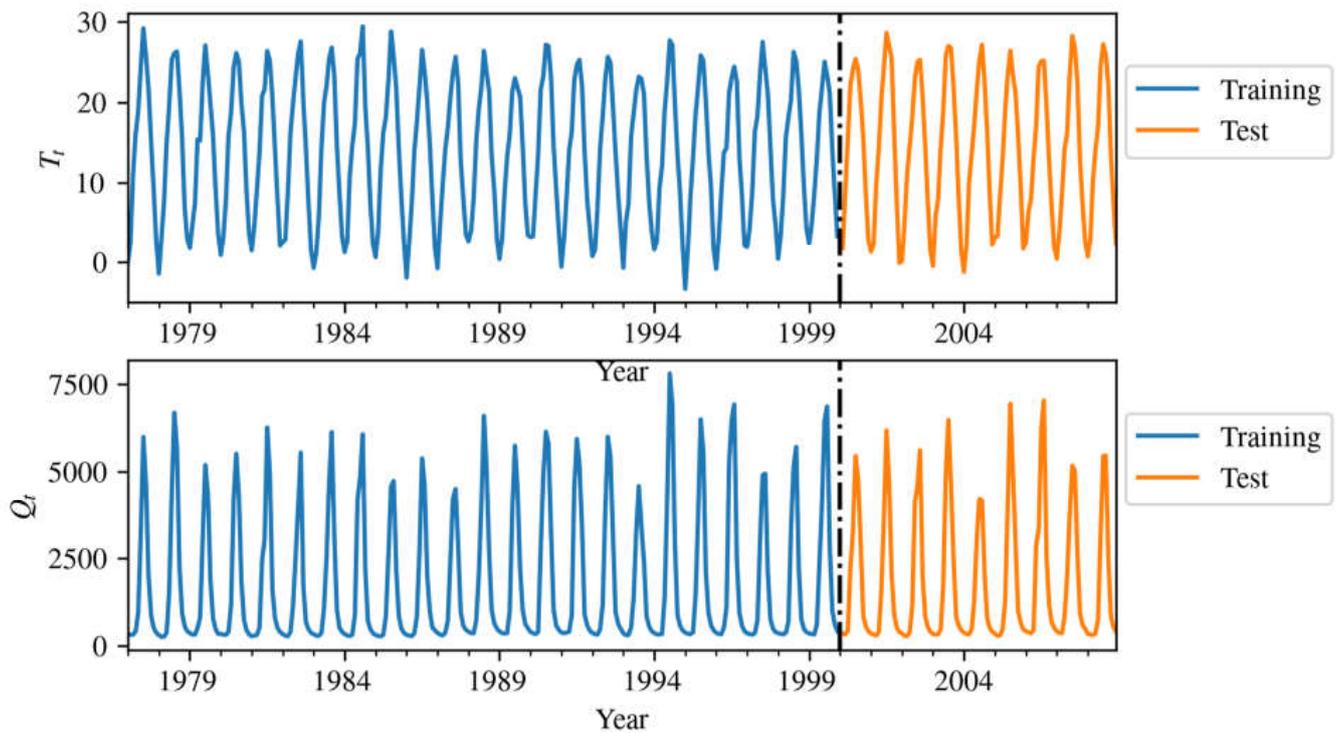


Figure 2. Training set and test set.

2.2. Streamflow Estimation Models

The prediction model considers the present streamflow as a function of antecedent streamflow and/or temperatures. The predictive model has the following form:

$$Q_t = F\left(T_{t-1}, \dots, T_{t-k} \overset{\text{temperatures}}{\underbrace{\quad}}, Q_{t-1}, \dots, Q_{t-k} \overset{\text{streamflows}}{\underbrace{\quad}}\right) \tag{1}$$

where Q_{t-k} represent streamflow at month $t - k$, T_{t-k} is the temperature measured at month $t - k$, and $F(\cdot)$ is an estimation function. Table 2 shows the input combinations to the model presented in Equation (1). Each input combination represents a case shown in the first column. The databases in the input and output nodes were normalized between -1 and 1 as below:

$$X_{tN} = \frac{2(X_i - X_{min})}{X_{max} - X_{min}} - 1 \tag{2}$$

where X_{tN} = normalized data of Q and T , whereas $i = t, t - 1, t - 2, t - 3$, and $t - 4$.

Table 2. Input features associated with models tested in this paper.

Case.	Input Features
Case 1	T_{t-1}
Case 2	T_{t-1}, T_t
Case 3	T_{t-11}, T_{t-1}, T_t
Case 4	$T_{t-12}, T_{t-11}, T_{t-1}, T_t$
Case 5	Q_{t-1}
Case 6	Q_{t-11}, Q_{t-1}
Case 7	$Q_{t-12}, Q_{t-11}, Q_{t-1}$

2.2.1. Elastic Net (EN)

The EN uses a generalized linear regression [42] to best fit the coefficients $w = [w_0, w_1, w_2, \dots, w_p]$ by minimizing the residual of the fitted model. The optimization problem of EN can be stated as,

$$\min_w \frac{1}{2N} \| Xw - y \|^2 + \alpha \rho \| w \|_1 + \frac{\alpha(1 - \rho)}{2} \| w \|^2 \tag{3}$$

where N represents sample size, X is the dataset, y is the outputs, $\alpha \geq 0$, $\| w \|_2$ and $\| w \|_1$ represent the L_2 - and L_1 -norm of the parameter vector, respectively, and ρ represents the L_1 -ratio parameter. EN model is trained with L_1 and L_2 for learning a sparse model with few non-zero weights while maintaining regularization properties. This trade-off is controlled using the parameter ρ , and the so-called L_1 -ratio parameter. EN is suitable for developing a model with multiple inter-correlated inputs [43].

2.2.2. Extreme Learning Machine (ELM)

ELM is a kind of a feedforward ANN consisting of a single hidden layer. It can learn fast and is generalized efficiently, and therefore, suitability for modeling [44]. ELMs have three stages of randomness: (1) completely connected hidden node parameters are created randomly; (2) inputs to hidden node connections are produced randomly; and (3) a hidden node acts as a subnetwork consisting of several nodes. ELM can be mathematically presented as [45],

$$\hat{y}(x) = \sum_{i=1}^L \beta_i G(w_i, b_i, x) \tag{4}$$

where \hat{y} represents the output for the input x , w_i denotes the i th hidden node’s weight, b_i denotes the hidden node’s biases of the neurons in the hidden layer, β_i represents the output weight, $G(\cdot)$ represents the activation function, and L denotes the hidden node number. The parameters (w, b) are random numbers, $N(0, 1)$ having a normal distribution, while β_i is decided logically. Table 3 provides the $G(w, b, x)$ for (w, b) .

Table 3. Activation functions used in ELM.

#	Name	Activation Function G
1	Identity	$G(w, b, x) = (w \cdot x + b)$
2	Sigmoid	$G(w, b, x) = \frac{1}{1 + \exp(-w \cdot x + b)}$
3	Hyperbolic tangent	$G(w, b, x) = \frac{1 - \exp(w \cdot x + b)}{1 + \exp(w \cdot x + b)}$
4	Gaussian	$G(w, b, x) = \exp$
5	Swish	$G(w, b, x) = \frac{(w \cdot x + b)}{1 + \exp(w \cdot x + b)}$
6	ReLU	$G(w, b, x) = \max_i(0, (w \cdot x + b))$

The values of $[\beta_1, \dots, \beta_L]$ are estimated by minimizing [46],

$$\min_{\beta \in R} (\|H\beta - y\| + C\|\beta\|^2) \tag{5}$$

where H represents the hidden layer output matrix. The optimal weight (β) can be estimated as,

$$\beta = (H^T H)^{-1} H^T y = H^\dagger y \tag{6}$$

where H^\dagger is the pseudoinverse of H .

2.2.3. Least Squares Support Vector Regression (LSSVR)

The LSSVR method [47] assumes that a nonlinear function (ϕ) represents a dataset $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ as,

$$f(x) = \omega^T \phi(x) + b \tag{7}$$

where $x \in R^n$, $y \in R$, ω represents weight, and b denotes bias. The unknown parameters can be estimated by minimizing a residual function $R(\omega, e, b)$

$$\min R(\omega, \epsilon, b) = \frac{1}{2} \|\omega\|^2 + \frac{1}{2} C \sum_{i=1}^n \epsilon_i^2 \tag{8}$$

where C is the regularization constant, and ϵ_i represents error. The objective function $R(\omega, e, b)$ consists of a loss and a regularization term, such as a feedforward ANN. However, $R(\omega, e, b)$ is hard to solve for larger ω . The Lagrange multiplier optimal programming method is generally employed for its solution,

$$L(\omega, b, \epsilon, \zeta) = R(\omega, \epsilon, b) - \sum_{i=1}^n \zeta_i (\omega^T \phi(x_i) + b + \epsilon_i - y_i) \tag{9}$$

where ζ_k is a Lagrange multiplier. It is solved by optimizing the following criteria:

$$\begin{aligned} \frac{\partial L}{\partial \omega} = 0 \rightarrow \omega = \sum_{i=1}^n \zeta_i \phi(x_i) = 0 \quad \frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^n \zeta_i = 0 \quad \frac{\partial L}{\partial \epsilon_i} = 0 \rightarrow \zeta_i = C\epsilon_k, i = 1, \dots, n \quad \frac{\partial L}{\partial \zeta_i} = 0 \rightarrow \omega \phi(x_k) + b + \epsilon_i - y_i, i = 1, \dots, n \end{aligned} \tag{10}$$

Eliminating ϵ and ω , the optimization problem can be presented as below:

$$\begin{bmatrix} 0 & 1_v^T & 1_v & \omega + \frac{1}{C} \end{bmatrix} \begin{bmatrix} b & \zeta \end{bmatrix} = \begin{bmatrix} 0 & y \end{bmatrix} \tag{11}$$

where $y = [y_1, \dots, y_n]$, $1_v = [1, \dots, 1]$, and $\zeta = [\zeta_1, \dots, \zeta]$, as well as the kernel function are defined as

$$\omega = \phi(x_i)^T \phi(x_k) = K(x_i, x_k) \tag{12}$$

where

$$K(x_i, x_k) = \exp \frac{\|x_i - x_k\|^2}{2\gamma} \tag{13}$$

is the radial basis function (RBF) kernel. The LSSVR model output can be written as,

$$y(x) = \sum_{i=1}^n \zeta_i K(x_i, x_k) + b \tag{14}$$

where ζ_k and b are the solutions to the linear system, and γ represents the kernel function parameter. The tuning parameters (C, γ) are generally defined by the user.

2.2.4. Support Vector Regression (SVR)

SVR is a classical algorithm [48] applied to solve regression problems [49]. It can be presented as,

$$f(x) = \sum_{j=1}^N w_j K(x, x_j) + b \tag{15}$$

where $K(\cdot, \cdot)$ represents kernel function, $w = [w_1, \dots, w_N]$ represent weights, b and N denote bias and sample size, respectively. RBF is used in this study, which can be presented as,

$$K(x, x_i) = \sum_{i=1}^N \exp(-\gamma \|x - x_i\|^2) \tag{16}$$

where γ is the bandwidth parameter; w and b is determined by minimizing

$$J = \sum_{i=1}^N w_i^2 + \frac{C}{N} \sum_{i=1}^N L_\epsilon(y_i - f(x_i)), L_\epsilon(y - f(x)) = \begin{cases} 0 & \text{if } |y - f(x)| \leq \epsilon \\ |y - f(x)| - \epsilon & \text{otherwise} \end{cases} \tag{17}$$

where y_i presents the output for inputs x_i , L_ϵ denotes the ϵ -insensitive loss function, and C denotes the regularization parameter [50]. In the SVR model presented in this paper, the parameters C and ϵ are to be adjusted to improve the SVR ability to model the regression problem [51].

2.2.5. Radial Basis Function Neural Network (RBFNN)

RBFNN [52] is a three-layer ANN commonly used for function interpolation problems. RBFNN is different from other ANN types in its universal approximation and rapid learning ability. The first and the last layers of RBFNN are used for attaining inputs and providing output, respectively. The hidden layer is composed of several radial basis nonlinear activation units. Compared to multilayer neural networks, RBFNN presents some differences: (1) the RBF network has only one hidden layer, (2) the activation function is according to distances instead of an inner product in multilayer networks; and (3) the approximation for an RBFNN network is a local one, while multilayer networks realize global approximation for the entire space.

For an array of points $\{x_i; i = 1, \dots, N\}$, $x_i \in R^n$ and their respective functional values, $y_i = f(x_i)$, $i = 1, \dots, N$, where $f(\cdot)$ is a functional input/output relationship, the RBFNN constructs an approximation to $f(\cdot)$ in the form

$$\hat{y}(x) = \sum_{i=1}^N w_i \phi(x_i, \epsilon) \tag{18}$$

where

$$\phi(x_i, \epsilon) = \phi(\|x - x_i\|, \epsilon), \epsilon > 0 \tag{19}$$

is the i th hidden layer's activation function, x_i id denotes the i th data point, and ϵ is the width of the activation function. Inserting the condition displayed in the previous equations gives the following set of linear equations $f = Aw$ where, $A = [a_{ij}]$, $f = [y_i]$, and $w = [w_j]$. The training procedure for the RBFNN includes calculating the widths and weights. The adjustment of weights is performed using least squares, adding a smoothing parameter

$$f = (A + \lambda I)w \Rightarrow w = (A + \lambda I)^{-1}f \tag{20}$$

where λ is the smooth parameter. The smoothness of the approximation is increased when $\lambda > 0$ and $\lambda = 0$ is for interpolation. As a result, the function will always go through the nodal points in this case. The activation function in the RBFNN formulation is crucial to its performance. Table 4 shows a set of radial basis activation functions used in this study. The smooth parameter ϵ is only used in multiquadric, inverse multiquadric, and Gaussian activation functions.

Table 4. Activation functions used in RBFNN.

#	Name	Activation Function ϕ
1	Linear	$\phi(r, \epsilon) = r$
2	Multiquadrics	$\phi(r, \epsilon) = \sqrt{\left(\frac{r}{\epsilon}\right)^2 + 1}$
3	Inverse multiquadrics	$\phi(r, \epsilon) = \frac{1}{\sqrt{\left(\frac{r}{\epsilon}\right)^2 + 1}}$
4	Gaussian	$\phi(r, \epsilon) = \exp(-r^2)$
5	Cubic	$\phi(r, \epsilon) = r^3$
6	Quintic	$\phi(r, \epsilon) = r^5$
7	Thin plate	$\phi(r, \epsilon) = r^2 \log(r)$

2.2.6. eXtreme Gradient Boosting (XGB)

XGB [53] is an ensemble algorithm that combines several weak learners to predict better. In XGB, weak learners are regularized decision trees [54]. The XGB can mathematically be presented as,

$$y = F_m(x_i) = \sum_{m=1}^{M_{est}} h_m(x_i) \tag{21}$$

where M_{est} is the number of estimators, h_m represents weak learners (decision trees) with depth defined by parameter m_{depth} . XGB is developed greedily,

$$F_m(x) = F_{m-1}(x) + \eta h_m(x) \tag{22}$$

where η is a learning rate and the newly added tree $h_m(x)$ is fitted to minimize a sum of losses L_m and is given by

$$h_m = \underset{h}{\operatorname{argmin}} L_m = \underset{h}{\operatorname{argmin}} \sum_{i=1}^N l(y_i, F_{m-1}(x_i) + h(x_i)) + \Omega(F_{m-1}) \tag{23}$$

In the XGB formulation, the loss function is written as

$$l(y_i, F_m(x)) = [y_i - F_m(x_i)]^2 \tag{24}$$

where N is the number of samples and

$$\Omega(F_m) = T + \frac{1}{2} \lambda_{reg} \|w\|^2 \tag{25}$$

is the regularization term, where T represents the number of leaves, w denotes leaf weights, and λ_{reg} is the L_2 regularization term on weights.

2.2.7. Gaussian Processes Regression (GPR)

The GPR predicts output based on the distribution over function instead of considering the function’s parameter distribution [55]. A Gaussian process model is represented as:

$$y = f(x) + \epsilon \tag{26}$$

where $f(\cdot)$ is a nonlinear function that is used to infer output, y from inputs, x , ϵ is Gaussian noise with zero means (i.e., $\epsilon \sim N(0, \sigma_v^2)$). The prior for $f(\cdot)$ is a Gaussian process, and $f(\cdot) \sim G(\mu, C)$ and C represent the covariance matrix.

The predictive distribution of the output parameter \hat{y} , given its input x , is formulated as [56],

$$\hat{y}(x) = k^T(x)(K + \alpha I)^{-1}y \tag{27}$$

where $y = [y_1, y_2, \dots, y_N]^T$ is the vector of training data, $k = [K(x, x_1), K(x, x_2), \dots, K(x, x_N)]^T$, and K represents the function of covariance matrix, and λ refers to regularization parameter. Equation (28) gives the covariance between x_i and x_j [57] as,

$$K_{ij} = K(x_i, x_j) = k_0 \frac{2^{1-\nu}}{\Gamma(\nu)} (\|x - x_j\| \sqrt{2\nu l^{-1}})^\nu K_\nu \left(\|x - x_j\| \sqrt{2\nu l^{-1}} \right) \quad (28)$$

where ν and l are positive parameters, $\Gamma(\cdot)$ and $K_\nu(\cdot)$ are the gamma and the modified Bessel function [58].

2.3. Parameter Tuning Guided by a CMAES

CMAES as an evolutionary algorithm [59] applies iterative optimization by sampling population involving λ solutions obtained from multi-dimensional normal distribution. CMAES is dependent on a means vector \mathbf{m} and a covariance matrix \mathbf{C} using a step size σ [60]. The pseudocode of CMAES is shown in Algorithm 1. Details of CMAES are provided in [61]. In CMAES, each candidate solution x encodes the internal parameters of the machine learning model in the form $x = (\theta_1, \theta_2, \dots, \theta_M)$, where M is the hyperparameter number of the machine learning model. Table 5 shows the internal parameters and their respective description and range.

Algorithm 1. Pseudocode of the CMAES searches the internal parameters of the machine learning models.

- 1: initialize $\mathbf{m}_0 = 0, \sigma_0 = 0.5, \mathbf{C}_0 = \mathbf{I}, \mathbf{p}_0 = 0, \mathbf{s}_0 = 0, k = 0, \mu = \lambda/2$
 - 2: initialize $\{(\theta_i, z_i = OF(\theta_i)), i = 1, \dots, \lambda\}$
 - 3: **while** $k < N_G$ **do**
 - 4: **for** $i = 1$ to λ **do**
 - 5: $\theta_i = \mathbf{m}_k + \sigma_k \mathcal{N}(0, \mathbf{C}_k)$
 - 6: $f_i = \text{RMSE}(\mathbf{x}_i)$
 - 7: **end for**
 - 8: sort solutions $\theta_{1:\lambda} \leftarrow \theta_{s(1):\dots s(\lambda)}$ with $s(i) = \text{argsort}_f(f_{1:\lambda}, i)$
 - 9: set $w_{1:\mu} = \{w_1 \geq w_2 \geq \dots \geq w_\mu > 0 \mid \sum_{i=1}^\mu w_i = 1\}$
 - 10: $\mathbf{m}_{k+1} = \mathbf{m}_k + \sum_{i=1}^\mu w_i (\theta_{1:\lambda} - \mathbf{m}_k)$
 - 11: $\mathbf{s}_{k+1} = (1 - c_\sigma) \mathbf{s}_k + \sqrt{\mu_{eff}} \sqrt{c_\sigma (2 - c_\sigma)} \mathbf{C}_k^{-1/2} \frac{\mathbf{m}_{k+1} - \mathbf{m}_k}{\sigma_k}$
 - 12: $\mathbf{p}_{k+1} = (1 - c_c) \mathbf{p}_k + h_\sigma \sqrt{\mu_{eff}} \sqrt{c_c (2 - c_c)} \frac{\mathbf{m}_{k+1} - \mathbf{m}_k}{\sigma_k}$
 - 13: $\mathbf{C}_{k+1} = (1 - c_1 - c_\mu) \mathbf{C}_k + c_1 \mathbf{p}_{k+1} \mathbf{p}_{k+1}^T + \frac{c_\mu}{\sigma_k^2} \sum_{i=1}^\mu w_i (\theta_{1:\lambda} - \mathbf{m}_k) w_i (\theta_{1:\lambda} - \mathbf{m}_k)^T$
 - 14: $\sigma_{k+1} = \sigma_k \exp \left[\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{s}_{k+1}\|}{\chi_n} - 1 \right) \right]$
 - 15: $k \leftarrow k + 1$
 - 16: **end while**
-

Table 5. Encoding of candidate solutions. The column IP indicates the internal parameter in the CMAES encoding.

Method	IP	Description	Settings/Range
ELM	q ₁	No. neurons in the hidden layer	[1, 500]
	q ₂	Regularization parameter, C	[0.0001, 10,000]
	q ₃	Activation function, G	1: Identity; 2: Sigmoid; 3: Hyperbolic Tangent; 4: Gaussian; 5: Swish; 6: ReLU;
EN	q ₁	Penalty term, a	[10 ⁻⁶ , 1]
	q ₂	L1-ratio parameter, r	[0, 1]
GPR	q ₁	Kernel parameter, k_0	[0.001, 10]
	q ₂	Kernel parameter, n	[0.001, 10]
	q ₃	Kernel parameter, k_0	[0, 100]
	q ₄	Regularization parameter, a	[10 ⁻⁸ , 1]
LSSVR	q ₁	Regularization parameter, C	[1, 1000]
	q ₂	Bandwidth parameter, g	0.001, 100]
RBFNN	q ₁	Activation function width, e	[1, 500]
	q ₂	Smoothing parameter, l	[0.0001, 10,000]
	q ₃	Activation function, f	1: Linear; 2: Multiquadric; 3: Inverse Multiquadric; 4: Gaussian; 5: Cubic; 6: Quintic; 7: Thin Plate;
SVR	q ₁	Loss parameter, e	[10 ⁻⁵ , 1]
	q ₂	Regularization parameter, C	[1, 10,000]
	q ₃	Bandwidth parameter, g	[0.001, 10]
XGB	q ₁	Learning rate, l	[10 ⁻⁶ , 1]
	q ₂	No. weak estimators, M_{est}	[10, 500]
	q ₃	Maximum depth, m_{depth}	[1, 20]
	q ₄	Regularization parameter, l_{reg}	[0, 100]

Root mean squared error (RMSE) between simulated and actual values is used as the objective function in CMAES to evaluate the solutions. For assessing an individual’s fitness, 5-fold was set and cross validation was employed. In this method, data are randomly split into 5 subsets. The machine learning model using the internal parameter vector x is trained using 4 subsets, and the remaining subset is used for RMSE calculation. The process is iterated 5 times and the average RMSE is computed. According to the RMSE, the population is evolved over N_G generations. The objective is the estimate of the optimal parameter x with the least RMSE. At each iteration k , the parameters m_k , C_k , and σ_k are updated considering the evaluated population based on the algorithm (see Algorithm 1). Figure 3 shows the architecture of the proposed approach and the integration of CMAES with machine learning models. The training modeling phase involves fine tuning the internal estimator parameters using the CMAES. The model decoded from the best individual is employed for predictions during testing.

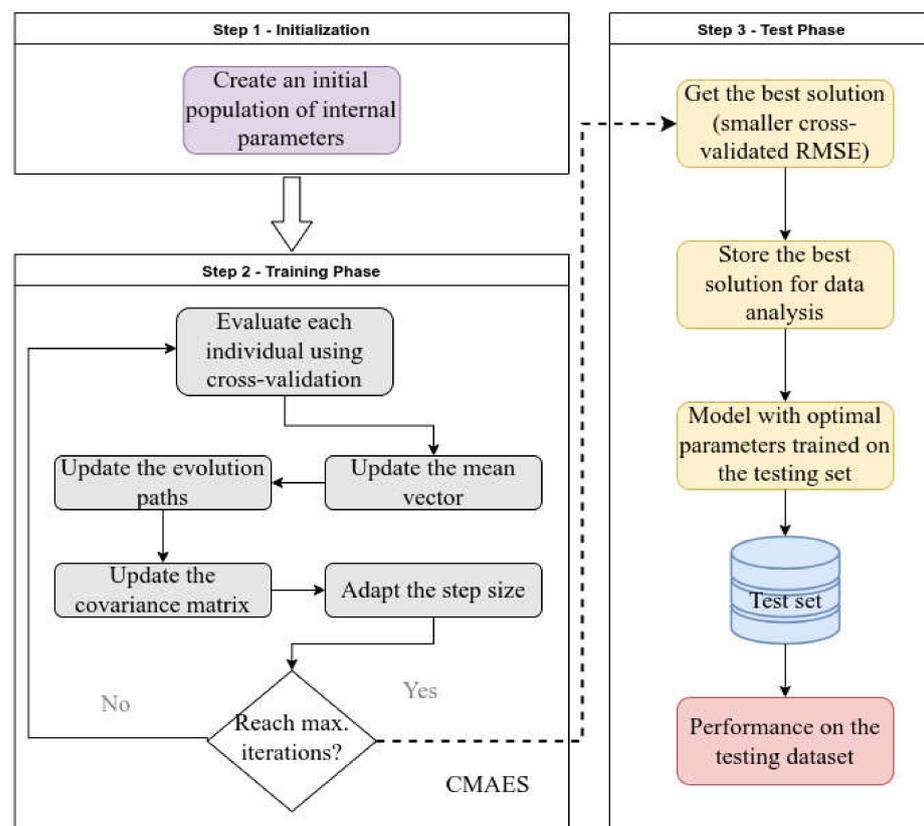


Figure 3. Framework used in this paper showing the combination of the CMAES evolutionary algorithm and the machine learning models.

2.4. Performance Metrics

Six metrics are used in this paper to assess the model’s performance. Table 6 represents the metrics applied and their expressions. In Table 6, MAE, RRMSE, and WI represent the mean absolute error, relative root mean squared error, and agreement index, respectively [62]. MAPE and NSE refer to the mean absolute percentage error and Nash–Sutcliffe efficiency [63]. KGE represents the Kling–Gupta efficiency between simulated and observed values [64]. The metrics used in this study were commonly and successfully used in recent years to model streamflow and hydrological variables [65–67]. The purpose of using multiple metrics was to show that the optimized models present good performance in many aspects. KGE and NSE metrics were explicitly developed for hydrological modeling and are used by hydrologists for decision making regarding the accuracy and consistency of models. MAE and MAPE metrics are widely used in machine learning research. RRMSE

allows for evaluating the performance of models in rivers with flows of different magnitudes and is helpful in cases where it is desired to assess the performance of the same model and rivers with wide channels (higher flows) or narrow ones (smaller flows). The WI index represents the ratio between the mean square error and the potential error and brings relevant information to specialists/hydrologists in the analysis of the calibration of forecast models. In Table 6, O_i and P_i represent observation and prediction, respectively. O represents observation mean, r represents Pearson product–moment correlation coefficient, a is the standard deviation (SD) of the predictions, and the SD of observation. B is the ratio of the prediction and observation means.

Table 6. Performance metrics used in the study.

Metric Acronym	Expression
MAE	$\frac{1}{N} \sum (O_i - P_i) $
MAPE	$\frac{100 \times 1}{N} \sum \frac{ O_i - P_i }{ O_i }$
NSE	$1 - \frac{\sum_{i=1}^N (O_i - P_i)^2}{\sum_{i=1}^N (O_i - O^-)^2}$
KGE	$1 - \sqrt{(r - 1)^2 + (a - 1)^2 + (B - 1)^2}$
WI	$1 - \frac{\sum (O_i - P_i)^2}{\sum (P_i - O^- + O_i - O^-)^2}$
RRMSE	$\frac{\sqrt{\frac{1}{N} \sum (O_i - P_i)^2}}{O^-}$

3. Results and Discussion

3.1. Results

This section presents the results and compares the performances of seven machine learning methods tuned by CMAES: ELM, EN, GPR, LSSVR, RBFNN, SVR, and XGB for predicting monthly streamflow. Methods were assessed considering the metrics given in Table 6. Various internal parameters were attempted in order to reach the optimal models for each method and input case. Table 5 sums up the considered ranges of the internal parameters.

Figure 4 displays the SVR internal parameters C , g , and e . C refers to the penalization parameter, e defines the penalization related to training loss function, and g refers to the kernel coefficient. Analyzing the boxplots, the distributions of the parameters g and e are similar in both cases. In addition, the parameter values of C are higher for Case 1 compared to other cases. C as a regularization parameter has an important role in support vector machines. The strength of regularization is directly proportional to $1/C$. Thus, the estimations are smoother for Case 0 compared to Case 1. The lower RMSE for Case 1 illustrated in the bar plots in Figure 4 (averaged in 50 runs) supports this interpretation.

Test outcomes of the optimal hybrid ELM, EN, GPR, LSSVR, RBFNN, SVR, and XGB models calibrated by CMAES are shown in Tables 7–13 for seven input cases. For input Case 1 (Table 7), SVR showed the best accuracy with the lowest RRMSE (0.375), MAE (412.98), and MAPE (25.91), and the highest WI (0.970), NSE (0.876), and KGE (0.915) in streamflow prediction, and the GPR follows it. At the same time, the EN acts worse than the other models. In other temperature-based input cases, the SVR offers the best accuracy, generally having the lowest SD (values within brackets). The models' efficiency decreases by adding more temperature inputs (from input Case 1 to 3), whereas the T_{t-12} input decreases the prediction results. The third input case provides the highest accuracy for all models among the temperature inputs. The temperature-based SVR model (input Case 3 with T_{t-11} , T_{t-1} , and T_t) has the lowest RRMSE (0.290), MAE (289.21), and the highest WI (0.981), NSE (0.925), and KGE (0.962), and EN provides the worst efficiency. From input Case 1 to Case 3, improvements in the RRMSE (NSE) accuracy of the ELM, EN, GPR, LSSVR,

RBFFNN, SVR, and XGB are 22.1 (6.8), 0.5 (0.6), 21.3 (5.8), 19.5 (5.3), 18.9 (5.3), 22.7 (5.6), and 14% (4%) in streamflow prediction, respectively. The best SVR model utilizing the input Case 3 improves the prediction accuracy (concerning RRMSE) by 7.3, 50, 4.3, 6.5, 7.6, and 12.9% compared to ELM, EN, GPR, LSSVR, RBFFNN, and XGB, respectively.

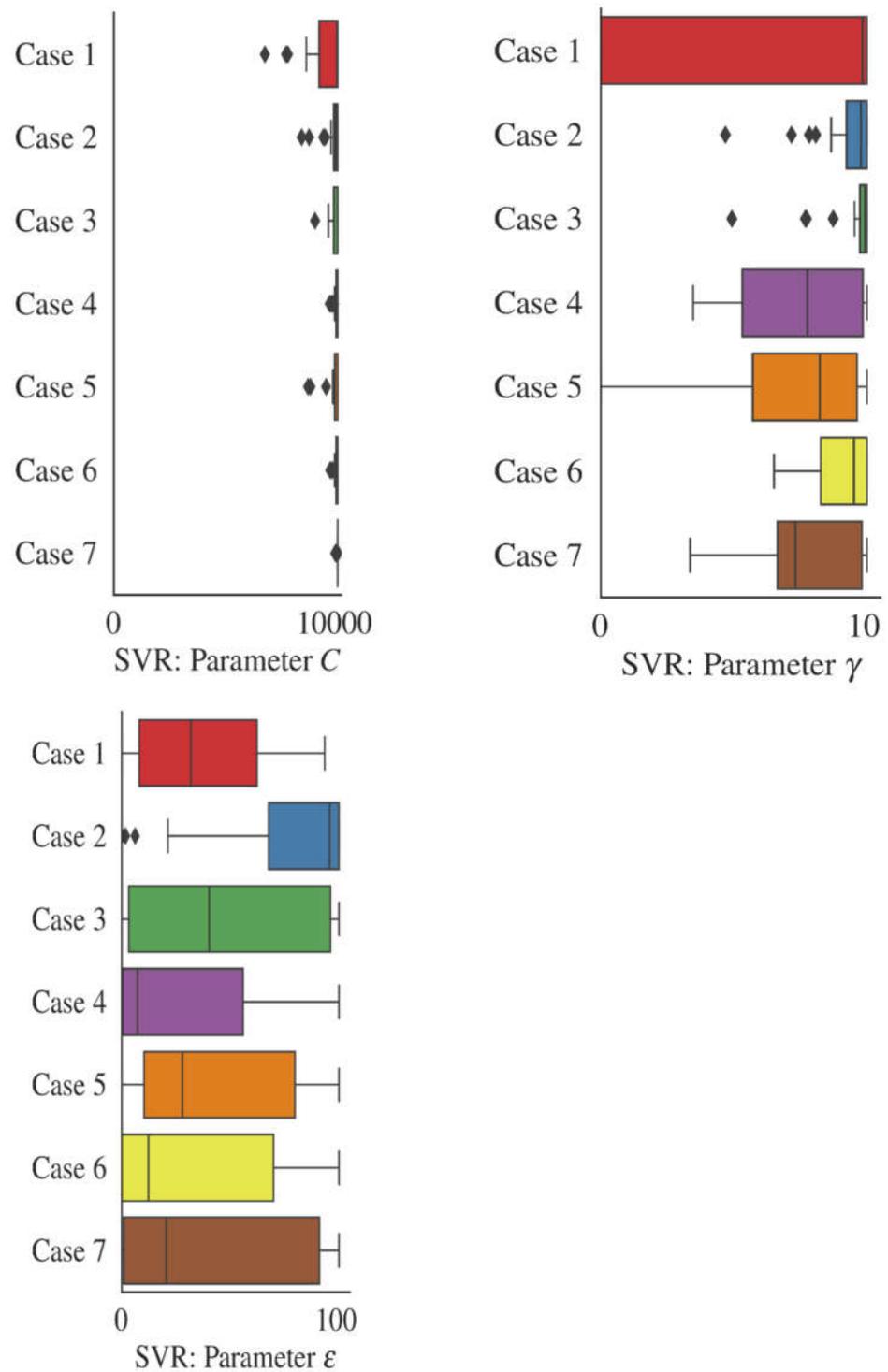


Figure 4. Distribution of the internal parameters for the SVR models over 50 independent runs. Each color represents a case study in Table 2. Black diamond shapes represent outliers.

Table 7. Model results for Case 1.

Estimator	WI	RRMSE	MAE	MAPE	NSE	Kq + GE
ELM	0.965 (0.007)	0.402 (0.044)	441.44 (26.13)	28.28 (3.500)	0.855 (0.037)	0.898 (0.025)
EN	0.908 (0.000)	0.578 (0.000)	875.05 (0.000)	120.11 (0.000)	0.703 (0.000)	0.774 (0.000)
GPR	0.968 (0.001)	0.385 (0.008)	427.99 (10.56)	26.69 (0.300)	0.868 (0.005)	0.904 (0.003)
LSSVR	0.968 (0.001)	0.385 (0.007)	428.95 (11.08)	27.18 (0.726)	0.869 (0.005)	0.904 (0.002)
RBFNN	0.968 (0.004)	0.387 (0.022)	442.57 (53.73)	30.55 (10.98)	0.867 (0.017)	0.904 (0.007)
SVR	0.970 (0.000)	0.375 (0.003)	412.98 (4.13)	25.91 (0.620)	0.876 (0.002)	0.915 (0.004)
XGB	0.968 (0.002)	0.387 (0.012)	434.24 (14.39)	27.37 (0.534)	0.867 (0.009)	0.898 (0.011)

Table 8. Model results for Case 2.

Estimator	WI	RRMSE	MAE	MAPE	NSE	KGE
ELM	0.971 (0.004)	0.363 (0.022)	394.59 (42.45)	26.54 (8.130)	0.883 (0.015)	0.922 (0.013)
EN	0.909 (0.000)	0.577 (0.000)	881.96 (0.000)	118.81 (0.000)	0.704 (0.000)	0.777 (0.000)
GPR	0.971 (0.004)	0.360 (0.023)	350.88 (14.52)	17.21 (1.540)	0.884 (0.015)	0.932 (0.010)
LSSVR	0.968 (0.005)	0.382 (0.026)	369.83 (10.97)	20.53 (1.300)	0.870 (0.018)	0.921 (0.003)
RBFNN	0.971 (0.001)	0.366 (0.009)	358.07 (9.61)	17.40 (0.438)	0.881 (0.006)	0.932 (0.002)
SVR	0.975 (0.00)	0.339 (0.003)	342.25 (5.11)	17.36 (0.646)	0.898 (0.002)	0.943 (0.002)
XGB	0.968 (0.002)	0.389 (0.015)	386.32 (19.42)	17.61 (0.987)	0.866 (0.010)	0.909 (0.009)

Table 9. Model results for Case 3.

Estimator	WI	RRMSE	MAE	MAPE	NSE	KGE
ELM	0.979 (0.003)	0.313 (0.021)	352.43 (40.86)	27.06 (6.670)	0.913 (0.012)	0.929 (0.011)
EN	0.910 (0.000)	0.575 (0.000)	879.59 (0.003)	118.37 (0.001)	0.707 (0.000)	0.778 (0.000)
GPR	0.980 (0.001)	0.303 (0.009)	307.32 (13.57)	16.12 (1.730)	0.918 (0.005)	0.947 (0.007)

Table 9. Cont.

Estimator	WI	RRMSE	MAE	MAPE	NSE	KGE
LSSVR	0.979 (0.002)	0.310 (0.013)	328.71 (21.43)	20.91 (1.880)	0.915 (0.007)	0.938 (0.004)
RBFNN	0.978 (0.002)	0.314 (0.015)	328.96 (34.73)	18.91 (8.270)	0.913 (0.009)	0.947 (0.009)
SVR	0.981 (0.00)	0.290 (0.001)	289.21 (5.09)	16.19 (0.669)	0.925 (0.00)	0.962 (0.001)
XGB	0.976 (0.003)	0.333 (0.019)	342.84 (23.62)	16.99 (1.520)	0.902 (0.011)	0.928 (0.012)

Table 10. Model results for Case 4.

Estimator	WI	RRMSE	MAE	MAPE	NSE	KGE
ELM	0.972 (0.012)	0.356 (0.063)	394.67 (36.17)	31.46 (8.530)	0.884 (0.053)	0.908 (0.026)
EN	0.909 (0.000)	0.578 (0.000)	885.37 (0.000)	119.59 (0.000)	0.704 (0.000)	0.777 (0.000)
GPR	0.973 (0.002)	0.350 (0.014)	358.39 (17.63)	18.14 (2.580)	0.891 (0.009)	0.931 (0.014)
LSSVR	0.976 (0.000)	0.334 (0.006)	347.08 (5.79)	20.15 (1.150)	0.901 (0.004)	0.915 (0.005)
RBFNN	0.925 (0.192)	0.900 (0.620)	376.58 (6.103)	21.21 (9490.7)	0.876 (0.005)	0.864 (0.309)
SVR	0.979 (0.000)	0.308 (0.003)	307.08 (5.060)	17.06 (0.820)	0.916 (0.002)	0.956 (0.002)
XGB	0.975 (0.002)	0.340 (0.017)	364.48 (22.67)	17.95 (2.210)	0.897 (0.010)	0.930 (0.012)

Table 11. Model results for Case 5.

Estimator	WI	RRMSE	MAE	MAPE	NSE	KGE
ELM	0.858 (0.020)	0.700 (0.090)	915.48 (45.45)	71.82 (2.850)	0.558 (0.142)	0.647 (0.052)
EN	0.834 (0.000)	0.715 (0.000)	999.14 (0.000)	85.40 (0.000)	0.547 (0.000)	0.562 (0.000)
GPR	0.854 (0.013)	0.761 (0.053)	919.25 (33.68)	72.80 (2.110)	0.484 (0.068)	0.718 (0.008)
LSSVR	0.887 (0.001)	0.647 (0.006)	848.64 (5.78)	65.94 (0.455)	0.628 (0.007)	0.748 (0.002)
RBFNN	0.851 (0.011)	0.775 (0.048)	931.38 (27.37)	72.58 (3.280)	0.465 (0.063)	0.715 (0.009)

Table 11. *Cont.*

Estimator	WI	RRMSE	MAE	MAPE	NSE	KGE
SVR	0.871 (0.002)	0.719 (0.004)	825.33 (13.78)	42.27 (3.210)	0.541 (0.005)	0.746 (0.004)
XGB	0.873 (0.007)	0.707 (0.030)	862.83 (14.08)	70.24 (1.310)	0.556 (0.036)	0.747 (0.004)

Table 12. Model results for Case 6.

Estimator	WI	RRMSE	MAE	MAPE	NSE	KGE
ELM	0.983 (0.002)	0.273 (0.017)	303.21 (26.09)	17.69 (3.960)	0.934 (0.009)	0.958 (0.007)
EN	0.959 (0.000)	0.409 (0.000)	527.22 (0.000)	47.25 (0.000)	0.852 (0.000)	0.893 (0.000)
GPR	0.983 (0.00)	0.275 (0.007)	292.42 (3.24)	13.49 (0.306)	0.933 (0.003)	0.950 (0.005)
LSSVR	0.983 (0.00)	0.280 (0.007)	294.22 (3.01)	13.41 (0.416)	0.930 (0.003)	0.948 (0.004)
RBFNN	0.982 (0.001)	0.287 (0.011)	295.48 (7.48)	12.14 (0.594)	0.927 (0.005)	0.956 (0.002)
SVR	0.983 (0.00)	0.277 (0.004)	283.49 (4.91)	13.24 (1.030)	0.932 (0.002)	0.960 (0.00400)
XGB	0.979 (0.002)	0.309 (0.015)	315.66 (15.48)	15.98 (1.900)	0.915 (0.008)	0.948 (0.007)

Table 13. Model results for Case 7.

Estimator	WI	RRMSE	MAE	MAPE	NSE	KGE
ELM	0.984 (0.003)	0.266 (0.019)	288.71 (14.91)	15.54 (1.690)	0.937 (0.010)	0.962 (0.005)
EN	0.972 (0.000)	0.341 (0.000)	389.77 (0.000)	24.34 (0.000)	0.897 (0.000)	0.926 (0.000)
GPR	0.985 (0.00)	0.261 (0.009)	256.68 (9.810)	11.69 (0.597)	0.939 (0.004)	0.967 (0.003)
LSSVR	0.985 (0.00)	0.256 (0.007)	255.54 (13.78)	12.28 (1.310)	0.942 (0.003)	0.966 (0.003)
RBFNN	0.978 (0.020)	0.301 (0.092)	293.20 (70.77)	13.11 (4.670)	0.912 (0.082)	0.950 (0.039)
SVR	0.984 (0.00)	0.266 (0.009)	263.44 (7.080)	12.44 (0.715)	0.937 (0.004)	0.962 (0.006)
XGB	0.977 (0.002)	0.317 (0.017)	338.54 (27.27)	17.01 (3.410)	0.910 (0.009)	0.949 (0.006)

Tables 11–13 sum up the testing outcomes of the implemented hybrid models for the discharge inputs (input Cases 5 to 7). The tables clearly show that only discharge (streamflow) of one previous month (Q_{t-1}) is insufficient in modeling streamflow. However, inclusion of Q_{t-11} and Q_{t-12} considering correlation analysis considerably increases the models' accuracies; increase (decrease) in RRMSE (NSE) of the ELM, EN, GPR, LSSVR, RBFNN, SVR, and XGB is 62 (68), 52 (64), 66 (94), 60 (50), 61 (96), 63 (73), and 55% (64%) from input Case 5 (Table 11) to input Case 7 (Table 13), respectively. The best discharge-based LSSVR model (input Case 7 with Q_{t-12} , Q_{t-11} , and Q_{t-1}) has the lowest RRMSE (0.256) and MAE (255.54) and the highest WI (0.985) and NSE (0.942), while the GPR has the lowest MAPE (11.69) and the highest WI (0.985) and KGE (0.967). The LSSVR performs slightly better than the GPR. SVR also has good accuracy and closely follows the LSSVR and GPR in streamflow prediction, using only previous values as inputs.

Comparison of the outcomes provided by the temperature-based and discharge-based models reveals that the discharge-based models have higher accuracy than the temperature-based models; decrease in RRMSE from 0.290 (SVR) to 0.256 (LSSVR), in MAE from 289.21 (SVR) to 255.54 (LSSVR), in MAPE from 16.12 (SVR) to 11.69 (GPR) and increase in WI from 0.981 (SVR) to 0.985 (GPR), in NSE from 0.925 (SVR) to 0.942 (LSSVR), and in KGE from 0.962 (SVR) to 0.967 (GPR). However, TB models also provide good accuracy.

For Cases 1 to 7, Figure 5 depicts the scatterplots of the observed and best model (SVR)-predicted streamflow in the test stage. As clearly observed from the graphs, temperature-based models also provide good predictions, especially with input Case 3. Considerable improvement in prediction efficiency by including Q_{t-11} and Q_{t-12} inputs (Case 6 and 7) is observed. Figure 6 illustrates the radar charts of the metrics used for model assessments. They were obtained, averaging over 50 runs for each method. These graphs were provided for visually justifying the statistics given in Tables 7–13.

The lowest accuracy of EN is observed from the charts, while the SVR generally provides superior accuracy. In some cases (Case 6 and 7), the LSSVR and XGB may perform better than the SVR. Taylor diagrams of the optimal models for all cases (averaged over 50 runs) are given in Figure 7. The diagrams show that the SVR generally has the lowest square error, highest association, and similar SD to observation than the other models. The inferior output of EN is depicted in the diagrams.

3.2. Uncertainty Analysis

The uncertainty analysis is based on the logarithm of the prediction errors for the test dataset $e_j = \log_{10}(Q_{tp_j}) - \log_{10}(Q_{tm_j}), j = 1, \dots, N$ that are used to calculate the mean and standard deviation of the prediction errors given by:

$$\bar{e} = \sum_{j=1}^N e_j \tag{29}$$

and

$$S_e = \sqrt{\frac{\sum_{j=1}^N (e_j - \bar{e})^2}{N - 1}} \tag{30}$$

where \bar{e} and S_e are the mean and standard deviation of the logarithm errors, Q_{tm_j} is the measured streamflow and Q_{tp_j} is the predicted streamflow for the i th sample. Using the error above, an approximately 95% confidence band around the predicted value Q_{tm_j} is defined by $(Q_{tm_j} \times 10^{-\bar{e}-1.96S_e}, Q_{tm_j} \times 10^{-\bar{e}+1.96S_e})$.

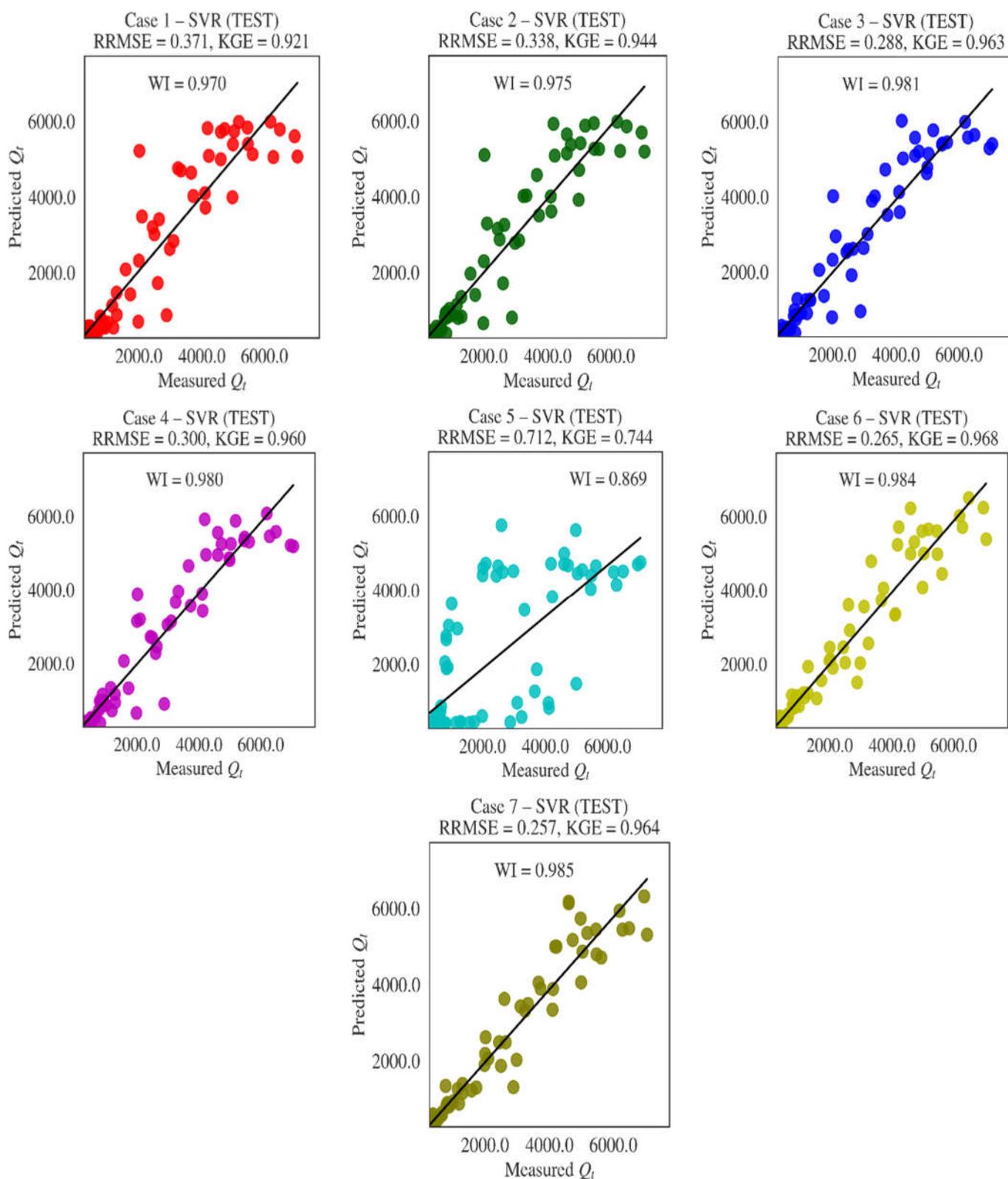


Figure 5. Scatterplots of the observed and predicted streamflow by the best models in the test stage.

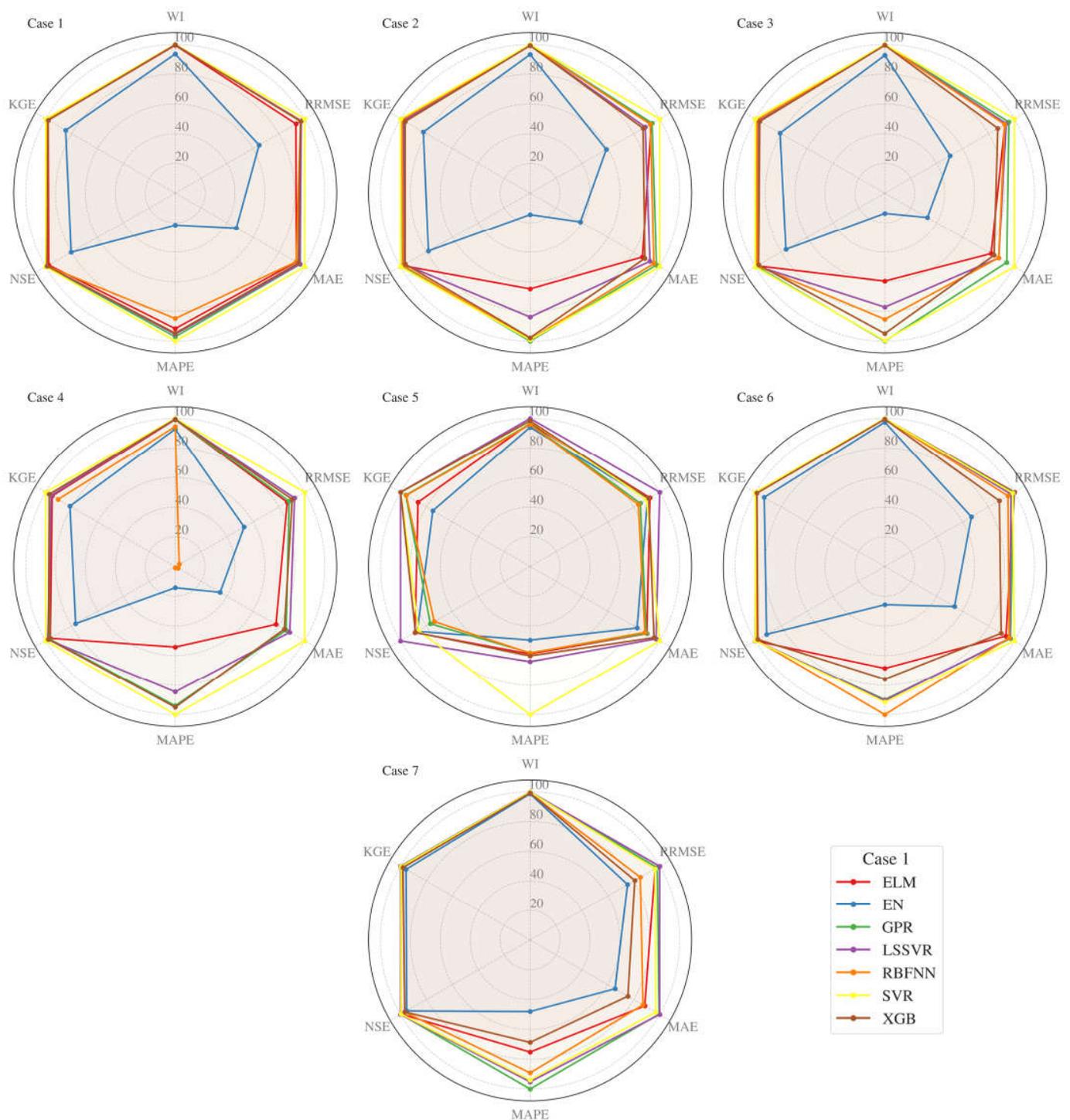


Figure 6. Radar charts for all metrics (averaged over 50 runs).

The uncertainty analysis allows for comparing the proposed models and the input cases in this study. Table 14 presents a quantitative evaluation of uncertainties in river flow forecasting for each machine learning model and each input case to see the effects of the uncertainty study, the models' mean prediction errors, the width of the uncertainty band, and the 95% prediction interval error. Negative mean prediction error averages indicate that the prediction models underestimated the observed values. On the other hand, positive values indicate that they overestimated the outcomes. Smaller mean predicted errors represent the ability to model the streamflow. The SVR model leads to smaller

overall mean prediction errors for all input cases. The uncertainty bands for both models ranged from 0.071 to 0.320, with SVR, XGB, LSSVR, and RBFNN models having smaller uncertainty bands.

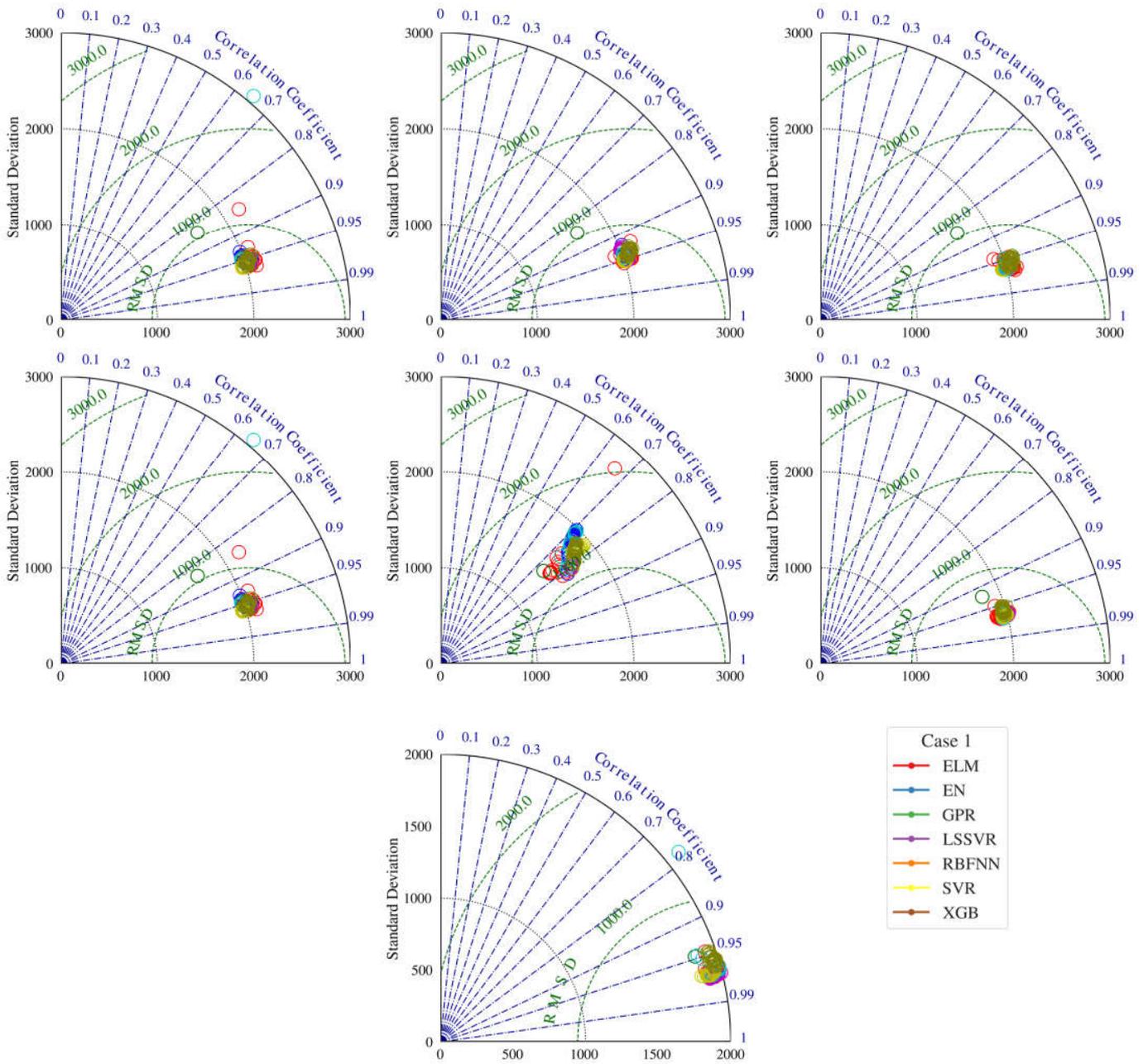


Figure 7. Taylor diagrams of the optimal models for all cases (averaged over 50 runs).

Table 14. Uncertainty estimates for machine learning models.

Model	Case	Mean Prediction Error	Width of Uncertainty Band	95% Prediction Error Interval
ELM	Case 1	−0.009	0.157	0.502 to 2.07
	Case 2	−0.031	0.154	0.535 to 2.15
	Case 3	−0.027	0.176	0.481 to 2.36
	Case 4	−0.034	0.157	0.533 to 2.19
	Case 5	+0.061	0.320	0.205 to 3.68
	Case 6	+0.013	0.085	0.663 to 1.42
	Case 7	+0.009	0.074	0.701 to 1.37
EN	Case 1	+0.166	0.260	0.211 to 2.21
	Case 2	+0.148	0.298	0.185 to 2.73
	Case 3	+0.161	0.265	0.209 to 2.28
	Case 4	+0.162	0.265	0.208 to 2.28
	Case 5	+0.131	0.281	0.208 to 2.63
	Case 6	−0.096	0.294	0.331 to 4.70
	Case 7	−0.008	0.125	0.579 to 1.79
GPR	Case 1	+0.007	0.140	0.524 to 1.85
	Case 2	−0.007	0.110	0.618 to 1.67
	Case 3	−0.006	0.101	0.642 to 1.60
	Case 4	−0.012	0.108	0.632 to 1.67
	Case 5	+0.084	0.259	0.256 to 2.65
	Case 6	+0.016	0.076	0.684 to 1.36
	Case 7	+0.009	0.071	0.710 to 1.35
LSSVR	Case 1	+0.003	0.144	0.519 to 1.90
	Case 2	−0.006	0.122	0.586 to 1.76
	Case 3	−0.012	0.115	0.611 to 1.73
	Case 4	−0.017	0.126	0.590 to 1.84
	Case 5	+0.080	0.262	0.255 to 2.71
	Case 6	+0.015	0.075	0.688 to 1.36
	Case 7	+0.009	0.076	0.695 to 1.38

Table 14. Cont.

Model	Case	Mean Prediction Error	Width of Uncertainty Band	95% Prediction Error Interval
RBFNN	Case 1	−0.020	0.159	0.509 to 2.15
	Case 2	−0.010	0.111	0.622 to 1.69
	Case 3	−0.004	0.101	0.639 to 1.59
	Case 4	−0.016	0.126	0.587 to 1.83
	Case 5	+0.088	0.268	0.243 to 2.74
	Case 6	+0.015	0.078	0.678 to 1.38
	Case 7	+0.009	0.073	0.704 to 1.36
SVR	Case 1	−0.002	0.143	0.526 to 1.92
	Case 2	−0.018	0.118	0.611 to 1.77
	Case 3	−0.014	0.107	0.636 to 1.67
	Case 4	−0.020	0.112	0.632 to 1.74
	Case 5	+0.002	0.282	0.278 to 3.55
	Case 6	+0.005	0.076	0.703 to 1.39
	Case 7	+0.00	0.076	0.708 to 1.41
XGB	Case 1	+0.006	0.155	0.490 to 1.98
	Case 2	−0.014	0.110	0.629 to 1.70
	Case 3	−0.013	0.103	0.648 to 1.64
	Case 4	−0.017	0.102	0.655 to 1.65
	Case 5	+0.085	0.259	0.256 to 2.64
	Case 6	+0.011	0.082	0.674 to 1.41
	Case 7	+0.012	0.092	0.643 to 1.47

In the experiments conducted for this research, the variations of the input parameters were modeled using the uniform distribution with the lower and upper bounds given by the minimum and maximum values for each parameter, as shown in Table 1. The deterministic outcome is calculated for each model developed in this paper for each MCS run. A total of 250,000 outcomes were calculated. The mean absolute deviation (MAD) around the median of the output distribution is written as

$$MAD = \frac{1}{250,000} \sum_{i=1}^{250,000} |Q_{tpi} - median(Q_{tp})| \quad (31)$$

while the uncertainty of the model output can be given by

$$Uncertainty\% = \frac{100 \times MAD}{median(Q_{tp})} \quad (32)$$

where Q_{tpi} is the predicted streamflow for the i th sample. A detailed description of the MCS method can be found in [68]. Uncertainty analysis results for the Q_i using the developed models are presented in Table 14. The uncertainty analysis shows that the SVR generally

has lower uncertainty than other machine learning methods in streamflow prediction, except for the EN. Even though EN has the lowest uncertainty, its prediction efficiency is very low in all cases. It is clear from the figure that the SVR generally has lower uncertainty compared to other methods. Additionally, increasing input parameters increases the methods' prediction efficiency (decrease in RMSE).

Table 15 shows the uncertainty analysis using the Monte Carlo simulation. The uncertainty analysis shows that SVR, LSSVR, GPR, and XGB produced lower uncertainty than other machine learning methods in streamflow prediction. Figure 8 shows the scatterplot of uncertainty (%) and relative root mean squared error for all models. Each pair (x,y) represents a particular model, depicted differently. The size of the point represents the number of features in the model. To facilitate the visualization, Cases 1–7 are shrunk to C1–C7. The scatter plots presented in Figure 8 show that the machine learning models using the input Cases 6 and 7 are near the left bottom corner of the figure, indicating that the input of these cases produced models with smaller uncertainty and smaller RRMSE. Cases C2 and C3 lead to models with higher uncertainties and moderate RRMSE. In contrast, input Case 5, although producing models with small uncertainties, leads to higher RRMSE values.

Table 15. Uncertainty analysis for the test set using Monte Carlo simulation (MCS).

Model	Case	No. Features	Median	MAD	Uncertainty %
ELM	Case 1	1	591.9	1307.0	220.8
	Case 2	2	500.6	1233.0	246.3
	Case 3	3	385.6	1325.2	343.7
	Case 4	4	712.2	967.9	135.9
	Case 5	1	3494.0	1177.7	33.7
	Case 6	2	4492.1	1519.7	33.8
	Case 7	3	4517.9	1644.5	36.4
EN	Case 1	1	1763.3	1381.6	78.4
	Case 2	2	1765.9	1240.2	70.2
	Case 3	3	1768.0	1085.4	61.4
	Case 4	4	1765.5	987.8	55.9
	Case 5	1	3159.1	1248.7	39.5
	Case 6	2	3989.3	1335.2	33.5
	Case 7	3	3787.5	1126.0	29.7
GPR	Case 1	1	572.1	1307.3	228.5
	Case 2	2	722.6	1018.9	141.0
	Case 3	3	664.4	891.6	134.2
	Case 4	4	770.5	815.7	105.9
	Case 5	1	3595.2	1120.7	31.2
	Case 6	2	4846.8	1531.6	31.6
	Case 7	3	4594.2	1321.5	28.8

Table 15. Cont.

Model	Case	No. Features	Median	MAD	Uncertainty %
LSSVR	Case 1	1	495.7	1317.2	265.7
	Case 2	2	739.8	1421.5	192.1
	Case 3	3	1162.0	835.6	71.9
	Case 4	4	707.2	1300.9	184.0
	Case 5	1	3307.0	1190.1	36.0
	Case 6	2	4626.8	1303.6	28.2
	Case 7	3	4736.3	1452.5	30.7
RBFNN	Case 1	1	449.8	1353.0	300.8
	Case 2	2	987.7	1033.0	104.6
	Case 3	3	939.3	844.1	89.9
	Case 4	4	1540.7	1018.3	66.1
	Case 5	1	3593.4	1109.9	30.9
	Case 6	2	4803.6	1587.2	33.0
	Case 7	3	4771.2	1678.5	35.2
SVR	Case 1	1	560.0	1264.4	225.8
	Case 2	2	917.8	932.8	101.6
	Case 3	3	1208.9	680.6	56.3
	Case 4	4	682.6	499.3	73.1
	Case 5	1	4407.5	1305.1	29.6
	Case 6	2	4700.0	1480.9	31.5
	Case 7	3	3852.7	1264.5	32.8
XGB	Case 1	1	629.4	1271.1	201.9
	Case 2	2	726.5	1035.4	142.5
	Case 3	3	768.3	823.1	107.1
	Case 4	4	633.9	625.6	98.7
	Case 5	1	3868.2	1085.5	28.1
	Case 6	2	4840.9	1374.7	28.4
	Case 7	3	3749.9	1262.1	33.7

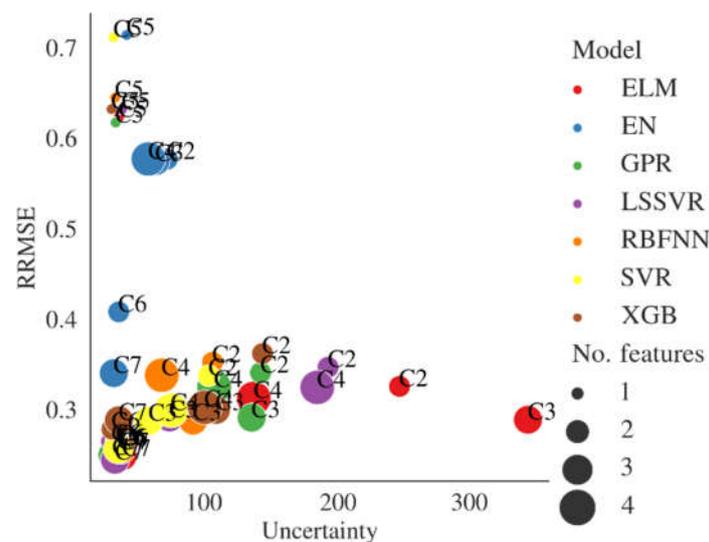


Figure 8. Scatter plots comparing the uncertainty and root mean squared error for all models.

3.3. Discussion

By the presented study, the accuracy of seven machine learning methods, ELM, EN, GPR, SVR, LSSVR, XGB, and RBFNN, tuned with covariance matrix adaptation evolution strategy, was assessed and compared in streamflow prediction considering different input combinations involving temperature and previous discharge values. The inputs of the models were decided using correlation analysis and it was observed that this method is very useful in the determination of the most suitable inputs for the machine learning methods. This finding is consistent with the previous research [25,69–72], which reported the necessity of correlation analysis in defining optimal inputs. It is clearly observed from quantitative and graphical assessments that the EN method provides inferior predictions. The low accuracy of the EN method may be explained by its linear structure, which prevents it from adequately mapping the streamflow phenomenon. It is observed from statistical and graphical comparison that the temperature-based models also provide good accuracy in streamflow prediction. They can be considered better alternatives to the discharge-based models, especially in case of unavailability of discharge (streamflow) data. Measuring streamflow is a very difficult task, especially in developing countries, for technical reasons. In such cases, temperature-based models can be successfully implemented. The results align with the earlier studies [25,72]. Adnan et al. [25] employed LSSVR, MARS, and optimally pruned ELM and M5 model trees in streamflow prediction and found good accuracies from the temperature-based models. In the study by Adnan et al. [72], they implemented the group method of data handling neural networks, dynamic evolving neural fuzzy system, and MARS methods in monthly streamflow prediction. They obtained promising accuracies from the temperature-based models. An advantage of using the machine learning model is the accurate streamflow prediction commonly reported in the literature. On the other hand, the machine learning model's performance mainly depends on the proper choice of internal parameters, which is a challenge for data modelers. Moreover, parameters can vary across different data sets and result in performance disparities. In this context, an advantage of the proposed approach is the automatic adjustment of models' parameters through the CMAES algorithm, allowing for the exploitation of the models' capabilities, considering the data's specificities. Furthermore, the proposed strategy employs a cross-validation strategy to avoid data leakage, providing a realistic forecast of monthly flow data. However, the lack of interpretability is an intrinsic aspect of machine learning models. Although they produce accurate predictions, the relationships between input and output variables are challenging for humans to interpret. Although they have a rigid and concise mathematical formulation, machine learning models can be built with several data processing procedures, which makes a posteriori analysis difficult.

Adnan et al. [5] compared the accuracy of optimally pruned ELM, MARS, ANN, M5 model tree, and neuro-fuzzy tuned with particle swarm optimization in streamflow prediction, and the best ELM gave the NSE of 0.761 and 0.711 for the Fujiangqiao and Shehang stations in China. Adnan et al. [27] compared the abovementioned machine learning methods and they obtained an NSE of 0.886 from the best MARS model, which uses precipitation, temperature, and streamflow inputs in prediction streamflow. Adnan et al. [72] assessed the abovementioned three machine learning methods in streamflow prediction and they found that the dynamic evolving neural fuzzy system performed the best with the NSE of 0.915. Compared to the abovementioned studies, the accuracy of the best SVR-CMAES model developed in this study seems promising, with the NSE of 0.937.

4. Conclusions

- This research evaluated the feasibility of seven hybrid machine learning methods in monthly streamflow forecasting. In the hybridization process, the metaheuristic covariance matrix adaptation evolution strategy assisted in adjusting the internal parameters of the models. In addition, an uncertainty analysis was performed on the resulting models. Methods were compared to each other considering the several metrics (e.g., WI, RRMSE, MAE, MAPE, NSE, and KGE together with standard deviation) and visual methods, such as radar chart, scatter, and Taylor diagrams. Seven input cases were taken into account, including temperature and discharge data. All the implemented methods provided better efficiency for discharge input cases, while temperature input cases also produced promising predictions. Among the implemented methods, the SVR generally performed superior to the other methods, especially for the temperature input cases. In contrast, the LSSVR and GPR were found to do better than the SVR in some discharge input cases. The EN method provided the worst streamflow predictions, according to all evaluation statistics and visual comparison. Uncertainty analysis revealed that the SVR generally has low uncertainty compared to other machine learning methods and the EN has the lowest uncertainty. Overall, temperature-based SVR models are highly recommended in monthly streamflow prediction. These outcomes are very useful in practical applications, especially in developing countries. Temperature data are easily measured, and the models that only use such data will be very beneficial for streamflow predictions in the basins where streamflow data are missing because of technical/economic reasons. The main limitation of this study is the use of limited data from one location. More data from different climatic regions are required to justify the accuracy of the proposed models. The covariance matrix adaptation evolution strategy was successfully applied for improving the accuracy of the ELM, EN, GPR, LSSVR, RBFNN, SVR, and XGB methods. This strategy can also be employed for other machine learning methods, such as ANN, ANFIS, and deep learning in future studies.

Author Contributions: Conceptualization: R.M.A.I., O.K. and L.G.; formal analysis: L.G., S.S. and O.K.; validation: O.K., L.G. and R.M.A.I.; supervision: O.K. and S.T.; writing—original draft: O.K., L.G. and R.M.A.I.; visualization: R.M.A.I. and S.T.; investigation: O.K., S.S. and S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study will be available on an interesting request from the corresponding author.

Conflicts of Interest: There is no conflict of interest in this study.

References

1. Erdal, H.I.; Karakurt, O. Advancing monthly streamflow prediction accuracy of CART models using ensemble learning paradigms. *J. Hydrol.* **2013**, *477*, 119–128. [\[CrossRef\]](#)
2. Yaseen, Z.M.; El-Shafie, A.; Jaafar, O.; Afan, H.A.; Sayl, K.N. Artificial intelligence based models for stream-flow forecasting: 2000–2015. *J. Hydrol.* **2015**, *530*, 829–844. [\[CrossRef\]](#)
3. Rasouli, K.; Hsieh, W.W.; Cannon, A.J. Daily streamflow forecasting by machine learning methods with weather and climate inputs. *J. Hydrol.* **2012**, *414–415*, 284–293. [\[CrossRef\]](#)
4. Liu, Z.; Zhou, P.; Zhang, Y. A Probabilistic Wavelet–Support Vector Regression Model for Streamflow Forecasting with Rainfall and Climate Information Input. *J. Hydrometeorol.* **2015**, *16*, 2209–2229. [\[CrossRef\]](#)
5. Adnan, R.M.; Liang, Z.; Trajkovic, S.; Zounemat-Kermani, M.; Li, B.; Kisi, O. Daily streamflow prediction using optimally pruned extreme learning machine. *J. Hydrol.* **2019**, *577*, 123981. [\[CrossRef\]](#)
6. Kisi, O.; Heddami, S.; Keshtegar, B.; Piri, J.; Adnan, R.M. Predicting Daily Streamflow in a Cold Climate Using a Novel Data Mining Technique: Radial M5 Model Tree. *Water* **2022**, *14*, 1449. [\[CrossRef\]](#)
7. Faye, C. *Comparative Analysis of Meteorological Drought Based on the SPI and SPEI Indices*; Ziguinchor University: Ziguinchor, Senegal, 2022.
8. Dawson, C.; Wilby, R. An artificial neural network approach to rainfall-runoff modelling. *Hydrol. Sci. J.* **1998**, *43*, 47–66. [\[CrossRef\]](#)
9. Demirel, M.C.; Venancio, A.; Kahya, E. Flow forecast by SWAT model and ANN in Pracana basin, Portugal. *Adv. Eng. Softw.* **2009**, *40*, 467–473. [\[CrossRef\]](#)
10. Bhadra, A.; Bandyopadhyay, A.; Singh, R.; Raghuvanshi, N.S. Rainfall-Runoff Modeling: Comparison of Two Approaches with Different Data Requirements. *Water Resour. Manag.* **2010**, *24*, 37–62. [\[CrossRef\]](#)
11. Sudheer, C.; Maheswaran, R.; Panigrahi, B.K.; Mathur, S. A hybrid SVM-PSO model for forecasting monthly streamflow. *Neural Comput. Appl.* **2014**, *24*, 1381–1389. [\[CrossRef\]](#)
12. Raghavendra, N.S.; Deka, P.C. Support vector machine applications in the field of hydrology: A review. *Appl. Soft Comput.* **2014**, *19*, 372–386. [\[CrossRef\]](#)
13. Nourani, V.; Baghanam, A.H.; Adamowski, J.; Kisi, O. Applications of hybrid wavelet–Artificial Intelligence models in hydrology: A review. *J. Hydrol.* **2014**, *514*, 358–377. [\[CrossRef\]](#)
14. Ibrahim, K.S.M.H.; Huang, Y.F.; Ahmed, A.N.; Koo, C.H.; El-Shafie, A. A review of the hybrid artificial intelligence and optimization modelling of hydrological streamflow forecasting. *Alex. Eng. J.* **2021**, *61*, 279–303. [\[CrossRef\]](#)
15. Kagoda, P.A.; Ndiritu, J.; Ntuli, C.; Mwaka, B. Application of radial basis function neural networks to short-term streamflow forecasting. *Phys. Chem. Earth Parts A/B/C* **2010**, *35*, 571–581. [\[CrossRef\]](#)
16. Sun, A.Y.; Wang, D.; Xu, X. Monthly streamflow forecasting using Gaussian Process Regression. *J. Hydrol.* **2014**, *511*, 72–81. [\[CrossRef\]](#)
17. Mehr, A.D.; Kahya, E.; Şahin, A.; Nazemosadat, M.J. Successive-station monthly streamflow prediction using different artificial neural network algorithms. *Int. J. Environ. Sci. Technol.* **2015**, *12*, 2191–2200. [\[CrossRef\]](#)
18. Kisi, O. Streamflow Forecasting and Estimation Using Least Square Support Vector Regression and Adaptive Neuro-Fuzzy Embedded Fuzzy c-means Clustering. *Water Resour. Manag.* **2015**, *29*, 5109–5127. [\[CrossRef\]](#)
19. Yaseen, Z.M.; Kisi, O.; Demir, V. Enhancing Long-Term Streamflow Forecasting and Predicting using Periodicity Data Component: Application of Artificial Intelligence. *Water Resour. Manag.* **2016**, *30*, 4125–4151. [\[CrossRef\]](#)
20. Modaresi, F.; Araghinejad, S.; Ebrahimi, K. A Comparative Assessment of Artificial Neural Network, Generalized Regression Neural Network, Least-Square Support Vector Regression, and K-Nearest Neighbor Regression for Monthly Streamflow Forecasting in Linear and Nonlinear Conditions. *Water Resour. Manag.* **2018**, *32*, 243–258. [\[CrossRef\]](#)
21. Worland, S.C.; Farmer, W.; Kiang, J.E. Improving predictions of hydrological low-flow indices in ungaged basins using machine learning. *Environ. Model. Softw.* **2018**, *101*, 169–182. [\[CrossRef\]](#)
22. Hadi, S.J.; Abba, S.I.; Sammen, S.S.; Salih, S.Q.; Al-Ansari, N.; Yaseen, Z.M. Non-Linear Input Variable Selection Approach Integrated With Non-Tuned Data Intelligence Model for Streamflow Pattern Simulation. *IEEE Access* **2019**, *7*, 141533–141548. [\[CrossRef\]](#)
23. Li, Y.; Liang, Z.; Hu, Y.; Li, B.; Xu, B.; Wang, D. A multi-model integration method for monthly streamflow prediction: Modified stacking ensemble strategy. *J. Hydroinform.* **2020**, *22*, 310–326. [\[CrossRef\]](#)
24. Ni, L.; Wang, D.; Wu, J.; Wang, Y.; Tao, Y.; Zhang, J.; Liu, J. Streamflow forecasting using extreme gradient boosting model coupled with Gaussian mixture model. *J. Hydrol.* **2020**, *586*, 124901. [\[CrossRef\]](#)
25. Adnan, R.M.; Liang, Z.; Heddami, S.; Zounemat-Kermani, M.; Kisi, O.; Li, B. Least square support vector machine and multivariate adaptive regression splines for streamflow prediction in mountainous basin using hydro-meteorological data as inputs. *J. Hydrol.* **2020**, *586*, 124371. [\[CrossRef\]](#)
26. Thapa, S.; Zhao, Z.; Li, B.; Lu, L.; Fu, D.; Shi, X.; Tang, B.; Qi, H. Snowmelt-Driven Streamflow Prediction Using Machine Learning Techniques (LSTM, NARX, GPR, and SVR). *Water* **2020**, *12*, 1734. [\[CrossRef\]](#)
27. Jiang, Y.; Bao, X.; Hao, S.; Zhao, H.; Li, X.; Wu, X. Monthly Streamflow Forecasting Using ELM-IPSO Based on Phase Space Reconstruction. *Water Resour. Manag.* **2020**, *34*, 3515–3531. [\[CrossRef\]](#)
28. Malik, A.; Tikhamarine, Y.; Souag-Gamane, D.; Kisi, O.; Pham, Q.B. Support vector regression optimized by meta-heuristic algorithms for daily streamflow prediction. *Stoch. Hydrol. Hydraul.* **2020**, *34*, 1755–1773. [\[CrossRef\]](#)

29. Parisouj, P.; Mohebzadeh, H.; Lee, T. Employing Machine Learning Algorithms for Streamflow Prediction: A Case Study of Four River Basins with Different Climatic Zones in the United States. *Water Resour. Manag.* **2020**, *34*, 4113–4131. [[CrossRef](#)]
30. Niu, W.-J.; Feng, Z.-K. Evaluating the performances of several artificial intelligence methods in forecasting daily streamflow time series for sustainable water resources management. *Sustain. Cities Soc.* **2021**, *64*, 102562. [[CrossRef](#)]
31. Jiang, Q.; Cheng, Y.; Le, H.; Li, C.; Liu, P.X. A Stacking Learning Model Based on Multiple Similar Days for Short-Term Load Forecasting. *Mathematics* **2022**, *10*, 2446. [[CrossRef](#)]
32. Yang, C.-H.; Shao, J.-C.; Liu, Y.-H.; Jou, P.-H.; Lin, Y.-D. Application of Fuzzy-Based Support Vector Regression to Forecast of International Airport Freight Volumes. *Mathematics* **2022**, *10*, 2399. [[CrossRef](#)]
33. Su, H.; Peng, X.; Liu, H.; Quan, H.; Wu, K.; Chen, Z. Multi-Step-Ahead Electricity Price Forecasting Based on Temporal Graph Convolutional Network. *Mathematics* **2022**, *10*, 2366. [[CrossRef](#)]
34. De-Prado-Gil, J.; Zaid, O.; Palencia, C.; Martínez-García, R. Prediction of Splitting Tensile Strength of Self-Compacting Recycled Aggregate Concrete Using Novel Deep Learning Methods. *Mathematics* **2022**, *10*, 2245. [[CrossRef](#)]
35. Samadianfard, S.; Jarhan, S.; Salwana, E.; Mosavi, A.; Shamshirband, S.; Akib, S. Support Vector Regression Integrated with Fruit Fly Optimization Algorithm for River Flow Forecasting in Lake Urmia Basin. *Water* **2019**, *11*, 1934. [[CrossRef](#)]
36. Tikhamarine, Y.; Souag-Gamane, D.; Kisi, O. A new intelligent method for monthly streamflow prediction: Hybrid wavelet support vector regression based on grey wolf optimizer (WSVR-GWO). *Arab. J. Geosci.* **2019**, *12*, 540. [[CrossRef](#)]
37. Hansen, N.; Müller, S.D.; Koumoutsakos, P. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evol. Comput.* **2003**, *11*, 1–18. [[CrossRef](#)]
38. Rouchier, S.; Woloszyn, M.; Kedowide, Y.; Bejat, T. Identification of the hygrothermal properties of a building envelope material by the covariance matrix adaptation evolution strategy. *J. Build. Perform. Simul.* **2016**, *9*, 101–114. [[CrossRef](#)]
39. Liang, Y.; Wang, X.; Zhao, H.; Han, T.; Wei, Z.; Li, Y. A covariance matrix adaptation evolution strategy variant and its engineering application. *Appl. Soft Comput.* **2019**, *83*, 105680. [[CrossRef](#)]
40. Chen, G.; Li, S.; Wang, X. Source mask optimization using the covariance matrix adaptation evolution strategy. *Opt. Express* **2020**, *28*, 33371–33389. [[CrossRef](#)]
41. Kaveh, A.; Javadi, S.M.; Moghanni, R.M. Reliability Analysis via an Optimal Covariance Matrix Adaptation Evolution Strategy: Emphasis on Applications in Civil Engineering. *Period. Polytech. Civ. Eng.* **2020**, *64*, 579–588. [[CrossRef](#)]
42. Hastie, T.; Tibshirani, R.; Friedman, J.H.; Friedman, J.H. *The Elements of Statistical Learning-Data Mining, Inference, and Prediction*, 2nd ed.; Springer: New York, NY, USA, 2009.
43. Zou, H.; Hastie, T. Regularization and Variable Selection via the Elastic Net. *J. R. Stat. Soc.* **2005**, *67*, 301–320. [[CrossRef](#)]
44. Guo, P.; Cheng, W.; Wang, Y. Hybrid evolutionary algorithm with extreme machine learning fitness function evaluation for two-stage capacitated facility location problems. *Expert Syst. Appl.* **2017**, *71*, 57–68. [[CrossRef](#)]
45. Saporetti, C.M.; Duarte, G.R.; Fonseca, T.L.; Da Fonseca, L.G.; Pereira, E. Extreme Learning Machine combined with a Differential Evolution algorithm for lithology identification. *RITA* **2018**, *25*, 43–56. [[CrossRef](#)]
46. Huang, G.; Huang, G.-B.; Song, S.; You, K. Trends in extreme learning machines: A review. *Neural Netw.* **2015**, *61* (Suppl. C), 32–48. [[CrossRef](#)]
47. Suykens, J.A.K.; Vandewalle, J. Least Squares Support Vector Machine Classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300. [[CrossRef](#)]
48. Chang, C.C.; Lin, C.J. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–27. [[CrossRef](#)]
49. Karthikeyan, M.; Vyas, R. Machine learning methods in chemoinformatics for drug discovery. In *Practical Chemoinformatics*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 133–194. [[CrossRef](#)]
50. Gunn, S.R. Support Vector Machines for Classification and Regression. *ISIS Tech. Rep.* **1998**, *14*, 5–16.
51. Kargar, K.; Samadianfard, S.; Parsa, J.; Nabipour, N.; Shamshirband, S.; Mosavi, A.; Chau, K.-W. Estimating longitudinal dispersion coefficient in natural streams using empirical models and machine learning algorithms. *Eng. Appl. Comput. Fluid Mech.* **2020**, *14*, 311–322. [[CrossRef](#)]
52. Broomhead, D.S.; Lowe, D. Radial basis functions, multi-variable functional interpolation and adaptive networks. In *Royal Signals; Radar Establishment Malvern: Malvern, UK*, 1988.
53. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; KDD'16. ACM: New York, NY, USA, 2016; pp. 785–794.
54. Chen, T.; He, T. Higgs boson discovery with boosted trees. In *NIPS 2014 Workshop on High-Energy Physics and Machine Learning*; PMLR: London, UK, 2015; pp. 69–80.
55. Ou, X.; Morris, J.; Martin, E. Gaussian Process Regression for Batch Process Modelling. *IFAC Proc. Vol.* **2004**, *37*, 817–822. [[CrossRef](#)]
56. Wang, B.; Chen, T. Gaussian process regression with multiple response variables. *Chemom. Intell. Lab. Syst.* **2015**, *142*, 159–165. [[CrossRef](#)]
57. Schulz, E.; Speekenbrink, M.; Krause, A. A Tutorial on Gaussian Process Regression: Modelling, Exploring, and Exploiting Functions. *J. Math. Psychol.* **2018**, *85*, 1–16. [[CrossRef](#)]
58. Kumar, R. The generalized modified Bessel function and its connection with Voigt line profile and Humbert functions. *Adv. Appl. Math.* **2020**, *114*, 101986. [[CrossRef](#)]

59. Hansen, N.; Ostermeier, A. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evol. Comput.* **2001**, *9*, 159–195. [[CrossRef](#)] [[PubMed](#)]
60. Beyer, H.G.; Schwefel, H.P. Evolution Strategies—a Comprehensive Introduction. *Nat. Comput.* **2002**, *1*, 3–52. [[CrossRef](#)]
61. Li, Z.; Zhang, Q.; Lin, X.; Zhen, H.-L. Fast Covariance Matrix Adaptation for Large-Scale Black-Box Optimization. *IEEE Trans. Cybern.* **2018**, *50*, 2073–2083. [[CrossRef](#)]
62. Willmott, C.J. ON the Validation of Models. *Phys. Geogr.* **1981**, *2*, 184–194. [[CrossRef](#)]
63. Nash, J.E.; Sutcliffe, J.V. River flow forecasting through conceptual models part I—A discussion of principles. *J. Hydrol.* **1970**, *10*, 282–290. [[CrossRef](#)]
64. Gupta, H.V.; Kling, H.; Yilmaz, K.K.; Martinez, G.F. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *J. Hydrol.* **2009**, *377*, 80–91. [[CrossRef](#)]
65. Kisi, O.; Shiri, J.; Karimi, S.; Adnan, R.M. Three different adaptive neuro fuzzy computing techniques for forecasting long-period daily streamflows. In *Big Data in Engineering Applications*; Springer: Singapore, 2018; pp. 303–321. [[CrossRef](#)]
66. Alizamir, M.; Kisi, O.; Adnan, R.M.; Kuriqi, A. Modelling reference evapotranspiration by combining neuro-fuzzy and evolutionary strategies. *Acta Geophys.* **2020**, *68*, 1113–1126. [[CrossRef](#)]
67. Yuan, X.; Chen, C.; Lei, X.; Yuan, Y.; Muhammad Adnan, R. Monthly runoff forecasting based on LSTM–ALO model. *Stoch. Environ. Res. Risk Assess.* **2018**, *32*, 2199–2212. [[CrossRef](#)]
68. Sattar, A.M.; Gharabaghi, B. Gene expression models for prediction of longitudinal dispersion coefficient in streams. *J. Hydrol.* **2015**, *524*, 587–596. [[CrossRef](#)]
69. Sudheer, K.P.; Gosain, A.K.; Ramasastri, K.S. A data-driven algorithm for constructing artificial neural network rainfall-runoff models. *Hydrol. Process.* **2002**, *16*, 1325–1330. [[CrossRef](#)]
70. Kisi, O. Constructing neural network sediment estimation models using a data-driven algorithm. *Math. Comput. Simul.* **2008**, *79*, 94–103. [[CrossRef](#)]
71. Wu, C.; Chau, K. Prediction of rainfall time series using modular soft computing methods. *Eng. Appl. Artif. Intell.* **2013**, *26*, 997–1007. [[CrossRef](#)]
72. Adnan, R.M.; Liang, Z.; Parmar, K.S.; Soni, K.; Kisi, O. Modeling monthly streamflow in mountainous basin by MARS, GMDH-NN and DENFIS using hydroclimatic data. *Neural Comput. Appl.* **2021**, *33*, 2853–2871. [[CrossRef](#)]