



Article

Incorporating Recursive Feature Elimination and Decomposed Ensemble Modeling for Monthly Runoff Prediction

Wei Ma ^{1,*} , Xiao Zhang ^{1,*} , Yu Shen ², Jiancang Xie ^{1,*}, Ganggang Zuo ¹, Xu Zhang ¹ and Tao Jin ³

¹ State Key Laboratory of Eco-Hydraulics in Northwest Arid Region, Xi'an University of Technology, No. 5 Jinhua South Road, Beilin District, Xi'an 710048, China; mawei0502@outlook.com (W.M.); zuoganggang@163.com (G.Z.); 1220410011@stu.xaut.edu.cn (X.Z.)

² Shanxi Provincial Hydrology and Water Resources Survey Station, Taiyuan 030001, China; adamuniversity@sina.com

³ College of Civil Engineering, Hefei University of Technology, Hefei 230001, China; jintao20200515@163.com

* Correspondence: zhangxiao1989@xaut.edu.cn (X.Z.); jxie@xaut.edu.cn (J.X.)

Abstract: Monthly runoff prediction is crucial for water resource allocation and flood prevention. Many existing methods use identical deep learning networks to understand monthly runoff patterns, neglecting the importance of predictor selection. To enhance predictive accuracy and reliability, this study proposes an RFECV–SSA–LSTM forecasting approach. It iteratively eliminates predictors derived from SSA decomposition and PACF using recursive feature elimination and cross-validation (RFECV) to identify the most relevant subset for predicting the target flow. LSTM modeling is then used to forecast flows 1–7 months into the future. Furthermore, the RFECV–SSA framework complements any machine-learning-based runoff prediction method. To demonstrate the method's reliability and effectiveness, its outputs are compared across three scenarios: direct LSTM, MIR–LSTM, and RFECV–LSTM, using monthly runoff historical data from Yangxian and Hanzhong hydrological stations in the Hanjiang River Basin, China. The results show that the RFECV–LSTM method is more robust and efficient than the direct LSTM and MIR–LSTM counterparts, with the smallest number of outliers for NSE, NRMSE, and PPTS under all forecasting scenarios. The MIR–LSTM approach exhibits the worst performance, indicating that single-metric-based feature selection may eliminate valuable information. The SSA time–frequency decomposition is superior, with NSE values remaining stably around 0.95 under all scenarios. The NSE value of the RFECV–SSA–LSTM method is greater than 0.95 under almost all forecasting scenarios, outperforming other benchmark models. Therefore, the RFECV–SSA–LSTM method is effective for forecasting highly nonlinear runoff series, exhibiting high accuracy and generalization ability.

Keywords: monthly streamflow forecasting; time–frequency decomposition; long short-term memory neural network; recursive feature elimination; cross-validation



Citation: Ma, W.; Zhang, X.; Shen, Y.; Xie, J.; Zuo, G.; Zhang, X.; Jin, T. Incorporating Recursive Feature Elimination and Decomposed Ensemble Modeling for Monthly Runoff Prediction. *Water* **2024**, *16*, 3102. <https://doi.org/10.3390/w16213102>

Academic Editor: Alexander Shiklomanov

Received: 6 September 2024

Revised: 21 October 2024

Accepted: 26 October 2024

Published: 29 October 2024



Copyright: © 2024 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

Accurate streamflow forecasting is of great significance for water resource management and flood control dispatching. However, there exist nonlinear, periodic, and trend characteristics in streamflow series which have made streamflow forecasting a great challenge for a long time [1]. These characteristics increase the complexity of streamflow forecasting, making it challenging for any model to achieve high prediction accuracy and stability unless it can simultaneously capture these multifaceted factors [2]. Extensive experiments on streamflow forecasting have shown that physics-based prediction models driven by hydrological processes require complex mathematical models, fluid physics simulations, and more hydro-meteorological data. Such forecasting models have problems like structural bias and a huge workload, resulting in low forecast accuracy and poor practicality. On the other hand, the currently prevalent data-driven models can capture

the latent information in streamflow series better, with fewer required forecasting factors, easier data acquisition, and higher prediction accuracy [3,4].

Quite a few experiments have thus far adopted moving average (MA) models [3], autoregressive (AR) models [5], autoregressive moving average (ARMA) models [6], and seasonal autoregressive integrated moving average (SARIMA) models [7], alongside other data-driven time series models to simulate and forecast streamflow [2,8,9]. Although these models are widely applied to time series analysis, they are limited to handling linear relationships and are unable to capture the complex nonlinear characteristics and patterns underlying hydrological processes [10]. Due to the nonlinear, trend, and periodic components commonly present in streamflow series, using these linear models for prediction may lead to performance limitations and lower forecast accuracy. Moreover, these models perform poorly when dealing with extreme climatic events or nonlinear dynamic changes, further restricting their applicability in complex hydrological environments [11].

With the development of artificial intelligence, intelligent algorithms have been increasingly applied to streamflow rolling forecasting, such as hierarchical clustering analysis [12], principal component analysis [13], artificial neural networks [14], decision trees [15], and support vector regression [16]. Compared with traditional regression forecasting models, these machine learning methods have greatly improved prediction accuracy and efficiency [17]. However, the capability of such models to handle streamflow series is limited, and they tend to deliver a poor convergence performance, they belong to shallow learning, and they are prone to getting stuck in local optima and to overfitting [18]. The long short-term memory (LSTM) network is a special type of neural network proposed on the basis of RNN. It is designed mainly to address the vanishing and exploding gradient problems of neural networks during backpropagation. Compared with RNN, LSTM has greater advantages in long sequence prediction [19]. Since LSTM models can better capture the temporal correlations in sequences, approaches like removing or reducing the number of neurons and increasing the regularization penalty can help control the overfitting of LSTM models. Also, LSTM models have faster convergence and can more easily capture the regularity of nonstationary sequences [20]. Therefore, LSTM models have been widely applied to long-term streamflow sequence forecasting [21].

Streamflow series are affected by trend, seasonal, periodic, and error components as well as by irregular noise disturbances [22,23]. A single machine learning model cannot accurately predict highly nonlinear sequences with high-noise components [24]. Therefore, it is necessary to preprocess the original sequence data, extract multiscale features that are simpler than the original signal, and establish a decomposition–ensemble model that combines time series preprocessing and machine learning to forecast streamflow [25]. Increasing research has shown that preprocessing data through appropriate time–frequency decomposition techniques can further extract useful information hidden in time series, greatly improving the prediction accuracy of data-driven models [26]. Discrete wavelet transform (DWT) [27], empirical mode decomposition (EMD) [28], ensemble empirical mode decomposition (EEMD) [29], complementary ensemble empirical mode decomposition with adaptive noise (CEEMDAN) [30], singular spectrum analysis (SSA) [31], and variational mode decomposition (VMD) [32] are commonly used time–frequency decomposition methods. Among these decomposition methods, discrete wavelet transform (DWT) has good time–frequency aggregation characteristics, but its decomposition effect depends on the choice of basis functions and has poor adaptability [33]. Although time–frequency decomposition algorithms can extract multiscale features of streamflow series, selecting the feature subsets with the greatest impact on the forecast target from the high-dimensional feature matrix is still key to improving the predictive performance. References [34,35] have found that using recursive feature elimination and cross-validation (RFECV) can screen out effective wetland classification features and achieve high-precision wetland mapping, providing technical support to the ecological health of wetlands. Chen [36] used RFECV to achieve the best results for the radiomics model, with radiomics outperforming the accuracy of conventional analysis by 67% for STS grading. Reference [37] used RFECV, Pearson's

correlation coefficient, and mutual information to identify the molecular descriptors most relevant to the kinematic viscosity of natural ester insulating oil, enabling efficient and accurate prediction of kinematic viscosity. Therefore, solely relying on time–frequency decomposition to extract multi-scale features is not enough. It is necessary to adopt appropriate feature selection methods to choose feature subsets that have more significant effects on the prediction target. This plays a certain role in improving prediction performance. This paper proposes an RFECV–LSTM model scheme for monthly streamflow forecasting. First, the original streamflow series is decomposed into multiple IMFs by using time–frequency decomposition algorithms (DWT, EMD, EEMD, CEEMDAN, VMD, SSA). Then, the optimal lag for each subsignal is obtained using the partial autocorrelation function (PACF). Based on the maximum lag operator, the feature factor input matrix is generated. To demonstrate the superiority of the RFECV–LSTM forecasting scheme, LSTM models without feature selection (direct LSTM), with mutual information (MIR) feature selection, and with RFECV feature selection are compared. During training, Bayesian optimization is used to optimize the hyperparameters of the LSTM model, resulting in the optimal RFECV–LSTM prediction neural network model. To validate the excellent streamflow forecasting performance of RFECV–LSTM, monthly streamflow data from Yangxian and Hanzhong hydrological stations in China’s Hanjiang River Basin are used for experiments.

2. Study Area and Data

The Hanjiang River Basin extends over an area of 159,000 km² and has a length of 1532 km, making it the largest tributary of the Yangtze River. It originates from the Zhong Mountain in Ningqiang County, between the Qinling and Micang Mountains in the southwestern Shaanxi Province. After flowing out of the Danjiangkou Reservoir in the southeast, it converges with the Yangtze River. As shown in Figure 1, the Hanzhong and Yangxian hydrological stations in the upper reaches of the Hanjiang River Basin are selected as the research sites. The Hanjiang River Basin belongs to the subtropical monsoon climate zone, with abundant but unevenly distributed annual runoff and large interannual variations. Accurately predicting runoff in this basin presents considerable challenges.

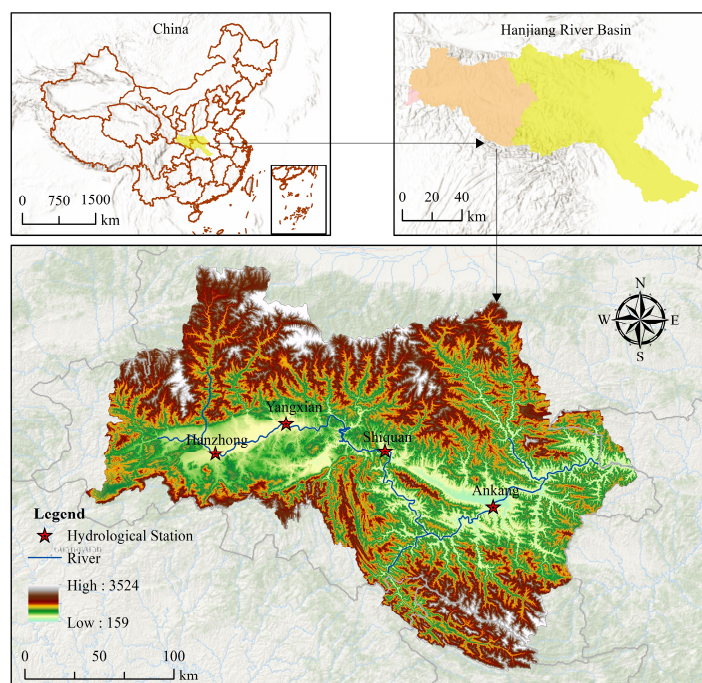


Figure 1. Distribution of hydrological stations in the upper reaches of the Hanjiang River Basin, China.

Monthly streamflow data from January 1967 to December 2020 from the Yangxian hydrological station ($107^{\circ}54'$ E, $33^{\circ}22'$ N) and from May 2000 to December 2022 from the Hanzhong hydrological station ($107^{\circ}02'$ E, $33^{\circ}05'$ N) in the Hanjiang River Basin were used to evaluate all predictive models in this study. The monthly streamflow time series data were calculated from daily monitored streamflow data collected by the Shaanxi Hydrological Information Center and Water Resources Survey Bureau. The first 80% of the data from each monthly streamflow sequence was used as the training set, and the remaining 20% was used as the validation and test sets. The original streamflow sequences are shown in Figure 2.

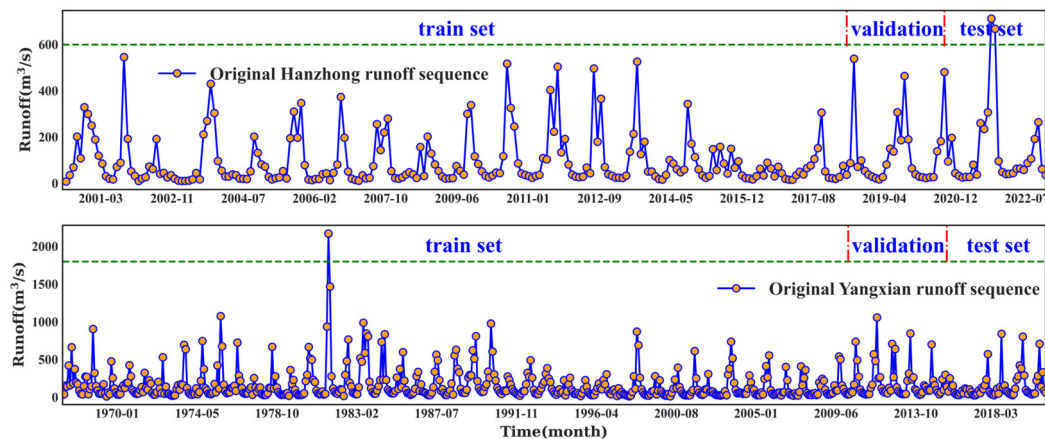


Figure 2. Monthly runoff at the Yangxian and Hanzhong hydrological stations.

3. Research Methods

3.1. Decomposed Ensemble Model for Streamflow Forecasting

In this study, to validate the stability, generalization capability, credibility, and computational efficiency of the decomposed ensemble forecasting model, several decomposed ensemble models were compared, including EMD–LSTM, EEMD–LSTM, CEEMDAN–LSTM, DWT–LSTM, SSA–LSTM, and VMD–LSTM. The optimal lag time operator for each decomposed subsignal was determined based on PACF to generate feature factor samples. The forecasting targets of monthly streamflow were set to 1, 3, 5, and 7 months into the future. Since the feature factors were directly used to train the LSTM model, this scheme was termed direct LSTM. However, given the susceptibility of LSTM models to overfitting and convergence issues, feature selection techniques were introduced to mitigate these challenges.

Specifically, the feature factors were selected using mutual information (MIR) and recursive feature elimination (RFECV), and the selected features were used as inputs to the LSTM model, while the original monthly streamflow and its forecasts 1, 3, 5, and 7 months into the future were used as the output. These models were termed MIR–LSTM and RFECV–LSTM, respectively. By applying feature selection, the dimensionality of input features was reduced, which, in turn, lowered the model's complexity and mitigated the risk of overfitting in the LSTM model. Additionally, this approach reduced the computational resources required and enhanced both the computational efficiency and model convergence. Furthermore, regularization and Bayesian optimization for hyperparameter tuning were employed to enhance the robustness of the LSTM model.

To validate the performance of the aforementioned time–frequency decomposition methods for streamflow forecasting, the EMD–LSTM, EEMD–LSTM, CEEMDAN–LSTM, DWT–LSTM, SSA–LSTM, and VMD–LSTM decomposed ensemble schemes were compared to identify the method best suited to streamflow sequences. Further, to evaluate the effects of the two feature selection methods on model performance, feature subsets selected by mutual information (MIR) and recursive feature elimination (RFECV) were compared as LSTM inputs against the use of all predictive factors directly, i.e., MIR–LSTM,

RFECV–LSTM, and direct LSTM. In this study’s experiments, forecasting 1, 3, 5, and 7 months into the future was performed for all the decomposed ensemble models.

The number of modes K for VMD was determined by observing the center frequency spectrum. Extensive experiments have shown that the secondary penalty term (α) and tolerance (ϵ) for VMD are optimal when set to 2000 and 1×10^{-9} , respectively [26]. To obtain uncorrelated VMD subsignals with low noise, the noise tolerance τ was set to 0 [38]. The number of decomposition levels L for SSA was also determined by inspecting the center frequency spectrum. Compared to EMD, EEMD introduces two additional parameters—the ensemble number (M) and the noise intensity (ϵ)—which improve the endpoint effect and mode mixing problems of EMD. Based on previous studies, M and ϵ were set to 100 and 0.2, respectively [39]. Relative to EEMD, CEEMDAN uses amplitude-equivalent but phase-opposite noise ensembles during signal reconstruction to compute a unique residue, effectively reducing reconstruction errors and avoiding mode mixing [40].

Discrete wavelet transform (DWT) decomposes the original streamflow series into detailed subsequences representing high-frequency components of length L , and approximation subsequences representing low-frequency trends. There is no unified standard or theory for selecting the wavelet mother function and decomposition levels for DWT. In this study, the mother wavelets tested were: haar, db2, bior3.3, db5, coif3, db10, db15, db20, db25, db30, db35, db40, and db45. The optimal mother wavelet was identified through experiments, and the decomposition levels (L) were set to 1, 2, and 3 under each mother wavelet to determine the optimal level.

The specific steps for the decomposed ensemble model for streamflow forecasting in this study can be summarized as shown in Figure 3.

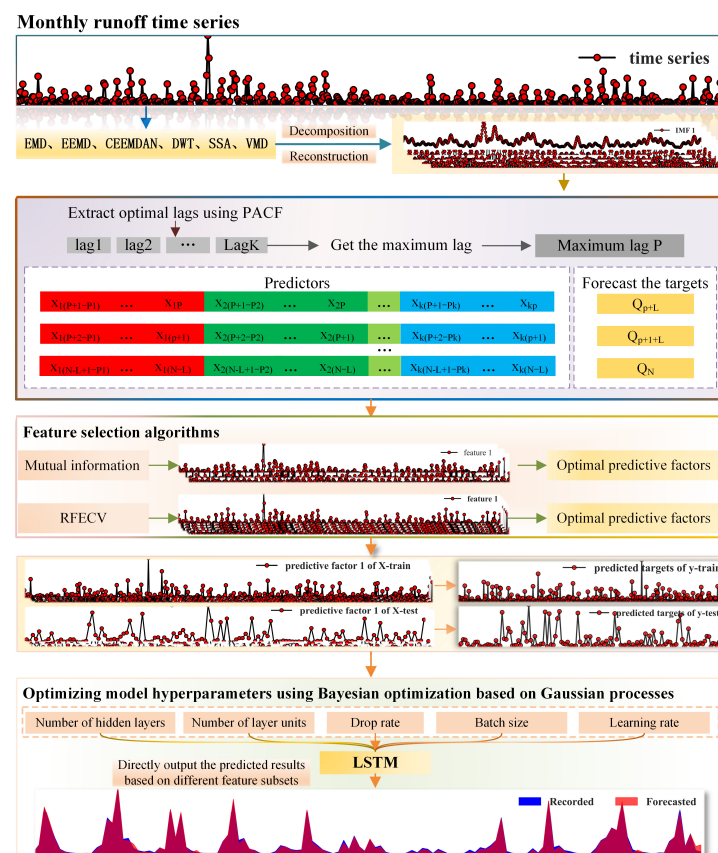


Figure 3. Flowchart of the SSA–LSTM implementation process.

Step 1: collect monthly streamflow time series data $Q(t)$ ($t = 1, 2, \dots, N$, where N is the data length).

Step 2: decompose the entire data sequence into K intrinsic mode function (IMF) components using the EMD, EEMD, CEEMDAN, DWT, SSA, or VMD decomposition algorithms. The optimal decomposition levels (K) for SSA and VMD are determined by inspecting the center frequency spectrum of the last subsignal.

Step 3: plot the partial autocorrelation (PACF) coefficients for each subsignal and identify lags exceeding the 95% confidence interval as predictor variables. These lags will be used as predictors to forecast the m step future streamflow, denoted as $Q(t + m)$.

Step 4: based on the predictor variables and predictand identified in Step 3, construct the input matrix for the training of the direct LSTM model using the training subsignals generated in Step 2.

Step 5: based on the input matrix constructed in Step 4, perform feature selection on the predictor variables to obtain two sets of feature subsets, one based on mutual information (MIR) and one based on recursive feature elimination with cross-validation (RFECV). These feature subsets form the input matrices for MIR-LSTM and RFECV-LSTM, respectively.

Step 6: divide the sample matrices generated in Steps 4 and 5 into training, validation, and test sets as follows: 80% of the samples should be kept for training, the next 10% for validation, and the remaining 10% for testing.

Step 7: normalize each subsequence in the training, validation, and test sets using min-max normalization ($X' = \frac{2(X - \min(X))}{\max(X) - \min(X)}$) to rescale the data to a range of $[-1, 1]$. This addresses differences in scale across features, accelerates model convergence, and improves predictive performance and generalization capability.

Step 8: using Gaussian process-based Bayesian optimization, perform hyperparameter tuning for each LSTM model (MIR-LSTM, RFECV-LSTM, direct LSTM) based on the corresponding sample matrices obtained in Step 7. This results in optimized LSTM models with the best set of hyperparameters.

Step 9: using the optimized LSTM models from Step 8, make predictions on the validation set and test set. Evaluate and compare the model performance of MIR-LSTM, RFECV-LSTM, and direct LSTM based on relevant metrics like NSE, NRMSE, and PPTS. Analyze the results to determine the best performing model.

3.2. Model Evaluation Criteria

Three error analysis criteria were adopted to evaluate the predictive performance of the proposed and benchmark models: normalized root mean square error (NRMSE), Nash efficiency coefficient (NSE) [41], and peak percentage of threshold statistics (PPTS) [42]. NRMSE directly reflects the normalized difference between predictions and observations, with lower values indicating a better model performance. NSE is widely used in hydrological modeling, with values closer to 1 denoting higher prediction accuracy. PPTS evaluates peak flow predictive capability by comparing errors in peak discharge predictions.

$$NSE = 1 - \frac{\sum_{t=1}^N (x(t) - \hat{x}(t))^2}{\sum_{t=1}^N (x(t) - \bar{x}(t))^2} \quad (1)$$

$$NRMSE = \frac{\sqrt{\frac{\sum_{t=1}^N (x(t) - \hat{x}(t))^2}{N}}}{\frac{\sum_{t=1}^N x(t)}{N}} \quad (2)$$

$$PPTS = \frac{100}{\gamma} \frac{1}{N} \sum_{t=1}^G \left| \frac{x(t) - \hat{x}(t)}{x(t)} \right| \cdot 100 \quad (3)$$

where $x(t)$ is the original streamflow sequence, $\hat{x}(t)$ is the predicted streamflow sequence, $\bar{x}(t)$ is the mean of the original streamflow sequence, N is the length of the streamflow

sequence, γ is the percentage of data chosen from the data sequence in descending order, i.e., the threshold level, and G is the number of data points exceeding the threshold level.

3.3. Singular Spectrum Analysis (SSA) for Data Decomposition

Singular spectrum analysis (SSA) decomposes the original time series into a sum of component series. Each decomposed component mainly consists of trend, periodic, noise, and other constituents. This separation of the time series into interpretable components enables noise reduction by reconstructing the original series without the noise components, thereby improving forecasting accuracy. SSA involves four main steps:

Embedding: the analysis object of SSA is a one-dimensional time series of finite length N . First, an appropriate window length L was selected to map the original time series into a trajectory matrix by lagged copies of the series:

$$\begin{pmatrix} X_1 & X_2 & \cdots & X_{N-L+1} \\ X_2 & X_3 & \cdots & X_{N-L+2} \\ \vdots & \vdots & \cdots & \vdots \\ X_L & X_{L+1} & \cdots & X_N \end{pmatrix} \tag{4}$$

Generally, $L < N/2$ is taken. Let $K = N - L + 1$, then the trajectory matrix X is an $L \times K$ matrix.

Decomposition: the trajectory matrix X was decomposed by performing singular value decomposition (SVD). Specifically, SVD decomposes the trajectory matrix into the following form:

$$X = U\Sigma V^T \tag{5}$$

U is the left matrix, Σ is a singular value with values only on the diagonal and other elements are zero, V is the right matrix, U and V are unit orthogonal matrices which meet the requirements of $UU^T = I$ and $VV^T = I$, respectively.

Since it is difficult to directly decompose the trajectory matrix, the covariance matrix of the trajectory matrix was first computed:

$$S = XX^T \tag{6}$$

Then, the eigenvalues of S were obtained by eigenvalue decomposition $\lambda_1 > \lambda_2 > \cdots > \lambda_L \geq 0$ and by the corresponding eigenvectors U_1, U_2, \dots, U_L . At this point, $U = [U_1, U_2, \dots, U_L]$, $\sqrt{\lambda_1} > \sqrt{\lambda_2} > \cdots > \sqrt{\lambda_L} \geq 0$ was the original sequence.

$$X = \sum_{m=1}^L \sqrt{\lambda_m} U_m V_m^T \tag{7}$$

$$m = 1, 2, \dots, L \tag{8}$$

Here, the eigenvector U_i corresponding to λ_i reflects the evolution of the time series and is called the time empirical orthogonal function.

Grouping: all components were divided into nonoverlapping groups, representing different trend components. The main components were then selected for reconstruction to obtain the reconstructed sequence.

$$X = X_{L1} + \cdots + X_{LC} \tag{9}$$

$$X_L = \sum_{m \in L} \sqrt{\lambda_m} U_m V_m^T = \left(\sum_{m \in L} U_m U_m^T \right) X \tag{10}$$

Reconstruction: the projection of the hysteresis sequence X_i on U_m was calculated:

$$a_i^m = X_i U_m = \sum_{j=1}^L x_{i+j} U_{m,j} \quad 0 \leq i \leq N - L \tag{11}$$

where X_i represents the i th column of the trajectory matrix X and a_i^m is the weight of the time evolution type reflected in the $X_i, X_{i+2}, \dots, X_{i+L}$ period of the original sequence, called the time principal component (TPC). The matrix formed by a_i^m is a right matrix without normalization, that is, $\sqrt{\lambda_m} V_m$.

The time series was then reconstructed using the time empirical orthogonal functions and principal components through the following steps:

$$X_i = \begin{cases} \frac{1}{L} \sum_{j=1}^i a_{i-j}^k U_{K,j}, & 1 \leq i \leq L - 1 \\ \frac{1}{L} \sum_{j=1}^i a_{i-j}^k U_{K,j}, & 1 \leq i \leq N - L + 1 \\ \frac{1}{N - i + 1} \sum_{j=i-N+L}^L a_{i-j}^k E_{K,j}, & N - L + 2 \leq i \leq N \end{cases} \tag{12}$$

The sum of all the reconstructed sequences should be equal to the original sequence:

$$X_i = \sum_{k=1}^L X_i^k, i = 1, 2, \dots, N \tag{13}$$

Typically, with SSA, the goal is to extract the dominant components of the original sequence, as in the case of denoising. In this case, only the first $K(K \leq L)$ components that contributed the most variance needed to be retained for reconstructing the original sequence, based on the singular values.

3.4. LSTM for Time Series Forecasting

Long short-term memory (LSTM) networks are a special type of recurrent neural network (RNN) that can capture long-term dependencies in sequence data and mitigate vanishing and exploding gradient problems to some extent [43]. LSTMs employ three main gating structures: forget gates, input gates, and output gates to protect and control the cell state [44]. A schematic of an LSTM network is shown in Figure 4.

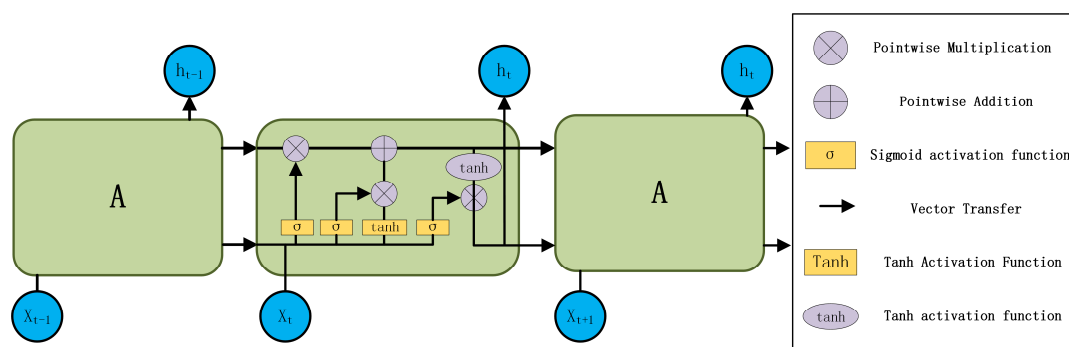


Figure 4. Schematic diagram of a long short-term memory network (A: Represents the LSTM unit, which includes various operations such as input modulation, gating mechanisms, and state updates. It is responsible for maintaining and updating the hidden state and cell state across time steps).

The forgetting gate determines what information is discarded from the cell state. Specifically, the forgetting gate takes as input the previous hidden state h_{t-1} and current input X_t . It applies a sigmoid activation to produce a vector f_t with values between 0 and 1,

where 1 represents completely retaining and 0 represents completely forgetting. This vector f_t is then multiplied elementwise with the previous cell state C_{t-1} to selectively discard information. The calculation formula is presented below.

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \quad (14)$$

The input gate determines what new information is stored in the cell state. First, a sigmoid layer called the input gate layer decides what values should be updated, generating the vector. Second, a tanh layer creates a new candidate value vector \tilde{C}_t . This is added to the cell state. The calculation formulas are presented below.

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \quad (15)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, X_t] + b_C) \quad (16)$$

The cell state refers to the process of updating from the old cell state C_{t-1} to the new C_t . First, the old cell state C_{t-1} is multiplied by f_t , discarding information that needs to be forgotten. Then, the product of i_t and \tilde{C}_t is added. \tilde{C}_t represents a new candidate value that determines how much each cell state should be updated. The calculation formula is presented below.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C} \quad (17)$$

The output gate determines the output value, which is the final filtered version. First, a sigmoid layer decides which parts of the cell state O_t should be output. Next, the cell state C_t passes through a tanh layer to obtain values between -1 and 1 . This is multiplied by the output from the sigmoid gate to obtain, ultimately, the cell output value h_t . The calculation formulas are presented below.

$$O_t = \sigma(W_o [h_{t-1}, X_t] + b_o) \quad (18)$$

$$h_t = O_t \cdot \tanh(C_t) \quad (19)$$

4. Case Study

4.1. Experimental Setup

The partial autocorrelation function (PACF) determined the optimal lag terms for the intrinsic mode functions (IMFs) decomposed from the monthly flows. Two feature selection methods were then applied to the generated feature subsets—mutual information (MIR) and recursive feature elimination with cross-validation (RFECV)—forming two forecasting approaches, namely MIR–LSTM and RFECV–LSTM, with their respective predictors. Along with the direct LSTM without feature selection, three predictor sets were generated. All three methods of MIR–LSTM, RFECV–LSTM, and direct LSTM targeted monthly flows forecasted 1, 3, 5, and 7 months into the future.

First, the original data series were decomposed using EMD, EEMD, CEEMDAN, SSA, DWT, and VMD time–frequency decomposition methods to reduce noise in the runoff sequences and capture trends. PACF was then used to determine the optimal lag terms for each IMF component, generating sample matrices with prediction targets of 1, 3, 5, and 7 months into the future. Finally, MIR and RFECV were applied to perform feature selection on the matrices composed of the forecast factors and prediction targets for the 1-, 3-, 5-, and 7-month lead times. The resulting selected forecast factors were combined with the unfiltered forecast factors to form the input sample matrices for three LSTM prediction frameworks: MIR–LSTM, RFECV–LSTM, and direct LSTM. In summary, the sample matrices for these three decomposition–integration prediction schemes were constructed through feature selection using MIR and RFECV on the lead time forecast factors.

The sample matrices were split into three parts: training set, validation set, and test set. The LSTM was trained on the training set. The hyperparameters of the LSTM were tuned by evaluating model performance on the validation set using metrics like MSE. The

tuned LSTM was then evaluated on the test set. Bayesian optimization was used to select the optimal hyperparameters for the LSTM model, including the learning rate, the number of hidden layers, the number of hidden units, and the dropout rate. The search strategy for the learning rate was logarithmic sampling in the range of $[10^{-4}, 10^{-1}]$. The search strategy for the number of hidden layers was [1:1:3], indicating the start, step size, and end of the search range. The search range for the number of hidden units was [16:16:64]. The dropout rate was searched in the range of $[0, 0.05, 0.5]$. The activation function was set to ReLU. The optimization method was Adam. The batch size was set to 32.

4.2. PACF for Optimal Lag Selection

The correlations among the modal components obtained through singular spectrum analysis (SSA) of the monthly streamflow data should be analyzed prior to the overall forecasting. The partial autocorrelation function (PACF) can determine the optimal lag operators among the subsignal components. The PACF results can guide selection of input feature vectors for the LSTM prediction model to capture different oscillation modes in the streamflow across periods. Careful input feature engineering using PACF enables effective LSTM modeling of the streamflow time series [4].

If the lagged autocorrelation coefficient of the partial autocorrelation function (PACF) exceeded the upper and lower bounds of the 95% confidence interval ($[-\frac{1.96}{\sqrt{n}}, \frac{1.96}{\sqrt{n}}]$), then that lagged term was included as one of the input variables and incorporated into the feature vector for the prediction model. If all the PACF coefficients were within the 95% confidence interval, then the previous value was used as the input variable. The PACF is described as follows. For a streamflow sequence, the lagged autocovariance of order k (where $k = 0$ represents the variance), denoted as γ_k , is calculated as:

$$\hat{\gamma}_k = \frac{1}{n} \sum_{t=1}^{n-k} (x_t - \bar{x})(x_{t+k} - \bar{x}), (k = 1, 2, \dots, M) \quad (20)$$

Here, \bar{x} denotes the mean of the streamflow sequence, $M = n/4$ is the maximum lag order of the partial autocorrelation function (PACF), and k represents the lag length of the PACF which can be expressed as:

$$\hat{\rho}_k = \hat{\gamma}_k / \hat{\gamma}_0 \quad (21)$$

Based on the autocovariance, the PACF of lag order k ($k = 1, 2, \dots, M$) denoted as f_{kk} can be calculated as:

$$\left. \begin{aligned} \hat{f}_{11} &= \hat{\rho}_1 \\ \hat{f}_{k+1,k+1} &= (\hat{\rho}_{k+1} - \sum_{j=1}^k \hat{\rho}_{k+1-j} \hat{f}_{kj}) / (1 - \sum_{j=1}^k \hat{\rho}_j \hat{f}_{k,j}) \\ \hat{f}_{k+1} &= \hat{f}_{kj} - \hat{f}_{k+1,k+1} \cdot \hat{f}_{k,k-j+1} \end{aligned} \right\} (j = 1, 2, \dots, k) \quad (22)$$

4.3. Data Normalization

Data normalization was performed in the experiments, as it can effectively reduce the difficulty of optimizing the objective function, improve the accuracy, stability, and generalization ability of the model, and accelerate model convergence. All the prediction factors and prediction targets were normalized to the range of $[-1, 1]$. The normalization equation is as follows:

$$x' = 2 \times \frac{x - x_{\min}}{x_{\max} - x_{\min}} - 1 \quad (23)$$

where x and x' are the original and normalized values, respectively, and x_{\min} and x_{\max} are the minimum and maximum values of the original sequence, respectively. In addition, to ensure the validation and test sets have the same distribution as the training set, the same maximum (x_{\max}) and minimum (x_{\min}) values from the training samples were used to normalize the validation and test set samples. Since the decomposition results of VMD,

EMD, EEMD, CEEMDAN, DWT, and SSA contain negative values, the data samples were scaled to the range of $[-1, 1]$.

4.4. Bayesian Optimization for Hyperparameter Tuning

The choice of model hyperparameters significantly impacts model performance. To obtain the best predictive performance of LSTM and enhance its generalization capability, hyperparameter tuning is required. During model training, the Bayesian optimization algorithm (BOA) was utilized to optimize the hyperparameters. Compared to commonly used optimization algorithms like grid search, random search, and halving grid search, BOA can accelerate the convergence speed and enable more efficient hyperparameter optimization [45]. Assuming the objective function is f_t , and taking a set of hyperparameters as input and outputting five-fold cross-validation results, the goal is to find the minimum value of f_t at the hyperparameter sample points. The Bayesian optimization process can be briefly described as follows:

$$x^* = \operatorname{argmin}_{x \in X} f(x) \quad (24)$$

Here, X represents the hyperparameter space, with $x^* \in X$ being the set of hyperparameters that generate the minimal value.

The surrogate model and acquisition function are two key steps in Bayesian optimization. In each iteration, the Bayesian optimization process constructs a surrogate function of the objective using a Gaussian process (GP), then finds the optimal hyperparameters for the surrogate via the acquisition function and uses those hyperparameters to sequentially refine the surrogate model. The main steps of BOA are as follows [46]:

Step 1: define the hyperparameter space and construct the five-fold cross-validation objective function.

Step 2: define the initial sample points using random sampling and the maximum number of iterations.

Step 3: randomly select sample points within the defined hyperparameter space to initialize the surrogate model.

Step 4: construct a surrogate model of the objective function.

Step 5: acquire the hyperparameters with optimal performance for the surrogate model, apply these points to the objective function to generate results.

Step 6: refine the surrogate model by incorporating the new results.

Step 7: repeat steps 4–6 until the maximum number of iterations is reached.

Step 8: extract the optimized hyperparameters and build the model.

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)} \quad (25)$$

where $p(D)$ denotes the marginal probability of the data, $p(w)$ denotes the prior distribution of the parameters w , $p(w|D)$ denotes the posterior distribution, and $p(D|w)$ denotes the likelihood of the data D given the parameters w .

5. Results Analysis

5.1. Data Decomposition

The modeling process of the decomposition integration model involves data decomposition, determining predictive factors and predictands, and streamflow forecasting in one integrated process. In this study, the optimal decomposition window lengths for VMD and SSA were determined based on the central frequency aliasing of the last component. Taking the SSA decomposition of the data from the Yangxian hydrological station as an example, window length values from 2 to 12 were evaluated, and central frequency aliasing occurred at $k = 9$, as shown in Figure 5. The emergence of central frequency aliasing indicates that the current window length had sufficiently extracted the information in the time series. Further increasing the window length would only increase the energy of the noise modes without obtaining more useful information. Thus, $k = 9 - 1 = 8$ was chosen as the SSA

decomposition window length. The optimal number of modes for VMD was determined in a similar manner. Figure 6 shows the results of the SSA decomposition into subsignal components for the monthly runoff at Yangxian station.

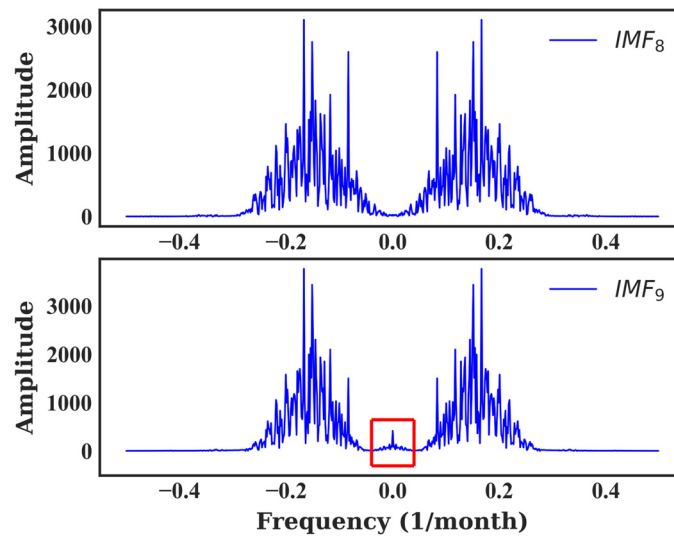


Figure 5. Central frequency aliasing of the last mode of SSA decomposition for the Yangxian hydrostation. The area surrounded by the red box indicates the aliasing phenomenon.

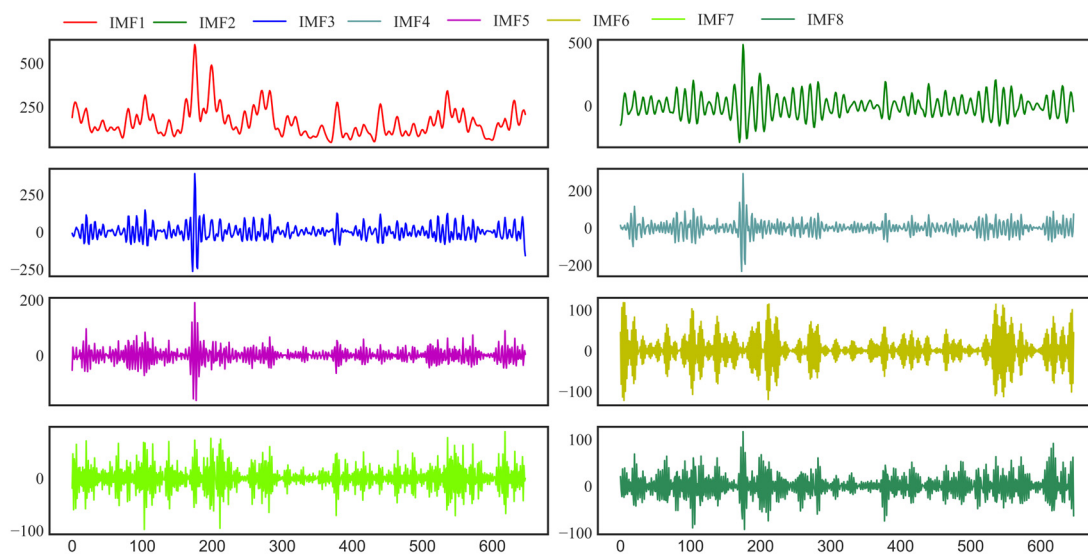


Figure 6. SSA decomposition results of monthly runoff at the Yangxian station.

Following the introduction of EMD, EEMD, and CEEMDAN in Section 3.1, the monthly streamflow data from Yangxian and Hanzhong were decomposed into seven and six subsignals, respectively, by using EMD, eight and seven subsignals, respectively, by using EEMD, and seven and six subsignals, respectively, by using CEEMDAN. When the decomposition levels of DWT were set to 1, 2 and 3, 2, 3, four subsignals were decomposed from the original streamflow data from Yangxian and Hanzhong, respectively.

5.2. Determining Input Variables Using PACF

As described in Section 4.2 for the determination of the temporal feature dimensions of time series, PACF was used to determine the input variables for the model. When the PACF was within the 95% confidence interval, the time series was linearly independent and could

be used as an input variable. If multiple correlation coefficients were within the confidence interval, then the lag term with the minimum correlation coefficient was chosen as the optimal input variable. Taking the first subsignal sequence of SSA decomposition of the data from the Yangxian hydrostation, i.e., IMF1, as an example, the process of determining the input and output variables from Figure 7 through the partial autocorrelation function is illustrated. When t was set to 8 months, the PACF within the 95% confidence interval was 0.03, therefore $t - 1 = 7$ months was the optimal lag term for IMF1. As introduced in Section 3.1, the monthly streamflow data from IMF1 from $t - 7$ months to $t - 1$ month for seven consecutive months as well as the streamflow corresponding to the lag dates determined by PACF for the other IMFs were used as input variables, and the monthly streamflow data for the predicted periods of $(t + 1)$ months, $(t + 3)$ months, $(t + 5)$ months, and $(t + 7)$ months were used as output variables. The generation of input data for the Hanzhong hydrostation was the same as that for the Yangxian hydrostation. Figure 7 shows the process of determining the input variables for Yangxian’s IMF1. The optimal input variables for each IMF of Yangxian are shown in Table 1.

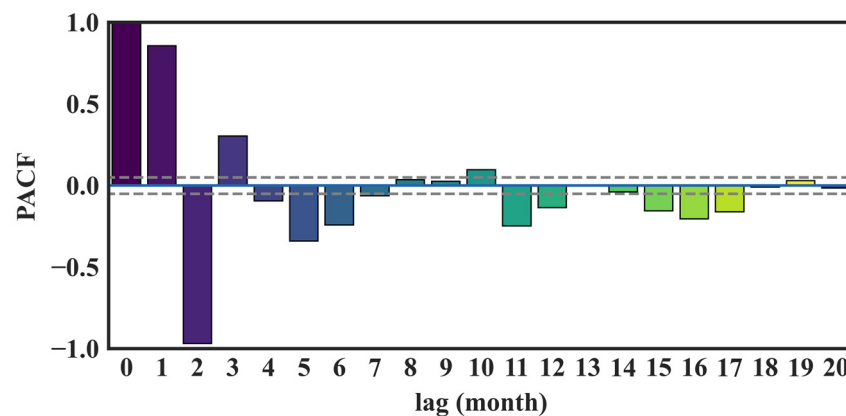


Figure 7. Partial autocorrelation functions of the first subsignal from SSA decomposition of runoff time series data for the Yangxian station (Dashed lines indicate the 95% confidence interval).

Table 1. Optimal input variables for each intrinsic mode function decomposed from the Yangxian time series data using singular spectrum analysis.

Decomposed IMFs	Numbers of Input	Input Variables
IMF ₁	7	$x_1(t - 1)(t - 2)(t - 3)(t - 4)(t - 5)(t - 6)(t - 7)$
IMF ₂	4	$x_2(t - 1)(t - 2)(t - 3)(t - 4)$
IMF ₃	5	$x_3(t - 1)(t - 2)(t - 3)(t - 4)(t - 5)$
IMF ₄	6	$x_4(t - 1)(t - 2)(t - 3)(t - 4)(t - 5)(t - 6)$
IMF ₅	5	$x_5(t - 1)(t - 2)(t - 3)(t - 4)(t - 5)$
IMF ₆	4	$x_6(t - 1)(t - 2)(t - 3)(t - 4)$
IMF ₇	5	$x_7(t - 1)(t - 2)(t - 3)(t - 4)(t - 4)(t - 5)$
IMF ₈	5	$x_8(t - 1)(t - 2)(t - 3)(t - 4)(t - 4)(t - 5)$

5.3. Mutual Information Method for Predictor Screening

Mutual information feature selection is an entropy-based method that can mine associations between different types of variables [47]. Entropy quantifies the amount of information provided by the data. For continuous variables, entropy can be calculated as:

$$h(X) = - \int P_X(x) \ln P_X(x) dx \tag{26}$$

where $h(X)$ is the entropy of feature X and $P_X(x)$ is the probability density function of feature X .

Before deriving the mutual information, another term from information theory must be introduced, i.e., conditional entropy $h(X|Y)$, which can be computed as:

$$h(X|Y) = - \iint P_{XY}(x, y) \ln P_{X|Y}(x|y) dx dy \quad (27)$$

where $P_{XY}(x, y)$ is the joint probability density function of feature X and outcome Y and $P_{X|Y}(x|y)$ is the conditional probability density function of feature X given the outcome Y . The term $h(X|Y)$ quantifies the uncertainty in feature X if the outcome Y is known.

The mutual information, denoted as $I(X; Y)$, measures the reduction in uncertainty about the outcome Y due to knowledge of the feature X . The mutual information can be computed using the following formula:

$$I(X; Y) = h(X) - h(X|Y) \quad (28)$$

This study employs mutual information (MI) to measure the importance of predictor factors. Using the runoff series from Yangxian, the feature matrix generated based on the SSA decomposition and PACF to determine the optimal lag operators was taken as a predictor factor. Taking the prediction target of the monthly runoff forecast one month into the future as an example, the optimal feature factors for the target flow were identified, as shown in Figure 8. In Figure 8, the feature factors with an importance greater than the median are: Periodic2{t - 1}, Periodic1{t - 3}, Periodic5{t - 1}, Periodic4{t - 2}, Periodic2{t - 2}, Periodic2{t - 3}, Periodic2{t - 5}, TREND{t - 3}, Periodic1{t - 4}, Periodic0{t - 2}, Periodic4{t - 1}, Periodic1{t - 1}, TREND{t - 2}, TREND{t - 1}, and Periodic0{t - 1}. The factors with an importance greater than the mean are: Periodic4{t - 2}, Periodic2{t - 2}, Periodic2{t - 3}, Periodic2{t - 5}, TREND{t - 3}, Periodic1{t - 4}, Periodic0{t - 2}, Periodic4{t - 1}, Periodic1{t - 1}, TREND{t - 2}, TREND{t - 1}, and Periodic0{t - 1}. Since there is no universally accepted importance threshold, factors with an importance greater than the mean were selected as predictors for the MIR-LSTM framework. In this way, MI could be used to screen the feature factors identified based on EMD, EEMD, CEEMDAN, DWT, SSA, and VMD decompositions, as well as PACF, to form predictors for the MIR-LSTM frameworks of Yangxian and Hanzhong, respectively.

5.4. Predictor Selection via Recursive Feature Elimination and Cross Validation

Considering the high dimensionality of predictors in the training data, machine learning methods can easily overfit the training data, resulting in poor regression prediction accuracy and weak generalization capability. In fact, not all sub-signals and lagged predictor features determined by PACF positively contributed to prediction accuracy. Recursive feature elimination (RFE) is an efficient feature selection method that reduces model complexity by eliminating irrelevant predictive variables. Technically, RFE is a wrapper feature selection algorithm that also uses filter-based feature selection internally; hence, different machine learning algorithms can be easily combined with the RFE algorithmic core to perform feature selection [48]. This study primarily combines the RFECV algorithm with the CART regression tree model. First, all features in the training set were used to train the CART model. Then, features were ranked according to importance (coefficients or feature importance) and the weakest features were progressively eliminated, with the model refitted at each iteration. Compared to RFE, RFECV adds a cross-validation step after RFE. Also, RFECV does not necessarily prune features down to a specified dimension but stops the iterations at or before the specified dimension is reached.

To prevent the overfitting of the CART regression tree, hyperparameter tuning via grid search (GridSearchCV) was performed at each iteration of subset feature model refitting, further enhancing the reliability of the RFECV feature selection results. The procedure for RFECV is as follows:

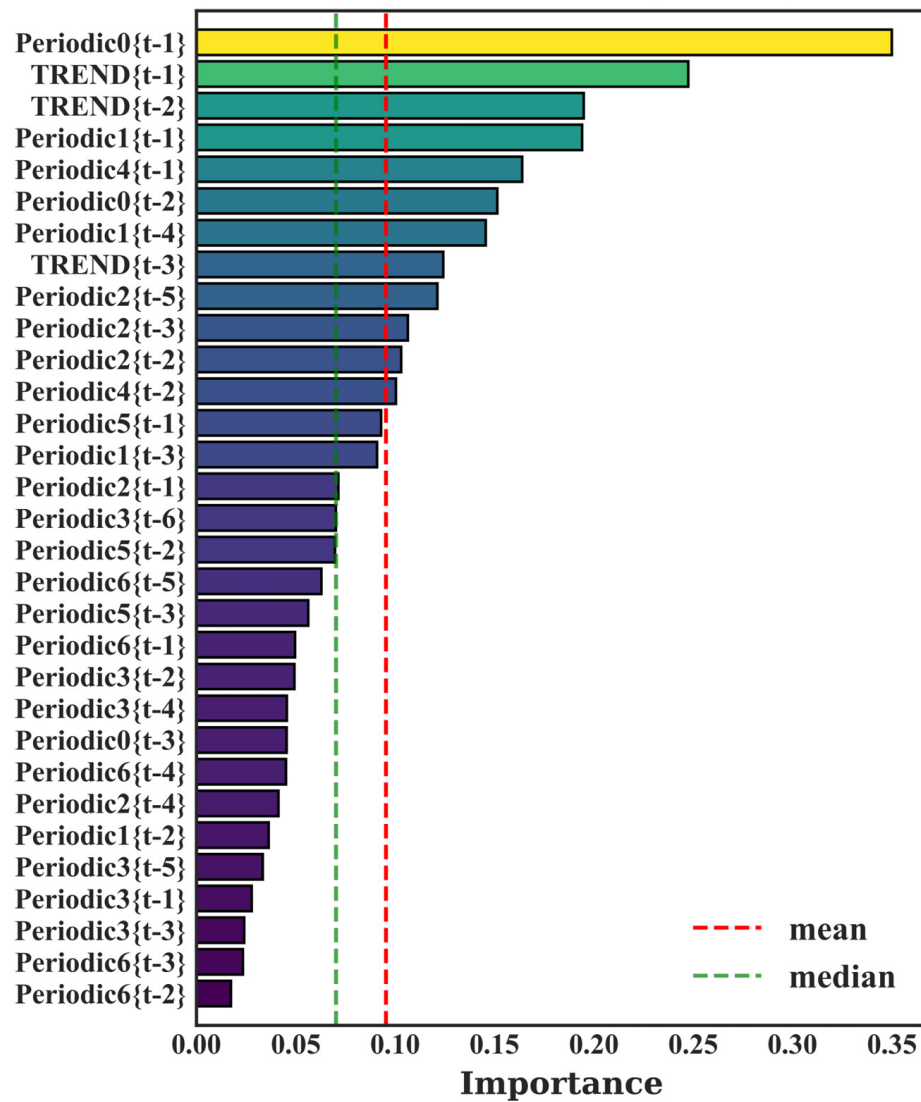


Figure 8. Mutual information between the predictor variables and the forecasting target at the Yangxian hydrological station.

Step 1: initially train the CART regression tree on the full set of features (A1) to obtain the model result r1 and the importance of each feature.

Step 2: eliminate the least important feature, i.e., the feature with the lowest importance score, to obtain a subset of features (A2). Retrain the model on A2 to obtain the result r2 and recalculate the importance of the remaining features.

Step 3: if r2 is better than r1, keep A2 and further eliminate the least important feature to obtain the subset A3. Retrain the model on A3 to obtain the result r3 and the importance of features in A3, repeating this process iteratively. Otherwise, if r1 is better than r2, keep A1 and stop the iterations. Alternatively, stop the iterations if the performance of any subset is weaker than that of its parent set during the iterative process.

With regard to RFECV, the feature importance metric only serves to guide the model toward the identification of relatively weaker features given the current dataset. The final decision on whether to eliminate a feature is still based on the comparison of model performance before and after the exclusion of the feature. As shown in Figure 9 using monthly streamflow data from Yangxian as input, model performance deteriorated after dropping one feature in the third iteration. Therefore, the iterations were stopped and the third subset of features was retained as the optimal predictors for streamflow forecasting.

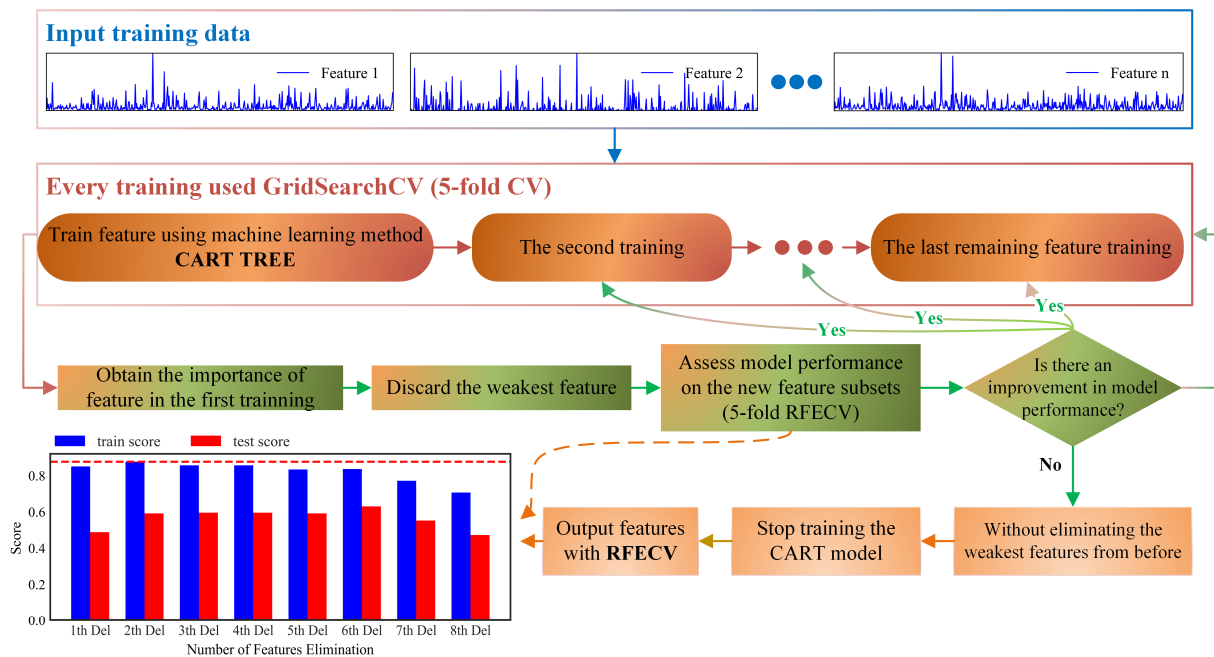


Figure 9. Schematic of the RFECV algorithm.

5.5. Comparison of Model Performance Across Prediction Schemes

Monthly runoff data from the Yangxian and Hanzhong hydrological stations were used to compare the model performance of the direct LSTM, MIR–LSTM, and RFECV–LSTM prediction schemes trained on decompositions from different methods. The evaluation metrics NSE, NRMSE, and PPTS were calculated for each scheme using the training and validation sets. Heatmaps in Figures 10–12 illustrate these results. The x-axis indicates the prediction scheme, station, lead time, and model stage (e.g., Direct_LSTM-YX-L1-T is the training stage of the direct LSTM scheme predicting the streamflow at the Yangxian station at 1 month lead time). The y-axis shows the various time–frequency decomposition algorithms. Beyond CEEMDAN, EEMD, EMD, SSA, and VMD, the remaining methods included DWT (e.g., bior 3.3-1 was DWT with bior 3.3 mother wavelet at decomposition level 1). The black rectangles indicate the decomposition method delivering the best performance for that prediction scheme, station, lead time, and stage. Together, the x-axis and the y-axis represent the full decomposition–integration prediction model. For example, the combination of Direct_LSTM-YX-L1-T (x-axis) and SSA (y-axis) refers to the evaluation metrics from the SSA-decomposed direct LSTM model trained to predict Yangxian monthly runoff at 1 month lead time.

As shown in Figures 10–12, the direct LSTM and RFECV–LSTM schemes exhibited higher NSE values and lower NRMSE and PPTS values compared to the MIR–LSTM scheme across all decomposition methods, stations, lead times, and model stages. This indicates that the direct LSTM and RFECV–LSTM schemes exhibited a superior prediction performance compared to that of the MIR–LSTM scheme, which directly discarded forecast factors below the mutual information threshold, losing valuable information and degrading the predictions. Under the same conditions, RFECV–LSTM achieved higher NSE values and lower NRMSE/PPTS values than the direct LSTM. During the hyperparameter tuning experiments, RFECV–LSTM more easily reached the optimal validation performance given the same upper and lower bounds, while the direct LSTM delivered an unstable validation performance with overfitting and underfitting, requiring multiple hyperparameter boundary adjustments to obtain a good performance. This suggests that RFECV’s lagged screening of the predictors removed those with minor influence or pure noise relative to the target flow. This improved model stability, generalizability, accuracy, and convergence over the direct LSTM using all the predictors. Thus, RFECV feature selec-

tion was effective for monthly runoff prediction. The SSA–LSTM model showed high NSE values and low NRMSE/PPTS values across all combinations of schemes, stations, lead times, and stages, suggesting that SSA decomposition generally improved LSTM runoff prediction. Regardless of decomposition, scheme, station, or stage, NSE values decreased while NRMSE/PPTS values increased with lead times, reflecting a worse performance for longer lead times. Further, by ignoring wavelet type, scheme, station, lead time, and stage, DWT–LSTM models with three levels of decomposition showed higher NSE values and lower NRMSE/PPTS values compared to models with one or two levels. This shows that increasing DWT decomposition levels enhanced prediction performance. With three levels, the 1-month lead time NSE value exceeded 0.95, validating this approach as effective for DWT–LSTM models. Among DWT–LSTM models, db20, db25, db30, db35, db40, and db45 at three levels showed the best performance compared to other wavelets. However, DWT exhibited poor validation, generalizability, and steep performance declines for longer lead times. In summary, RFECV–LSTM demonstrated superior generalizability and optimization convergence, retaining a prediction accuracy close to that of the direct LSTM despite the pruning of some factors. RFECV–LSTM showed better stability and lower dimensionality compared to the all-factor direct LSTM. The MIR–LSTM scheme was less reliable than the RFECV strategies. Comparing all decompositions, the RFECV–SSA–LSTM model performed best in terms of training fit and validation accuracy. Thus, this study recommends the RFECV–SSA–LSTM scheme for monthly runoff prediction.

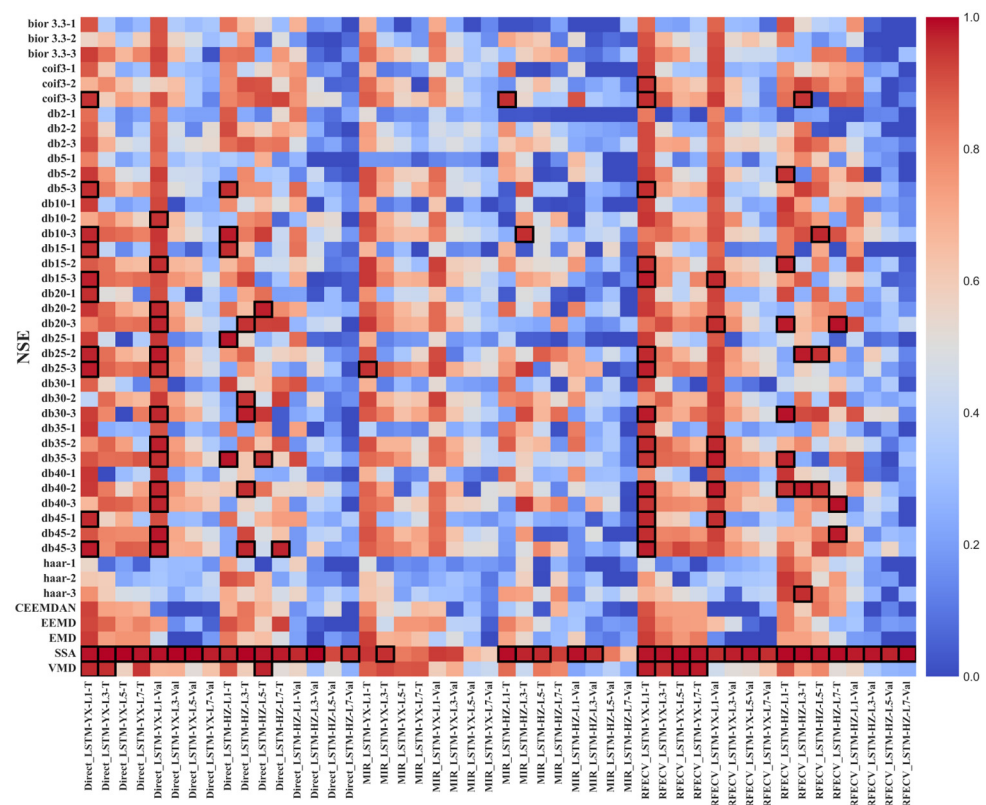


Figure 10. NSE values for direct LSTM, MIR–LSTM, and RFECV–LSTM models using the training and validation sets of data from the Yangxian and Hanzhong stations.

Horizontal comparison experiments were then conducted for the three models: direct LSTM, MIR–LSTM, and RFECV–LSTM, as shown in Figure 13. In Figure 13a, it can be seen that RFECV–LSTM had fewer NSE outlier whiskers than its direct LSTM counterpart when using both validation and test sets, especially notable when using the validation set. MIR–LSTM showed the most NSE outlier whiskers when using the test set, reflecting its poor performance compared to the other models. Although MIR–LSTM had no outlier

whiskers when ran on the validation set, the heatmaps revealed that its predicted NSE values were generally low, thus concentrated at lower values without outliers. Due to the large outlier values, the specific NSE values of the three models are unclear in Figure 13a. However, based on the heatmaps, both the RFECV–LSTM and the direct LSTM models achieved higher NSE values than MIR–LSTM across all scenarios. This demonstrates RFECV–LSTM’s superior generalizability compared to the direct LSTM model, while MIR–LSTM delivered the worst performance in terms of both model stability and accuracy. In Figure 13b, the outlier whiskers of NRMSE for the RFECV–LSTM (val) model are shorter than those associated with the direct LSTM (val) model, while they appear to be comparable between the RFECV–LSTM (test) and the direct LSTM (test) models, with NRMSE clustered around 1. The MIR–LSTM (test) model had the longest outlier whiskers, suggesting that RFECV–LSTM performed best among the three models and was more stable than the direct LSTM model, while MIR–LSTM delivered the worst performance. In Figure 13c, the direct LSTM (val) model has longer outlier whiskers than those associated with the other models, and both the direct LSTM and the RFECV–LSTM models exhibited PPTS around 5 and 50, while MIR–LSTM exhibited PPTs around 70, a value higher than the other two. This indicates the better performance of the direct LSTM and RFECV–LSTM models over MIR–LSTM, while RFECV–LSTM showed no PPTS outliers when using the validation and test sets, yielding more stable and generalizable predictions.

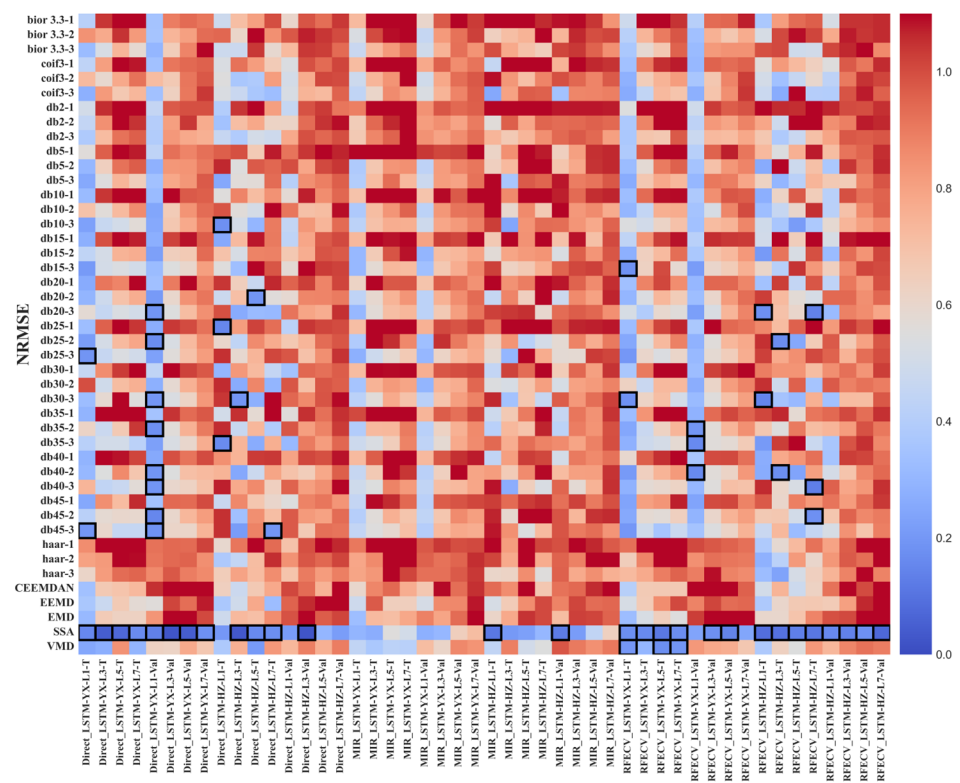


Figure 11. NRMSE values for direct LSTM, MIR–LSTM, and RFECV–LSTM models using the training and validation sets of data from the Yangxian and Hanzhong stations.

As the previous analyses showed that MIR–LSTM had the worst prediction accuracy among the three models, the comparison here focuses on RFECV–LSTM and direct LSTM. Based on the heatmaps, db35-3, db40-2, db45-3, VMD, and SSA were the best performing decomposition algorithms. Box plots were generated to compare RFECV–LSTM and direct LSTM across different model schemes and algorithms, as illustrated in Figure 14.

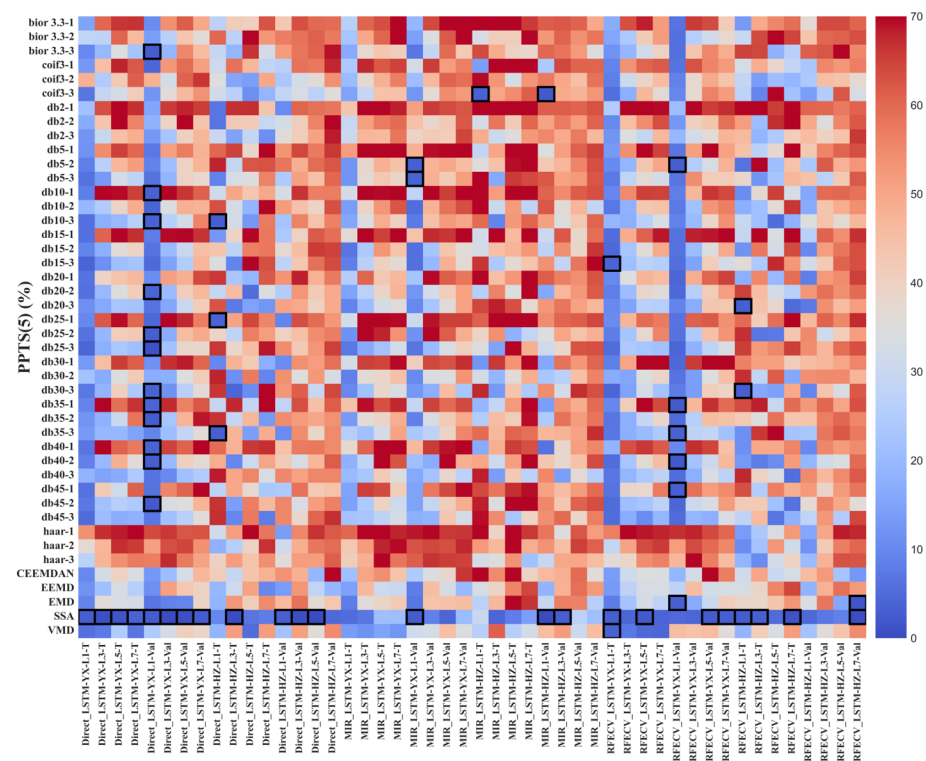


Figure 12. PPTS values for direct LSTM, MIR-LSTM, and RFECV-LSTM models using the training and validation sets of data from the Yangxian and Hanzhong stations.

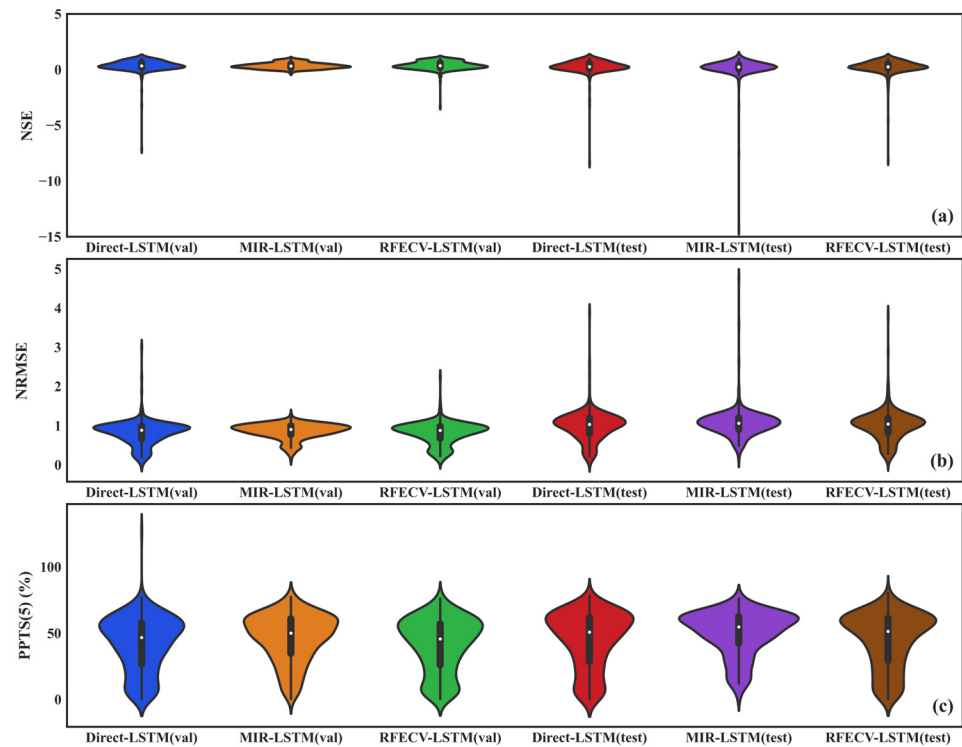


Figure 13. Violin plots showing (a) NSE, (b) NRMSE, and (c) PPTS for validation and test sets of different LSTM models (Direct-LSTM, MIR-LSTM, RFECV-LSTM), where white dots represent the median values.

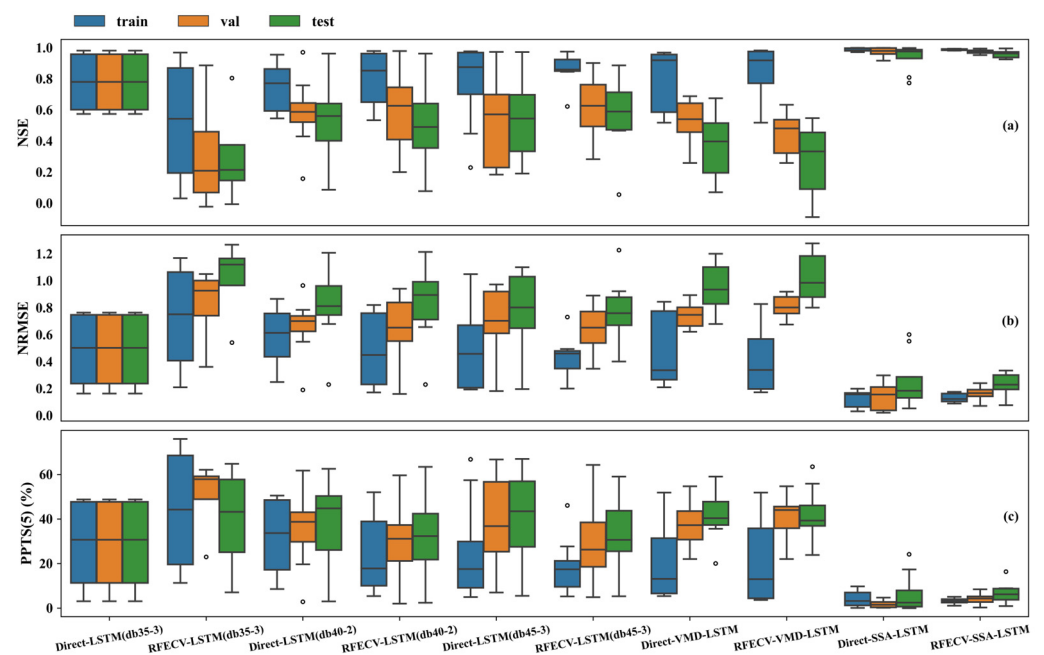


Figure 14. Boxplots comparing model evaluation metrics (a) NSE, (b) NRMSE, and (c) PPTS for training, validation, and test sets of different LSTM models, where white circles represent outliers.

In Figure 14a, both direct SSA and RFECV-SSA achieved high NSE values with very small interquartile ranges close to 1, indicating that the SSA decomposition was capable of the best runoff predictions among all the algorithms. Moreover, RFECV-SSA-LSTM showed smaller interquartile ranges for NSE than the direct SSA-LSTM when using training, validation, and test sets. This demonstrates RFECV-LSTM's superior stability and generalizability compared to the direct LSTM model, corroborated by the NRMSE and PPTS box plots in Figure 14b,c.

Figure 14a also shows that RFECV-db45-3-LSTM exhibited markedly higher mean NSE values and much smaller interquartile ranges than the direct db45-3-LSTM model when using the training set. The NSE advantages persisted when using the validation and test sets, suggesting substantial improvements in both the stability and the accuracy of RFECV-db45-3-LSTM over that of the direct db45-3-LSTM. For VMD decomposition, RFECV-LSTM exhibited a smaller NSE interquartile range than the direct LSTM when using the training set, while the mean NSE value was comparable across training, validation, and test sets, indicating improved stability but similar accuracy. However, as Figure 14a shows, RFECV-LSTM delivered a worse performance than that of the direct LSTM model on both validation and test sets when using db35-3, demonstrating that RFECV-LSTM does not universally improve the direct LSTM model. Overall, RFECV-LSTM managed to avoid overfitting and underfitting better than the direct LSTM model given the quality of the data, leading to enhanced generalizability, convergence, and stability.

5.6. Comparison of Direct and RFECV Prediction Schemes

This study compares the performance of the direct SSA-LSTM, RFECV-SSA-LSTM, direct VMD-LSTM, RFECV-VMD-LSTM, direct DWT-LSTM (db45-3), and RFECV-DWT-LSTM (db45-3) schemes using data from both Yangxian and Hanzhong hydrological stations, with respect to predictions 1, 3, 5, and 7 months into the future using the test set, as shown in Figures 15 and 16. In the plots, direct SSA-LSTM and RFECV-SSA-LSTM exhibited similar predictions clustered around the ideal fitting line, with the smallest angle between linear and ideal fittings. This indicates that the SSA decomposition delivered the best performance and was the most suitable for monthly runoff predictions.

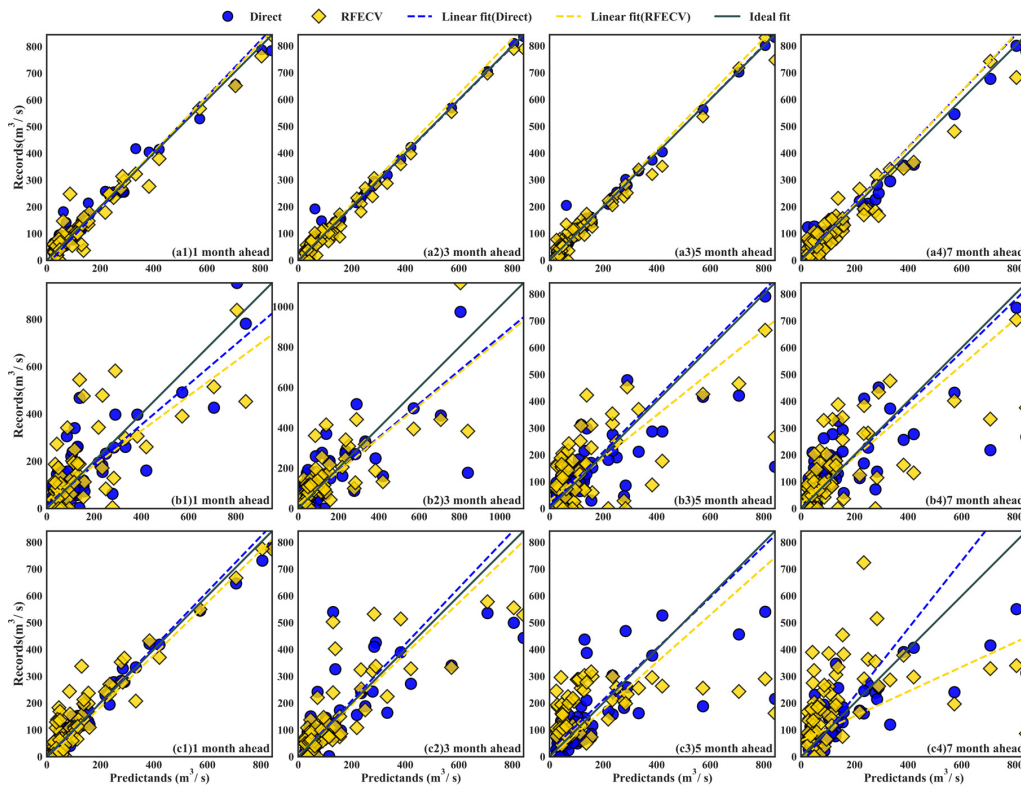


Figure 15. QQ plots for direct and RFECV schemes during testing at the Yangxian (YX) station: (a1–a4) direct SSA–LSTM and RFECV–SSA–LSTM, (b1–b4) direct VMD–LSTM and RFECV–VMD–LSTM, (c1–c4) direct DWT–LSTM (db45-3), and RFECV–DWT–LSTM (db45-3).

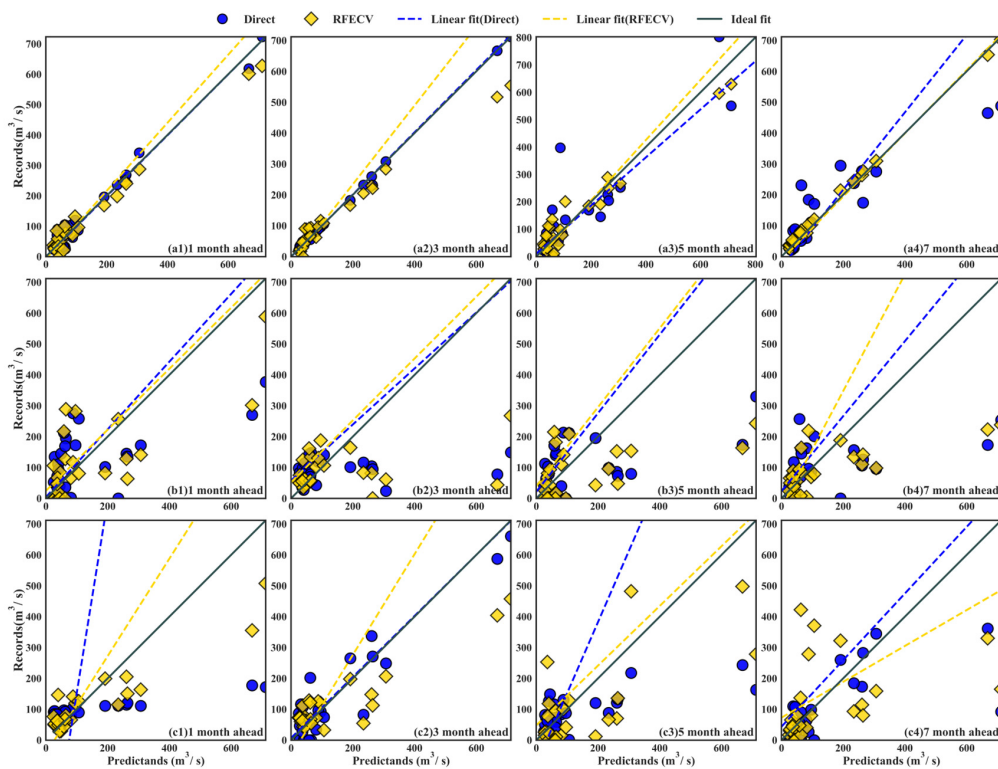


Figure 16. QQ plots for direct and RFECV schemes during testing at the Hanzhong (HZ) station: (a1–a4) direct SSA–LSTM and RFECV–SSA–LSTM, (b1–b4) direct VMD–LSTM and RFECV–VMD–LSTM, (c1–c4) direct DWT–LSTM (db45-3), and RFECV–DWT–LSTM (db45-3).

Comprehensively analyzing the QQ plots of the direct VMD–LSTM, RFECV–VMD–LSTM, direct DWT–LSTM, and RFECV–DWT–LSTM schemes shows the linear fittings commonly lying above the ideal fitting, suggesting that these models generally underestimated the original values, as it can also be observed for the direct SSA–LSTM and RFECV–SSA–LSTM models. At the Hanzhong station, RFECV–DWT–LSTM showed smaller angles between the true linear and ideal fittings compared to the direct DWT–LSTM model overall, while the angle fluctuated more irregularly for the direct DWT–LSTM model, as shown in Figure 16c1–c4. These results indicate minimal performance differences between the direct LSTM and the RFECV–LSTM schemes. However, RFECV feature selection played a major role in filtering out the less relevant factors and improving model performance when data quality was low, as evidenced by the db45-3 decomposition.

6. Discussion

The superior performance of SSA-based LSTM models over other LSTM models using VMD, DWT, CEEMDAN, EEMD, and EMD decompositions, as concluded in Section 5.5, can be explained by the comparisons in Figures 17 and 18. Figure 17 shows heatmaps of Pearson’s correlation coefficients between subsignals from SSA, VMD, and DWT decompositions. Figure 18 presents the spectra of the most difficult to predict subsignals, i.e., the last IMF for SSA and VMD and the first detail coefficient for DWT. As shown in the heatmaps in Figure 17, the VMD and DWT subsignals exhibited weaker Pearson’s correlations than SSA, indicating higher independency. However, as shown in Figure 18, the most difficult SSA subsignal displayed very low noise levels in the low-frequency domain, whereas much greater noise levels were observed for its VMD and DWT counterparts. Although less correlated, the VMD and DWT subsignals contained more noise than the SSA subsignals. Despite their inferior performance relative to SSA, VMD- and DWT-based LSTM models remain powerful decomposition–ensemble forecasting frameworks. Our experiments demonstrate that SSA decomposition is better at controlling mode mixing and noise, lending superior applicability to streamflow forecasting.

As shown in Section 5.5, the RFECV–LSTM forecasting scheme demonstrated superior maturity, reliability, and efficacy over the direct LSTM by incrementally pruning irrelevant features to ensure high correlation between the predictors and the forecasting target. Further removal of low-correlation predictors by RFECV after time–frequency decomposition maximized the accuracy of streamflow forecasting. Moreover, reducing the dimension of the predictors improved LSTM efficiency by lowering computational resource requirements and model training time. The MIR–LSTM delivered the worst performance compared to its RFECV–LSTM and direct LSTM counterparts, as MIR directly discarded predictors with average mutual information below a threshold, omitting valuable information from model training. Although pruning some features may reduce overfitting risks, forecasting performance is paramount.

Section 5.6 demonstrates that the performance gap between the RFECV–LSTM and the direct LSTM forecasting schemes was minor, except for superior RFECV–LSTM results under DWT–(db45-3)–LSTM. This indicates that RFECV is an efficient feature selection approach that can prune predictors with a weak correlation to the forecast target, while preserving predictive capabilities, especially when the predictor set contains members with weak correlation to the target or high dimensionality. In this study, three sample sets were utilized to compare the direct LSTM, MIR–LSTM, and RFECV–LSTM schemes: the training set for model training, the validation set for hyperparameter tuning and model selection, and the test set for evaluating model performance and generalizability. To ensure consistent distributions across the training, validation, and test sets for normalization, the maximum and minimum values from the training set were used. Moreover, the test samples were completely excluded from model training and hyperparameter optimization, further ensuring the credibility of the experimental results.

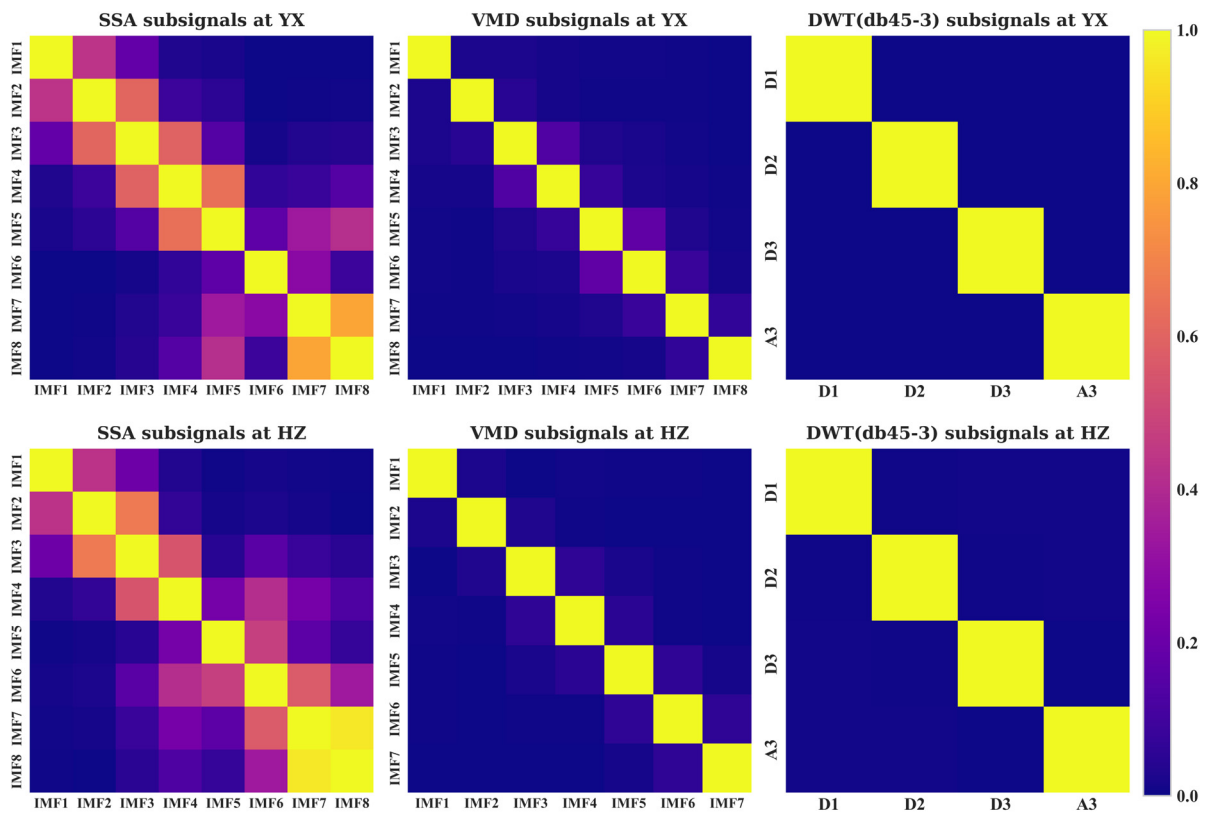


Figure 17. Pearson’s correlation coefficients of subsignals from the Yangxian (YX) and Hanzhong (HZ) stations decomposed by SSA, VMD, and DWT (db45-3).

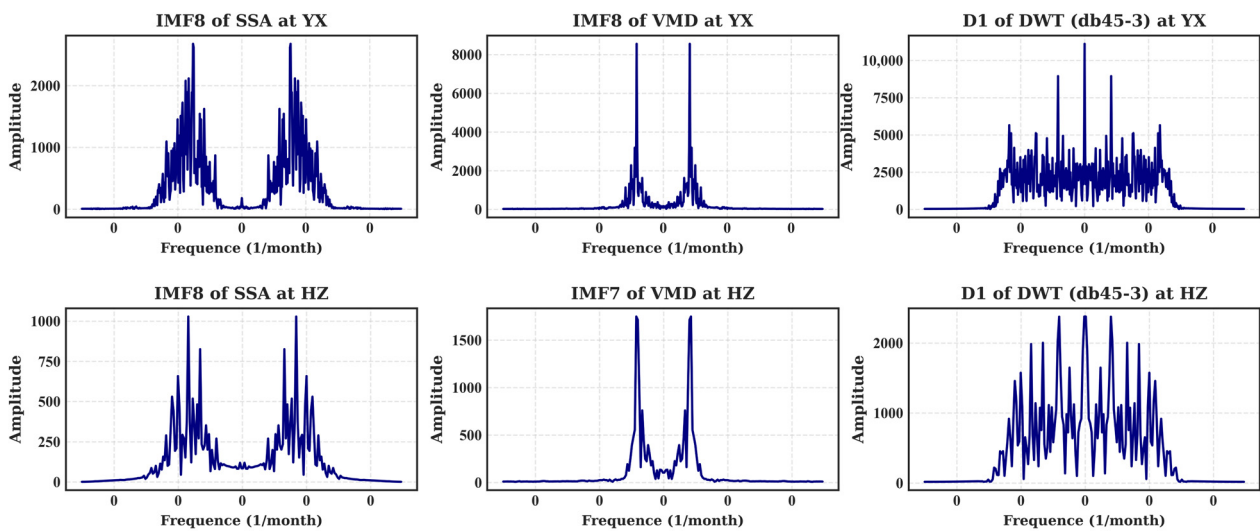


Figure 18. Frequency spectra of IMF8 of SSA, IMF8 of VMD, and D1 of DWT (db45-3) at the Yangxian (YX) station, and IMF8 of SSA, IMF7 of VMD, and D1 of DWT (db45-3) at the Hanzhong (HZ) station.

Although this study used Bayesian optimization to optimize the hyperparameters of the LSTM model and achieved good results, Bayesian optimization has certain limitations. First, in the hyperparameter search space with high dimensionality, Bayesian optimization involves high computational complexity, and constructing and updating the surrogate model increases computational overhead. Additionally, Bayesian optimization is highly sensitive to the selection of initial sampling points, and poor initial points may cause the optimization to fall into a local optimum, thus affecting the final outcome.

Finally, Bayesian optimization is more suitable to small-to-medium-scale optimization problems. If the search space is too large or the training process is very time-consuming, its efficiency may be lower than that of other methods. Therefore, in practical applications, it is necessary to comprehensively consider these factors to determine the suitability of Bayesian optimization.

To address the “black-box” issue in machine learning models and enhance the interpretability of streamflow forecasting, the integration of physics-based models with machine learning models is essential. Physics-based models, grounded in the physical understanding of hydrological and meteorological processes, offer interpretative constraints that ensure that predictions adhere to natural laws. In contrast, machine learning models excel at capturing complex nonlinear and high-dimensionality relationships. The combination of these models not only enhances prediction accuracy, but also improves model transparency and credibility by incorporating physical explanations. Furthermore, this integration offers a framework for attribution analysis, allowing for the identification and quantification of the contribution of various input variables to the prediction outcomes. This approach enhances model interpretability and offers clear guidelines for model optimization. In future studies, the fusion of physics-based and data-driven models will play a key role in overcoming the limitations of current “black-box” models, facilitating more informed decision-making and enhancing model reliability.

7. Conclusions

To enhance the performance, reliability, and efficiency of decomposition–integration models for practical applications, this study proposes an RFECV–LSTM forecasting scheme which iteratively prunes weakly correlated predictors via recursive feature elimination and cross validation (RFECV) to reduce overfitting risks while improving model performance. This lowers computational requirements and increases generalizability and efficiency. The RFECV–LSTM scheme is combined with SSA time–frequency decomposition as RFECV–SSA–LSTM to forecast streamflow 1, 3, 5, and 7 months into the future. To demonstrate the performance of RFECV–SSA–LSTM, comparative analyses are conducted against the direct LSTM and MIR–LSTM schemes based on VMD, CEEMDAN, EEMD, EMD, and DWT decomposition. All comparison experiments utilize monthly streamflow records from 1967 to 2020 from the Yangxian station and from 2000 to 2022 from the Hanzhong station in China’s Hanjiang River Basin. The main conclusions are:

1. The decomposition levels of SSA and VMD can be determined by observing central frequency aliasing in the last subprocess to prevent frequency overlap across subcomponents, minimize intercorrelation, and avoid spurious or redundant elements.
2. When using the db45-3 decomposition, RFECV–LSTM exhibits higher NSE values and lower NRMSE and PPTS values compared to the direct LSTM model when applied to data from the Hanzhong station, with the most pronounced performance gap across all cases. This demonstrates that the recursive pruning of weakly correlated predictors can further enhance predictive accuracy. With high predictor dimensionality and weak correlation to the forecast target, RFECV–LSTM shows a superior forecasting performance over its direct LSTM counterpart. MIR–LSTM performs much worse than the other two schemes by directly removing predictors below the average mutual information value, resulting in some loss of valuable predictive information.
3. Although VMD and DWT yield lower intercorrelation among subcomponents than SSA, their most difficult to predict subcomponents have higher noise levels in the frequency spectrum compared to SSA. Thus, SSA–LSTM achieves the best predictive performance.
4. The proposed RFECV–SSA–LSTM model achieves NSE values greater than 0.9 across all lead times of 1, 3, 5, and 7 months, outperforming the RFECV–LSTM, MIR–LSTM, and direct LSTM forecasting models based on different decomposition methods. Thus, RFECV–SSA–LSTM is a mature, reliable, and effective streamflow forecasting scheme.

In summary, the empirical validation demonstrates that RFECV–SSA–LSTM is the optimal decomposition–integration model and a highly reliable forecasting scheme for predicting highly nonlinear monthly streamflow in the context practical applications. Moreover, RFECV–SSA–LSTM can be implemented to forecast other nonlinear processes such as precipitation, evaporation, temperature, etc. By screening all predictors and selecting those most strongly correlated with the forecast target, predictive accuracy can be further enhanced. When applying the RFECV–SSA–LSTM forecasting model, the following criteria should be met: (1) the decomposed subcomponents must satisfy orthogonality to avoid spurious elements; the SSA decomposition level can be determined by observing central frequency aliasing in the last subcomponent, (2) the data should be split into training, validation, and test sets to avoid using future information; the test set should not participate in model training or hyperparameter tuning to ensure credible model evaluation, (3) min–max normalization should be performed using the min–max values from the training set to ensure the validation and test sets have identical distributions to the training set, (4) when meteorological records are incomplete for the target watershed, RFECV–SSA–LSTM can effectively forecast streamflow solely based on available historical flow data.

Author Contributions: W.M.: methodology, writing, original draft preparation, data analysis. X.Z. (Xiao Zhang): conceptualization, data collection, supervision. Y.S.: visualization, review, supervision. J.X.: review, English editing, G.Z.: revision preparation, X.Z. (Xu Zhang): refinement, T.J.: oversight, data collection. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China, grant number 2023YFC3006503, and by the National Natural Science Foundation of China, grant number 52309034.

Data Availability Statement: The dataset is available upon request from the authors.

Acknowledgments: The authors thank the editor and anonymous reviewers for their comments and suggestions.

Conflicts of Interest: The authors declare no competing interests.

References

1. Xu, Z.; Mo, L.; Zhou, J.; Fang, W.; Qin, H. Stepwise decomposition-integration-prediction framework for runoff forecasting considering boundary correction. *Sci. Total Environ.* **2022**, *851*, 158342. [[CrossRef](#)] [[PubMed](#)]
2. Yang, X.; Chen, Z.; Qin, M. Monthly Runoff Prediction Via Mode Decomposition-Recombination Technique. *Water Resour. Manag.* **2024**, *38*, 269–286. [[CrossRef](#)]
3. Wu, C.L.; Chau, K.W. Data-driven models for monthly streamflow time series prediction. *Eng. Appl. Artif. Intell.* **2010**, *23*, 1350–1367. [[CrossRef](#)]
4. Xie, T.; Zhang, G.; Hou, J.; Xie, J.; Lv, M.; Liu, F. Hybrid forecasting model for non-stationary daily runoff series: A case study in the Han River Basin, China. *J. Hydrol.* **2019**, *577*, 123915. [[CrossRef](#)]
5. Ribeiro, V.H.A.; Reynoso-Meza, G.; Siqueira, H.V. Multi-objective ensembles of echo state networks and extreme learning machines for streamflow series forecasting. *Eng. Appl. Artif. Intell.* **2020**, *95*, 103910. [[CrossRef](#)]
6. Wagena, M.B.; Goering, D.; Collick, A.S.; Bock, E.; Fuka, D.R.; Buda, A.; Easton, Z.M. Comparison of short-term streamflow forecasting using stochastic time series, neural networks, process-based, and Bayesian models. *Environ. Modell. Softw.* **2020**, *126*, 104669. [[CrossRef](#)]
7. Adnan, R.M.; Liang, Z.; Heddam, 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](#)]
8. Kumar, V.; Unal, S.; Bhagat, S.K.; Tiyasha, T. A data-driven approach to river discharge forecasting in the Himalayan region: Insights from Aglar and Paligaad rivers. *Results Eng.* **2024**, *22*, 102044. [[CrossRef](#)]
9. Costa, G.E.D.M.; Menezes Filho, F.C.M.D.; Canales, F.A.; Fava, M.C.; Brandão, A.R.A.; de Paes, R.P. Assessment of Time Series Models for Mean Discharge Modeling and Forecasting in a Sub-Basin of the Paranaíba River, Brazil. *Hydrology* **2023**, *10*, 208. [[CrossRef](#)]
10. Wang, J.; Li, Y.; Gao, R.X.; Zhang, F. Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability. *J. Manuf. Syst.* **2022**, *63*, 381–391. [[CrossRef](#)]
11. Shen, Q.; Mo, L.; Liu, G.; Wang, Y.; Zhang, Y. Interpretable probabilistic modeling method for runoff prediction: A case study in Yangtze River basin, China. *J. Hydrol. Reg. Stud.* **2024**, *52*, 101684. [[CrossRef](#)]

12. Zhai, X.B.; Li, Y.P.; Liu, Y.R.; Huang, G.H. Assessment of the effects of human activity and natural condition on the outflow of Syr Darya River: A stepwise-cluster factorial analysis method. *Environ. Res.* **2021**, *194*, 110634. [[CrossRef](#)] [[PubMed](#)]
13. Yu, Y.; Zhu, R.; Ma, D.; Liu, D.; Liu, Y.; Gao, Z.; Yin, M.; Bandala, E.R.; Rodrigo-Comino, J. Multiple surface runoff and soil loss responses by sandstone morphologies to land-use and precipitation regimes changes in the Loess Plateau, China. *Catena* **2022**, *217*, 106477. [[CrossRef](#)]
14. Tan, Q.; Lei, X.; Wang, X.; Wang, H.; Wen, X.; Ji, Y.; Kang, A. An adaptive middle and long-term runoff forecast model using EEMD-ANN hybrid approach. *J. Hydrol.* **2018**, *567*, 767–780. [[CrossRef](#)]
15. Wang, S.; Peng, H.; Hu, Q.; Jiang, M. Analysis of runoff generation driving factors based on hydrological model and interpretable machine learning method. *J. Hydrol. Reg. Stud.* **2022**, *42*, 101139. [[CrossRef](#)]
16. Ditthakit, P.; Pinthong, S.; Salaeh, N.; Weekaew, J.; Thanh Tran, T.; Bao Pham, Q. Comparative study of machine learning methods and GR2M model for monthly runoff prediction. *Ain Shams Eng. J.* **2022**, *14*, 101941. [[CrossRef](#)]
17. Yong, W.; Zhang, H.; Fu, H.; Zhu, Y.; He, J.; Xie, J. Improving prediction accuracy of high-performance materials via modified machine learning strategy. *Comput. Mater. Sci.* **2022**, *204*, 111181. [[CrossRef](#)]
18. Mao, G.; Wang, M.; Liu, J.; Wang, Z.; Wang, K.; Meng, Y.; Zhong, R.; Wang, H.; Li, Y. Comprehensive comparison of artificial neural networks and long short-term memory networks for rainfall-runoff simulation. *Phys. Chem. Earth Parts A/B/C* **2021**, *123*, 103026. [[CrossRef](#)]
19. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *8*, 1735–1780. [[CrossRef](#)]
20. Li, W.; Liu, C.; Xu, Y.; Niu, C.; Li, R.; Li, M.; Hu, C.; Tian, L. An interpretable hybrid deep learning model for flood forecasting based on Transformer and LSTM. *J. Hydrol. Reg. Stud.* **2024**, *54*, 101873. [[CrossRef](#)]
21. Man, Y.; Yang, Q.; Shao, J.; Wang, G.; Bai, L.; Xue, Y. Enhanced LSTM Model for Daily Runoff Prediction in the Upper Huai River Basin, China. *Engineering* **2022**, *24*, 229–238. [[CrossRef](#)]
22. Gou, J.; Miao, C.; Duan, Q.; Zhang, Q.; Guo, X.; Su, T. Seasonality and Impact Factor Analysis of Streamflow Sensitivity to Climate Change Across China. *Earth's Future* **2022**, *10*, e2022EF003062. [[CrossRef](#)]
23. Kumar, V.; Sen, S. Analysis of Spring Discharge in the Lesser Himalayas: A Case Study of Mathamali Spring, Aglar Watershed, Uttarakhand. *Water Sci. Technol. Libr.* **2018**, *78*, 321–338.
24. Apaydin, H.; Taghi Sattari, M.; Falsafian, K.; Prasad, R. Artificial intelligence modelling integrated with Singular Spectral analysis and Seasonal-Trend decomposition using Loess approaches for streamflow predictions. *J. Hydrol.* **2021**, *600*, 126506. [[CrossRef](#)]
25. Zuo, G.; Luo, J.; Wang, N.; Lian, Y.; He, X. Decomposition ensemble model based on variational mode decomposition and long short-term memory for streamflow forecasting. *J. Hydrol.* **2020**, *585*, 124776. [[CrossRef](#)]
26. Gao, Z.; Yin, X.; Zhao, F.; Meng, H.; Hao, Y.; Yu, M. A two-layer SSA-XGBoost-MLR continuous multi-day peak load forecasting method based on hybrid aggregated two-phase decomposition. *Energy Rep.* **2022**, *8*, 12426–12441. [[CrossRef](#)]
27. Shoaib, M.; Shamseldin, A.Y.; Melville, B.W.; Khan, M.M. A comparison between wavelet based static and dynamic neural network approaches for runoff prediction. *J. Hydrol.* **2016**, *535*, 211–225. [[CrossRef](#)]
28. Huang, S.; Chang, J.; Huang, Q.; Chen, Y. Monthly streamflow prediction using modified EMD-based support vector machine. *J. Hydrol.* **2014**, *511*, 764–775. [[CrossRef](#)]
29. Wang, W.; Chau, K.; Qiu, L.; Chen, Y. Improving forecasting accuracy of medium and long-term runoff using artificial neural network based on EEMD decomposition. *Environ. Res.* **2015**, *139*, 46–54. [[CrossRef](#)]
30. Xiao, H.; Zhang, J. Multi-temporal relations between runoff and sediment load based on variable structure cointegration theory. *Int. J. Sediment Res.* **2023**, *38*, 216–227. [[CrossRef](#)]
31. Marques, C.A.F.; Ferreira, J.A.; Rocha, A.; Castanheira, J.M.; Melo-Gonçalves, P.; Vaz, N.; Dias, J.M. Singular spectrum analysis and forecasting of hydrological time series. *Phys. Chem. Earth Parts A/B/C* **2006**, *31*, 1172–1179. [[CrossRef](#)]
32. Tan, R.; Hu, Y.; Wang, Z. A multi-source data-driven model of lake water level based on variational modal decomposition and external factors with optimized bi-directional long short-term memory neural network. *Environ. Modell. Softw.* **2023**, *167*, 105766. [[CrossRef](#)]
33. Ni, L.; Wang, D.; Singh, V.P.; Wu, J.; Wang, Y.; Tao, Y.; Zhang, J. Streamflow and rainfall forecasting by two long short-term memory-based models. *J. Hydrol.* **2020**, *583*, 124296. [[CrossRef](#)]
34. Wang, C.; Xiao, Z.; Wu, J. Functional connectivity-based classification of autism and control using SVM-RFECV on rs-fMRI data. *Phys. Medica* **2019**, *65*, 99–105. [[CrossRef](#)] [[PubMed](#)]
35. Xing, H.; Niu, J.; Feng, Y.; Hou, D.; Wang, Y.; Wang, Z. A coastal wetlands mapping approach of Yellow River Delta with a hierarchical classification and optimal feature selection framework. *Catena* **2023**, *223*, 106897. [[CrossRef](#)]
36. Chen, B.; Steinberger, O.; Fenioux, R.; Duverger, Q.; Lambrou, T.; Dodin, G.; Blum, A.; Gondim Teixeira, P.A. Grading of soft tissues sarcomas using radiomics models: Choice of imaging methods and comparison with conventional visual analysis. *Res. Diagn. Interv. Imaging* **2022**, *2*, 100009. [[CrossRef](#)] [[PubMed](#)]
37. Zheng, H.; Lv, W.; Wang, Y.; Feng, Y.; Yang, H. Molecular kinematic viscosity prediction of natural ester insulating oil based on sparse Machine learning models. *J. Mol. Liq.* **2023**, *385*, 122355. [[CrossRef](#)]
38. Ladouali, S.; Katipoğlu, O.M.; Bahrami, M.; Kartal, V.; Sakaa, B.; Elshaboury, N.; Keblouti, M.; Chaffai, H.; Ali, S.; Pande, C.B.; et al. Short lead time standard precipitation index forecasting: Extreme learning machine and variational mode decomposition. *J. Hydrol. Reg. Stud.* **2024**, *54*, 101861. [[CrossRef](#)]

39. Wu, Z.; Huang, N.E. Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method. *Adv. Adapt. Data Anal.* **2009**, *1*, 1–41. [[CrossRef](#)]
40. Feng, Z.; Niu, W.; Wan, X.; Xu, B.; Zhu, F.; Chen, J. Hydrological time series forecasting via signal decomposition and twin support vector machine using cooperation search algorithm for parameter identification. *J. Hydrol.* **2022**, *612*, 128213. [[CrossRef](#)]
41. Bai, P.; Liu, X.; Xie, J. Simulating runoff under changing climatic conditions: A comparison of the long short-term memory network with two conceptual hydrologic models. *J. Hydrol.* **2021**, *592*, 125779. [[CrossRef](#)]
42. Zhou, Y.; Guo, S.; Xu, C.; Chang, F.; Yin, J. Improving the Reliability of Probabilistic Multi-Step-Ahead Flood Forecasting by Fusing Unscented Kalman Filter with Recurrent Neural Network. *Water* **2020**, *12*, 578. [[CrossRef](#)]
43. Adaryani, F.R.; Jamshid Mousavi, S.; Jafari, F. Short-term rainfall forecasting using machine learning-based approaches of PSO-SVR, LSTM and CNN. *J. Hydrol.* **2022**, *614*, 128463. [[CrossRef](#)]
44. Bhandari, H.N.; Rimal, B.; Pokhrel, N.R.; Rimal, R.; Dahal, K.R.; Khatri, R.K.C. Predicting stock market index using LSTM. *Mach. Learn. Appl.* **2022**, *9*, 100320. [[CrossRef](#)]
45. Anh, D.T.; Pandey, M.; Mishra, V.N.; Singh, K.K.; Ahmadi, K.; Janizadeh, S.; Tran, T.T.; Linh, N.T.T.; Dang, N.M. Assessment of groundwater potential modeling using support vector machine optimization based on Bayesian multi-objective hyperparameter algorithm. *Appl. Soft Comput.* **2023**, *132*, 109848. [[CrossRef](#)]
46. Su, Z.; Wang, Y.; Tan, B.; Cheng, Q.; Duan, X.; Xu, D.; Tian, L.; Qi, T. Performance prediction of disc and doughnut extraction columns using bayes optimization algorithm-based machine learning models. *Chem. Eng. Process. Process Intensif.* **2023**, *183*, 109248. [[CrossRef](#)]
47. Yan, X.; Liu, D.; Xu, W.; He, D.; Hao, H. Hydraulic fracturing performance analysis by the mutual information and Gaussian process regression methods. *Eng. Fract. Mech.* **2023**, *286*, 109285. [[CrossRef](#)]
48. Li, L.; Ching, W.; Liu, Z. Robust biomarker screening from gene expression data by stable machine learning-recursive feature elimination methods. *Comput. Biol. Chem.* **2022**, *100*, 107747. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.