

Article

Comparative Study for Daily Streamflow Simulation with Different Machine Learning Methods

Ruonan Hao ^{1,2}  and Zhixu Bai ^{3,4,*} 

¹ Academician Workstation in Anhui Province, Anhui University of Science and Technology, Huainan 232001, China; rolinhao@zju.edu.cn

² School of Earth and Environment, Anhui University of Science and Technology, Huainan 232001, China

³ College of Civil Engineering and Architecture, Wenzhou University, Wenzhou 325035, China

⁴ Key Laboratory of Engineering and Technology for Soft Soil Foundation and Tideland Reclamation of Zhejiang Province, Wenzhou 325035, China

* Correspondence: zhixu_bai@wzu.edu.cn

Abstract: Rainfall–runoff modeling has been of great importance for flood control and water resource management. However, the selection of hydrological models is challenging to obtain superior simulation performance especially with the rapid development of machine learning techniques. Three models under different categories of machine learning methods, including support vector regression (SVR), extreme gradient boosting (XGBoost), and the long-short term memory neural network (LSTM), were assessed for simulating daily runoff over a mountainous river catchment. The performances with different input scenarios were compared. Additionally, the joint multifractal spectra (JMS) method was implemented to evaluate the simulation performances during wet and dry seasons. The results show that: (1) LSTM always obtained a higher accuracy than XGBoost and SVR; (2) the impacts of the input variables were different for different machine learning methods, such as antecedent streamflow for XGBoost and rainfall for LSTM; (3) XGBoost showed a relatively high performance during dry seasons, and the classification of wet and dry seasons improved the simulation performance, especially for LSTM during dry seasons; (4) the JMS analysis indicated the advantages of a hybrid model combined with LSTM trained with wet-season data and XGBoost trained with dry-season data.

Keywords: LSTM; XGBoost; different input scenarios; joint multifractal spectra



Citation: Hao, R.; Bai, Z.

Comparative Study for Daily Streamflow Simulation with Different Machine Learning Methods. *Water* **2023**, *15*, 1179. <https://doi.org/10.3390/w15061179>

Academic Editor: Marco Franchini

Received: 15 February 2023

Revised: 9 March 2023

Accepted: 15 March 2023

Published: 18 March 2023



Copyright: © 2023 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

Runoff simulation and forecasting has always been a research hotspot in hydrological science due to the complex runoff fluctuations and its essential role in guiding water resource management [1–3]. Runoff generation and routing involve the coupling effects of meteorological, geographic, geological, soil, and vegetation factors, etc. Some unclear runoff processes still exist at a catchment scale. Furthermore, a significantly increasing trend of runoff variation with the intensification of climate change and human activities has been shown [4,5]. These issues challenge the accuracy of runoff modeling, especially for runoff with the characteristics of rising and falling sharply in mountain rivers. Besides, water conservancy projects in mountain rivers are relatively inadequate for flood control and water supply. The contradiction between water demand and supply has been a long-standing issue. Consequently, accurate runoff prediction in mountain rivers is especially critical for alleviating damages by floods and droughts.

Recently, machine learning (ML) methods have shown great potential in runoff simulation and forecasting [6–8]. ML methods can be classified into artificial neural networks, decision trees and ensemble methods, support vectors machines, Bayesian methods, and so on [9,10]. Many studies have compared the performances of different ML methods

to reveal their applicability in streamflow simulation and forecasting in different catchments. Parisouj et al. [11] showed that support vector regression (SVR) produced higher accuracy than extreme learning machine (ELM) and feed-forward neural network (FFNN). Li et al. [12] and Liu et al. [13] concluded that extreme gradient boosting (XGBoost) performed much better than random forest. Long-short term memory neural network (LSTM) always presented better results than convolutional neural network or traditional machine learning models [14–16]. Particularly, a LSTM model with one hidden layer can always satisfy the demand for streamflow forecasting, and obtain a better performance than one with multiple hidden layers [14,15].

However, there is always no unified guidance for selecting ML methods in order to obtain superior simulation accuracy. Runoff simulation performances of the same ML method are distinct in different catchments. Nevertheless, the various streamflow simulation and forecasting performances in different case studies facilitate the comparative analysis of the efficiency and flexibility of different ML methods, guiding the applicable conditions of various ML methods. Generally, simulation performances at coarser time scales and within a shorter lead time are always better than those at finer time scales and within a longer lead time [12]. The relationship between the watershed area and the simulation effects of ML models is not obvious [11,13,17]. Besides, ML methods always work well even in mountainous or snow-dominated watersheds, but perform poorly in low streamflow regimes [14,18,19].

Benefiting from rapid growth in related studies, some insights into selecting inputs of ML or feature selection can also be concluded for streamflow forecasting. Moosavi et al. [20] revealed that input data were the most important factor affecting the forecasting accuracy when compared to the factors of model type and preprocessing. The factors antecedent streamflow and rainfall are often selected as the first choice [16,17,21]. Other meteorological factors, such as temperature, can also be considered in modeling [11]. Global climate indexes, for example, the Nino index, can facilitate the performance of daily forecasting under long lead times or monthly forecasting [13,22]. According to hydrologic knowledge, the impacts of other related input factors, such as vegetation cover and groundwater storage, have also been explored [23,24].

Besides, a reforecast dataset, including the products of the global forecasting system model or global flood awareness system, can be useful for improving the performance of streamflow forecasting [13,25]. Particularly, snow cover area may be essential for predicting streamflow in snowmelt-dominated basins [11,14]. Thus, the selection of input variables is of great importance for model construction.

The goal of this study is to analyze the performances of various ML methods with different input scenarios and training data for simulating daily runoff over a mountainous river catchment. Three methods, namely SVR, XGBoost, and LSTM, which are under different classifications of ML methods, were chosen. To analyze the impacts of rainfall and antecedent streamflow on modeling accuracy, various input scenarios, including two single-input scenarios and three multiple-input scenarios, were compared. Additionally, their simulation performances were analyzed during dry and wet seasons. The hypothesis was that different rainfall–runoff mechanisms during wet and dry seasons would lead to significant simulation differences during different seasons and for different machine learning methods.

2. Material and Methods

2.1. Study Area and Data Preprocessing

The north tributary of the Ao River (ARNT), a small mountainous catchment in Zhejiang Province, China, was examined in this study. The Ao River covers 1580.4 km² with two main tributaries flowing into the East China Sea. In this study, Daitou Hydrological Station, a national hydrological station, was selected as the control section, and the basin area is 346 km² (Figure 1). The mean annual discharge at the Daitou Station is 16.33 m³/s, while the maximum discharge is 3680 m³/s and the minimum discharge is 0.57 m³/s. The

average annual precipitation and temperature in the ARNT catchment is about 2000 mm and 17.8 °C, respectively. Generally, there are two rainy periods each year, which are caused by spring–summer monsoons (March to June, 42% of annual precipitation) and typhoons (August to September, 29% of annual precipitation). Typhoons always bring concentrated and extreme rainfall and cause extreme floods. By contrast, when there are fewer typhoons in the year, drought is more likely to occur. Thus, it is of equal importance to study the hydrological processes of both wet seasons (April to October) and dry seasons (November to March) in the ARNT catchment thoroughly.

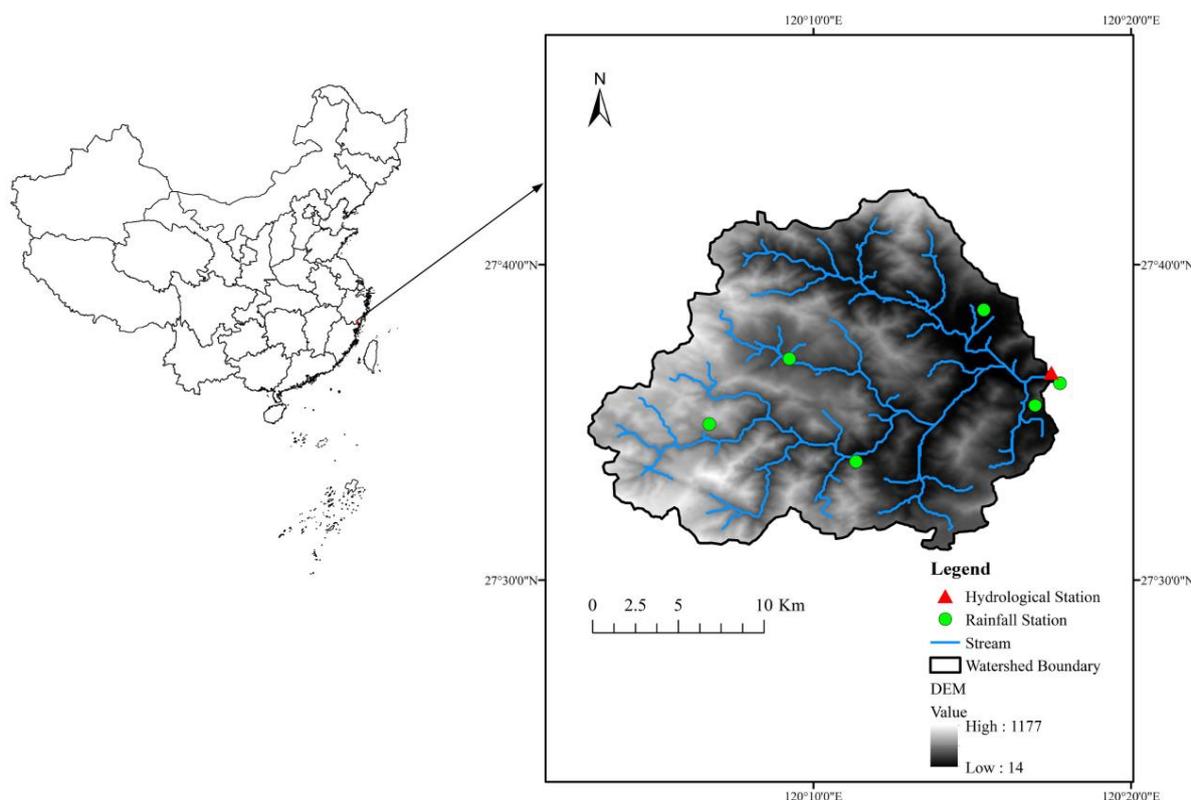


Figure 1. North tributary of the Ao River (ARNT) catchment.

The data used in this study include daily precipitation data from six rainfall stations noted as green circles and daily discharge data from one hydrological station (Daitou Station) noted as a red triangle in Figure 1. The study period is from 1990 to 2013, before the Shunxi Reservoir, the most important hydraulic project in the ARNT catchment, was completed. The areal average rainfall was calculated by the Thiessen polygon method. For constructing runoff simulation models, all the data were divided into three datasets: data measured over 14 years (1991–2004) for training, five years (2005–2009) for validation, and four years (2010–2013) for testing. Five input scenarios, including two single-input scenarios and three multiple-input scenarios, are listed in Table 1. P_i denotes the daily rainfall from one rainfall station. \bar{P} is the areal average rainfall. The subscript t stands for the time.

Table 1. Five input scenarios for three machine learning methods.

Input Scenarios	Input Variables
I	Q_{t-1}
II	\bar{P}_t
III	$P_{1,t}, P_{2,t}, P_{3,t}, P_{4,t}, P_{5,t}, P_{6,t}$
IV	\bar{P}_t, Q_{t-1}
V	$P_{1,t}, P_{2,t}, P_{3,t}, P_{4,t}, P_{5,t}, P_{6,t}, Q_{t-1}$

2.2. Support Vector Regression (SVR)

SVR is the application of the SVM (support vector machine; SVM) in regression problems based on the rule of structural risk minimization [26]. Owing to the robust performance and simple operation of SVR, it has been a classical machine learning method in streamflow forecasting [27,28]. The regression function of SVR can be described as follows:

$$y = w\phi(x) + b \quad (1)$$

where w and b are the weights and bias, respectively. $\phi(\cdot)$ is the mapping function mapping from the input space, x , to a high dimensional space.

To allow a predefined error, ε , in the regression function, the ε -insensitive loss function is defined:

$$L_\varepsilon(d, y) = |d - y|_\varepsilon = \max(0, |d - y| - \varepsilon) \quad (2)$$

The loss is zero when the deviation between target values, d , and output values, y , is within the tolerance error. SVR aims at finding an optimal hyperplane, which is as flat as possible with the minimum loss function [29]. Hence, the optimization problem can be written as:

$$\min f = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N |d - y|_\varepsilon \quad (3)$$

where the minimum of the penalty term, $\|w\|^2$, ensures the flatness of the regression function. N is the sample size. C is a constant that determines the trade-off between the flatness and deviations above the pre-defined threshold.

2.3. Extreme Gradient Boosting (XGBoost)

XGBoost, proposed by Chen and Guestrin [30], is an improved gradient-boosting decision tree algorithm (GBDT). The rapid and accurate learning makes XGBoost a superior machine learning model in data sciences, including hydro-meteorological modeling [13,31,32]. XGBoost is a tree ensemble model using K additive models, and its predicted values can be written as:

$$y = \sum_{k=1}^K f_k(x) \quad (4)$$

where f_k corresponds to a tree structure with T leaves and leaf weights, w .

A major difference between XGBoost and GBDT is that they use different objective functions. A regularization term is added for the XGBoost model, expressed as:

$$\text{Obj} = \sum_{i=1}^N l(y_i, d_i) + \sum_{k=1}^K \Omega(f_k) \quad (5)$$

where l is the loss function measuring the distance between the predicted value, y , and target value, d , such as the mean-square error used in this study. The second-order Taylor expansion is performed in the loss function to ensure its accuracy. The second term, $\sum_{k=1}^K \Omega(f_k)$, sums up the complexity of each base function, f , to avoid overfitting, defined as:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (6)$$

where γ and λ are penalty parameters, controlling the number of leaves, T , and the L^2 norm of the leaf weights, w , respectively.

2.4. Long-Short Term Memory Neural Network (LSTM)

LSTM is a modified version of the recurrent neural network, which deals with long-term dependency in sequences by constructing the memory cell, C , and hidden state, h , [33].

The advantage of learning from long-term dependent time series makes LSTM successful in many fields, such as rainfall–runoff modeling in hydrology [7,34]. LSTM has a chain of repeating modules of LSTM cells, transferring the memory cell, C , and hidden state, h , to the next LSTM cell. Each LSTM cell is composed of a forget gate (f_t), input gate (i_t), and output gate (o_t). The architecture of a LSTM cell is shown in Figure 2.

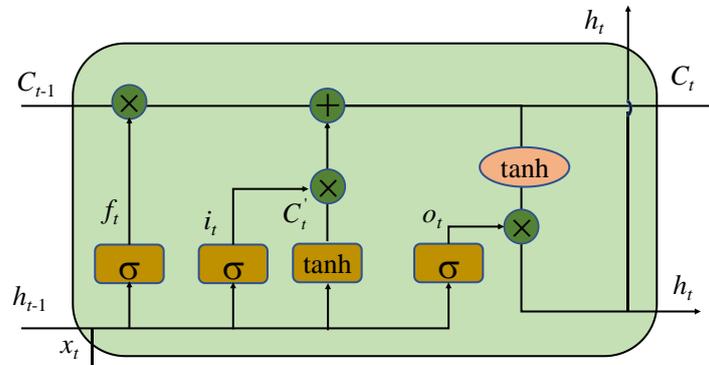


Figure 2. Architecture of LSTM cell.

The forget gate, f_t , determines the extent to which information is discarded for the memory cell, C , defined as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{7}$$

where σ is the sigmoid activation function. W and b are the weights and bias, respectively. The subscript f stands for the forget gate. X is the input vector.

The input gate, i_t , is designed to add new information for updating memory cell, C_t . The calculation formulas can be

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{8}$$

$$C'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{9}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C'_t \tag{10}$$

where \tanh represents the hyperbolic tangent activation function. The subscripts i and c denote the input gate and new candidate value, C'_t , respectively.

The output gate, o_t , outputs the value at each moment, t , and is obtained by:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{11}$$

The output gate, o_t , and memory cell, C_t , determines the hidden state, h_t , which is expressed as:

$$h_t = o_t \cdot \tanh(C_t) \tag{12}$$

In this study, the mean squared error cost function was set to measure how well the model fit the training data.

2.5. Bayesian Optimization

Bayesian optimization is an efficient global optimization algorithm for identifying the optimal hyperparameters in many machine learning applications when the objective function is non-convex and multimodal [35–38]. Bayesian optimization explores for the next parameter set based on sequential optimization. In the exploration, Baye’s theorem is used:

$$p(f|D) = \frac{p(D|f)p(f)}{p(D)} \tag{13}$$

where f is the objective function, and the accuracy rate of the validation set was used in this study. $D = \{\chi, f(\chi)\}$ is the pair of hyperparameters, χ , and its objection function value. The posterior distribution, $p(f | D)$, is obtained through the prior distribution, $p(f)$, and the likelihood function, $p(D | f)$, which are calculated according to the samples of the hyperparameters, χ . According to two critical procedures, namely the probabilistic surrogate model and the acquisition function, the new samples are added to update the posterior distribution at each iteration.

The probabilistic surrogate model describes these probability distributions, and determines where the hyperparameters are located corresponding to the maximum or minimum of the objection function. The acquisition function (e.g., the expected improvement) locates the candidate points where the uncertainty in the surrogate model is large or where the model prediction is high. Then, the candidate points are added to the dataset, D , updating the probabilistic surrogate model until the maximum iterations are reached.

2.6. Evaluation Criteria

The performances of various simulated models were evaluated using three criteria, namely Nash–Sutcliffe efficiency (NSE) [39], root-mean-square error (RMSE), and correlation coefficient (CC), which are defined as follows:

$$\text{NSE} = 1 - \frac{\sum_{i=1}^n (d_i - y_i)^2}{\sum_{i=1}^n (d_i - \bar{d})^2} \quad (14)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - y_i)^2} \quad (15)$$

$$\text{CC} = \frac{\sum_{i=1}^n (d_i - \bar{d})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (d_i - \bar{d})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (16)$$

where y_i and d_i are the forecasting value and observation at time, i , respectively, n is the length of the observations, and \bar{y} and \bar{d} are the average of the simulation values and observations.

Besides, the joint multifractal spectra (JMS) method was also used to provide a comprehensive evaluation of the simulation performances during wet and dry seasons. The JMS method is designed with the principle that a well-simulated runoff series should reproduce the fractal characteristics of an observed runoff series. The spectra contain information on high-flow and low-flow simulation without manual intervention. The details of the JMS method can be found in [40].

3. Results and Discussion

The comparison of SVR, XGBoost, and LSTM streamflow-simulating models was assessed with single- and multiple-input scenarios. Furthermore, the simulation performances during wet and dry seasons were evaluated to explore the impacts of rainfall–runoff mechanisms at different periods.

3.1. Simulation Performances with Single-Input Scenarios

The antecedent streamflow combined with the areal rainfall as a single-input variable were fed to the three simulating models, respectively. Their performances are shown in Table 2 according to the criteria evaluation of NSE, RMSE, and CC. Except for LSTM with the input of areal rainfall, none of them could obtain satisfactory results. The obvious contrast between the input variable, antecedent streamflow and areal rainfall, for the LSTM model suggests the significant impact of rainfall on streamflow fluctuation rather than the impact of antecedent streamflow. When the areal rainfall was fed to the LSTM model, fluctuations of streamflow were captured relatively well, although the peak flow was always overestimated. The simulated streamflow of the LSTM model with the antecedent streamflow input could not surpass $100 \text{ m}^3/\text{s}$ without the rainfall driving force (Figures 3 and 4).

Besides, the simulated streamflow during low-flow periods was significantly higher than the observed streamflow overall.

Table 2. Performances of three simulated models with a single-variable input.

Input Scenario	Model	Training			Validation			Testing		
		NSE	RMSE (m ³ /s)	CC	NSE	RMSE (m ³ /s)	CC	NSE	RMSE (m ³ /s)	CC
I	SVR	0.26	31.55	0.51	0.22	63.08	0.47	0.23	27.79	0.48
	XGBoost	0.32	30.21	0.56	0.23	62.69	0.50	0.27	26.95	0.53
	LSTM	0.09	34.83	0.31	0.04	69.97	0.21	0.08	30.34	0.29
II	SVR	0.11	34.46	0.47	0.13	66.46	0.63	0.09	30.12	0.37
	XGBoost	0.22	32.29	0.47	0.22	63.01	0.55	0.10	29.92	0.37
	LSTM	0.69	20.42	0.83	0.68	40.10	0.83	0.64	19.09	0.83

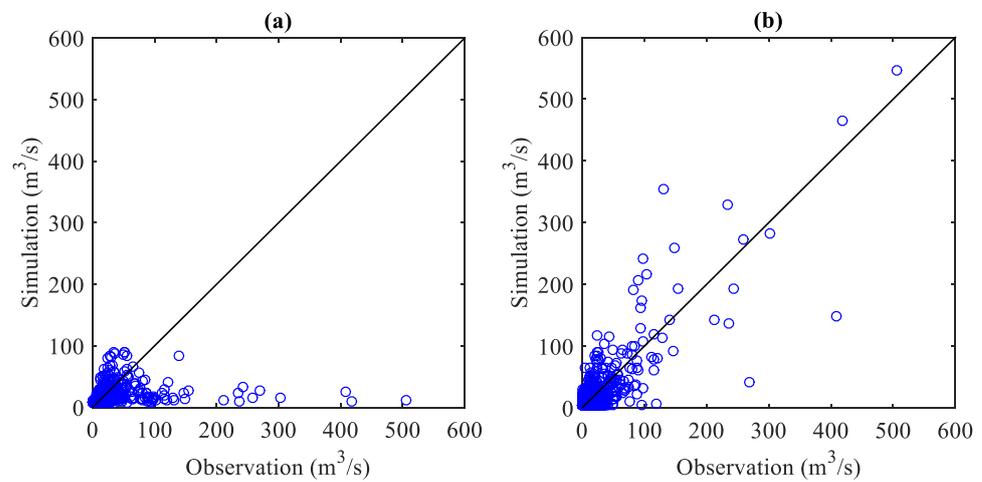


Figure 3. Scatter plots between observations and simulations of discharge during testing periods for LSTM models with input scenarios (a) I and (b) II.

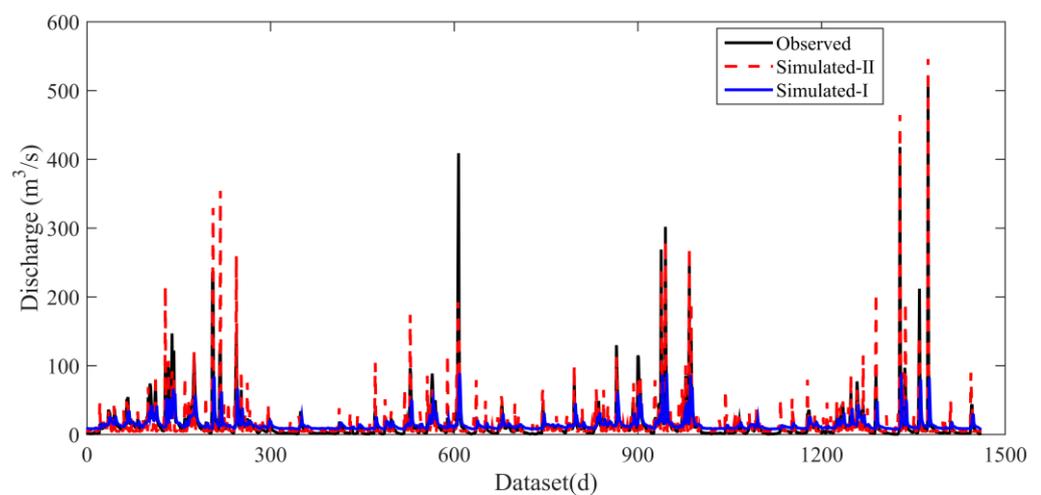


Figure 4. Time series plots for observed and LSTM simulated streamflow with input scenarios I and II during testing periods.

Despite the poor performances of SVR and XGBoost with single-input scenarios, both of them, with the antecedent streamflow input variable, were much better than those with the input of areal rainfall. This indicates the significant difference between neural network

methods and support vector machine methods, and decision trees and ensemble methods for exploring the input and output relationship.

3.2. Simulation Performances with Multiple-Input Scenarios

Table 3 presents the performances of the three models with multiple-input scenarios during the training, validation, and testing periods. Except for the SVR and XGBoost models with the input scenario III, all the models with multiple-variable inputs were better than those with a single-variable input. This indicates that the impacts of various inputs can be different for different models, for example, that of the antecedent streamflow for the SVR and XGBoost models rather than for the LSTM model. Obviously, LSTM outperformed the other machine learning methods with different input scenarios, demonstrating its great potential in streamflow prediction. Both the XGBoost and SVR models still performed unsatisfactorily, with NSE values of less than 0.40 during the testing periods, even though more variables were added for the input.

Table 3. Performances of three simulated models with multiple-variable inputs.

Input Scenario	Model	Training			Validation			Testing		
		NSE	RMSE (m ³ /s)	CC	NSE	RMSE (m ³ /s)	CC	NSE	RMSE (m ³ /s)	CC
III	SVR	0.12	34.34	0.48	0.14	66.25	0.63	0.10	30.03	0.37
	XGBoost	0.25	31.60	0.51	0.31	59.13	0.61	0.08	30.40	0.35
	LSTM	0.70	20.08	0.84	0.72	38.09	0.85	0.67	18.22	0.84
IV	SVR	0.35	29.57	0.60	0.32	58.70	0.58	0.31	26.34	0.56
	XGBoost	0.48	26.36	0.70	0.40	55.07	0.70	0.37	25.02	0.62
	LSTM	0.72	19.30	0.85	0.68	40.32	0.83	0.70	17.27	0.85
V	SVR	0.35	29.56	0.60	0.32	58.91	0.57	0.31	26.31	0.56
	XGBoost	0.61	22.82	0.78	0.54	48.58	0.75	0.33	25.85	0.60
	LSTM	0.75	18.23	0.87	0.72	37.96	0.85	0.74	16.29	0.87

For the LSTM models, the spatial distribution of rainfall in the catchment improved the simulation accuracy significantly, whether this was from the comparisons of input scenarios II and III, or IV and V. The model with input scenario III improved the simulation performance by increasing the NSE by 5.20%, decreasing the RMSE by 4.59%, and increasing the CC by 1.09%, compared with input scenario II for the testing periods. However, the LSTM model with input scenario III always underestimated the recession limbs during floods, and overestimated the fluctuations during low-flow periods (Figures 5 and 6). Another reason for its weakness was that it sometimes maintained a relatively fixed flow during rainless periods. In addition, scenario V for the testing datasets improved the model by increasing the NSE by 4.70%, decreasing the RMSE by 5.66%, and increasing the CC by 2.23%, compared with scenario IV. The LSTM model with input scenario V showed the best performance especially in simulating the peak flow and the recession limbs of floods. Nevertheless, the trained model still could not capture the peak flow when the discharge was over 600 m³/s. This may have resulted from the relatively coarse temporal resolution for rainfall homogenization and the inadequate data for high flows over 600 m³/s.

Although the information of rainfall distribution slightly improved the performances of the XGBoost and SVR models for training, the added information deteriorated the simulation accuracy for testing according to the comparison of their performances between input scenarios II and III. This suggests the inferior ability for generalization of the XGBoost and SVR models for simulating rainfall–runoff processes. Furthermore, the XGBoost models always performed better than the SVR models.

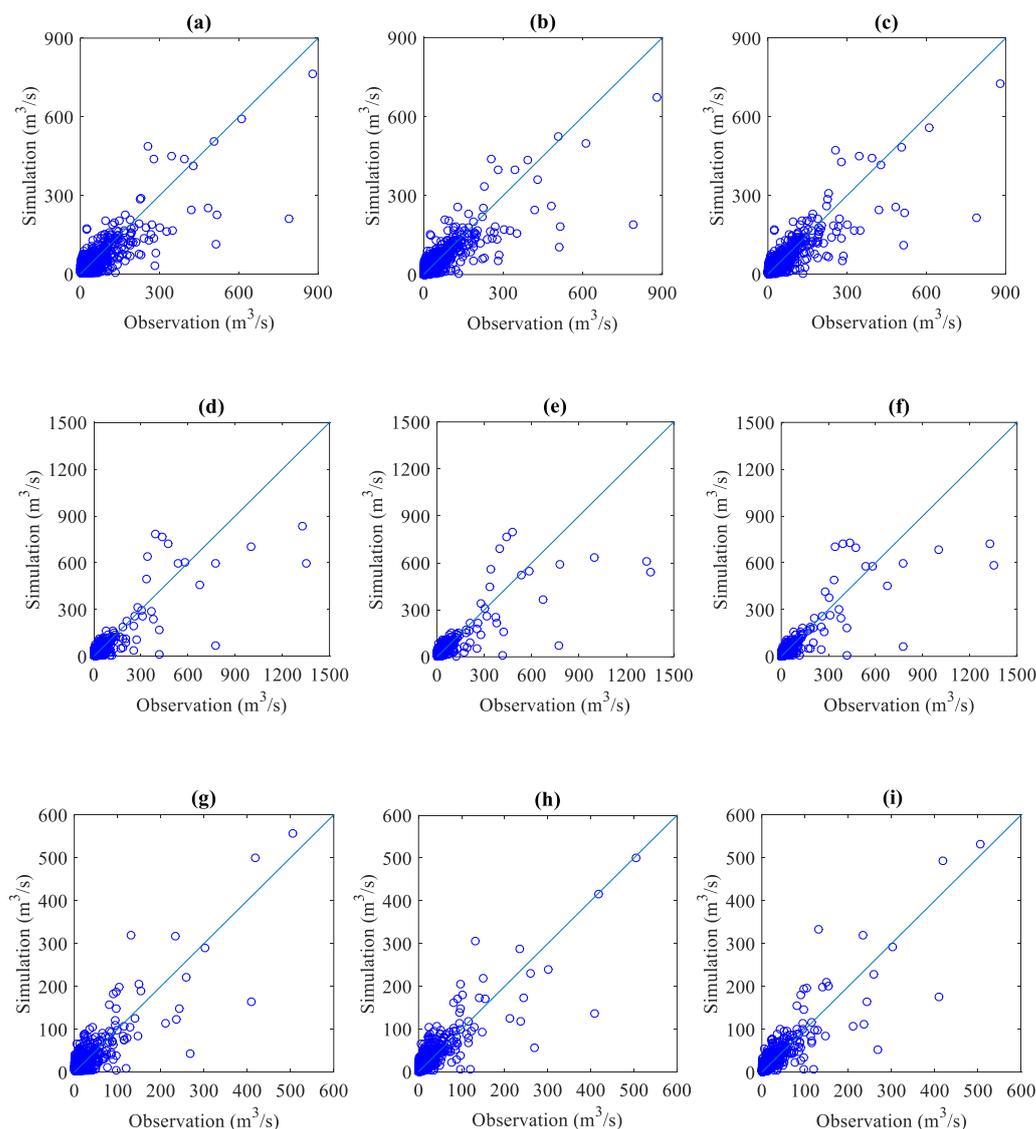


Figure 5. Scatter plots between observations and simulations of discharge during training (a–c), validation (d–f), and testing periods (g–i) for LSTM models with input scenarios III (a,d,g), IV (b,e,f), and V (c,f,i).

3.3. Simulation Performances during Wet and Dry Seasons

Taking the best input scenario as an example, Table 4 shows the simulation performances of these three models during wet and dry seasons. The significantly different performances between wet and dry seasons demonstrated the various rainfall–runoff relationships explored by these models. The LSTM model during wet seasons performed much better than the XGBoost and SVR models (Figures 7 and 8). However, its NSE value during dry seasons was rather low, being less than 0.40, for testing. It was sensitive to the minor disturbances of streamflow, and overestimated the peak flow during dry seasons (Figures 9 and 10). This finding agrees with that of Kim et al. [19], which also showed that better performance can be obtained in the high-flow regime for LSTM. However, other machine learning methods, such as SVR and XGBoost, may not obey this rule. Specially, the XGBoost model captured the streamflow fluctuations fairly during dry seasons, with an NSE value of 0.58 for testing. Additionally, it always underestimated peak flows during wet seasons with an NSE value of 0.30 for testing. The SVR model underestimated runoff peaks during both dry and wet seasons and overestimated low flow during wet seasons.

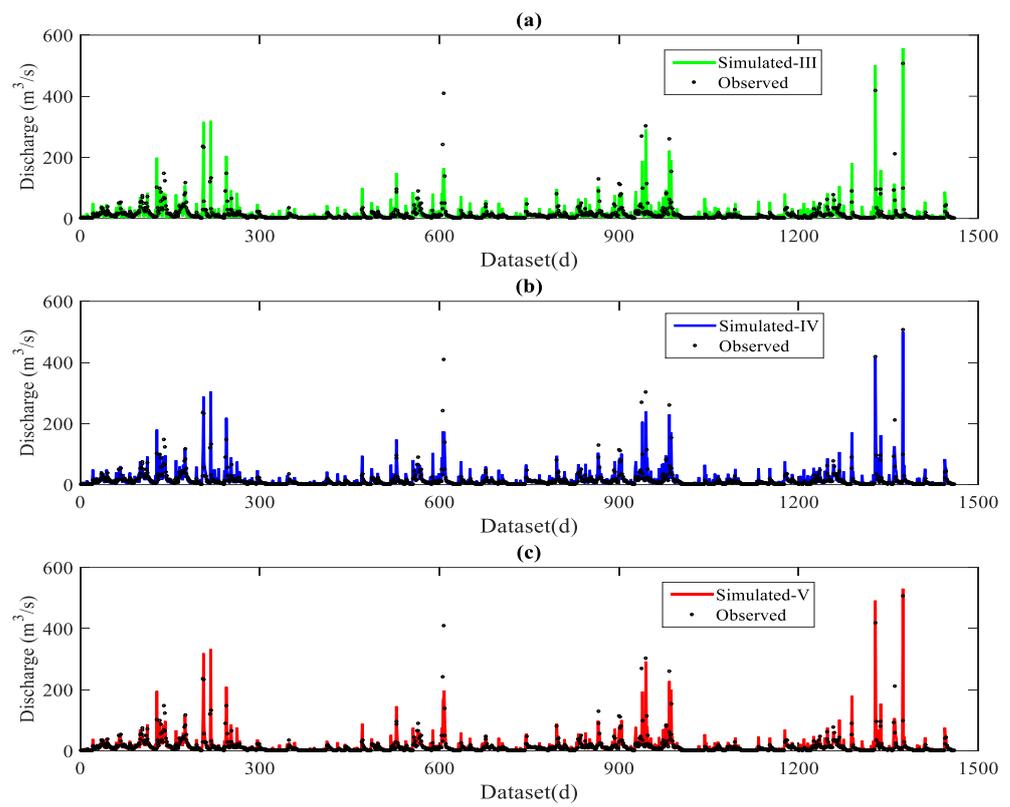


Figure 6. Time series plots for observed and LSTM-simulated streamflow with multiple-variable input scenarios (a) III, (b) IV, and (c) V for testing.

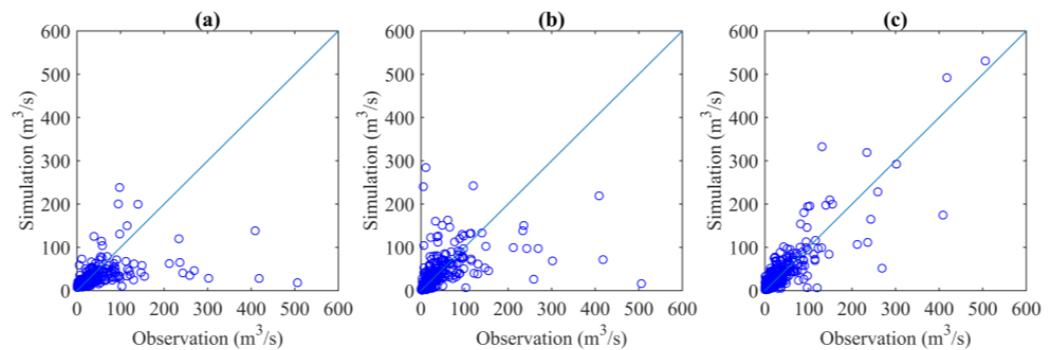


Figure 7. Scatter plots between observations and simulations of discharge during testing wet periods for (a) SVR; (b) XGBoost; and (c) LSTM.

Table 4. Comparison of performances for SVR, XGBoost, and LSTM models during wet and dry seasons.

Model	Period	Training			Validation			Testing		
		NSE	RMSE (m ³ /s)	CC	NSE	RMSE (m ³ /s)	CC	NSE	RMSE (m ³ /s)	CC
SVR	Wet	0.32	37.77	0.58	0.30	76.64	0.53	0.28	33.84	0.53
	Dry	0.38	9.55	0.73	0.07	8.29	0.67	0.32	7.13	0.77
XGBoost	Wet	0.60	29.03	0.78	0.52	63.22	0.74	0.30	33.43	0.57
	Dry	0.56	8.06	0.76	0.42	6.54	0.69	0.58	5.60	0.79
LSTM	Wet	0.75	23.19	0.86	0.71	49.29	0.84	0.74	20.47	0.87
	Dry	0.72	6.44	0.86	0.45	6.38	0.83	0.39	6.76	0.78

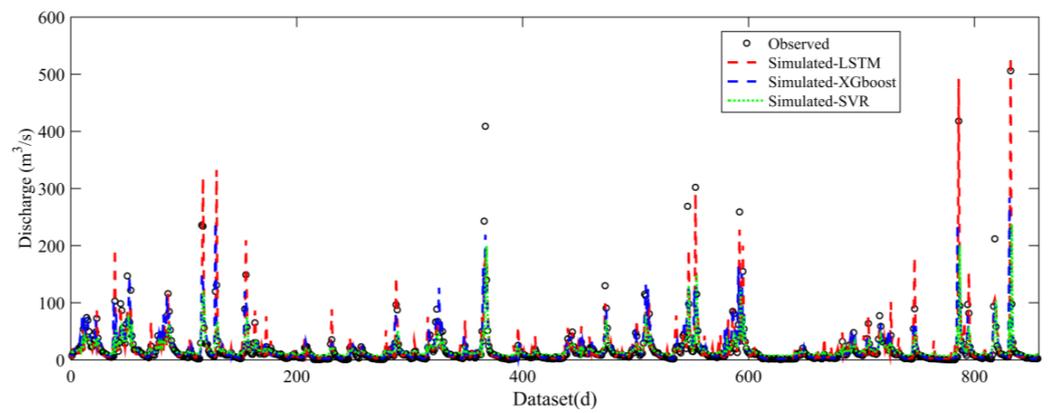


Figure 8. Time series plots for observed and simulated streamflow during testing wet periods.

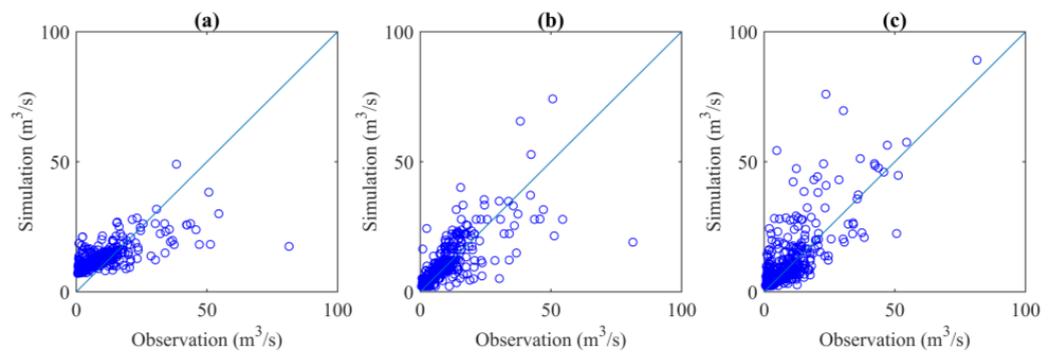


Figure 9. Scatter plots between observations and simulations of discharge during testing dry periods for (a) SVR; (b) XGBoost; and (c) LSTM.

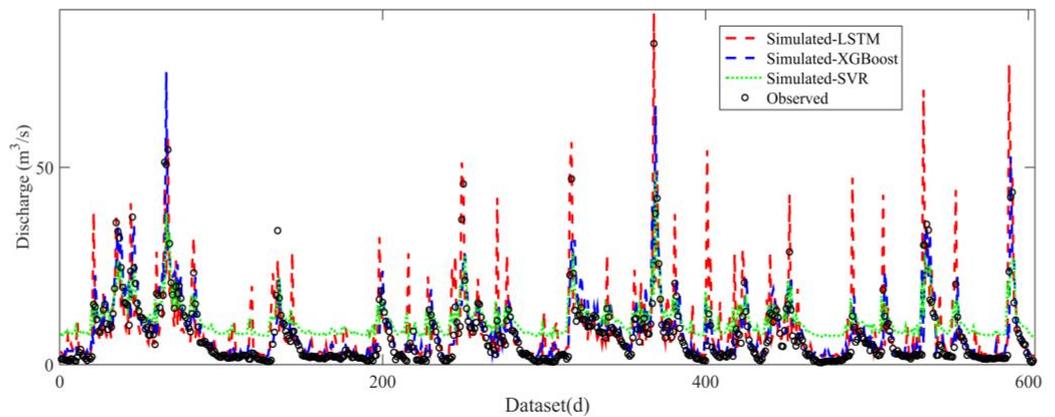


Figure 10. Time series plots for observed and simulated streamflow during testing dry periods.

3.4. Classification of Wet and Dry Seasons for Simulation

Figure 11 shows the performances of the LSTM and XGBoost models with input scenario V trained with different datasets classified by wet and dry seasons, respectively. Moreover, all simulated data combined with wet and dry seasons were assessed according to the models trained with distinct datasets. Similar with the models trained with all datasets (Table 4), XGBoost trained with datasets during dry seasons and LSTM trained with datasets during wet seasons always obtained higher accuracy than XGBoost for wet seasons and LSTM for dry seasons, respectively, in terms of NSE and CC. This may be owed to the different model structures between artificial networks and decision trees methods. The classification of different datasets did not improve the simulation accuracy of the XGBoost models, basically. Nevertheless, the LSTM models trained with different

datasets were better than the LSTM model trained with all datasets, especially during the dry seasons. Although the classification of the datasets reduced the amount of data for training, the improved performance suggests that the distinction of various rainfall–runoff processes is beneficial for LSTM modeling.

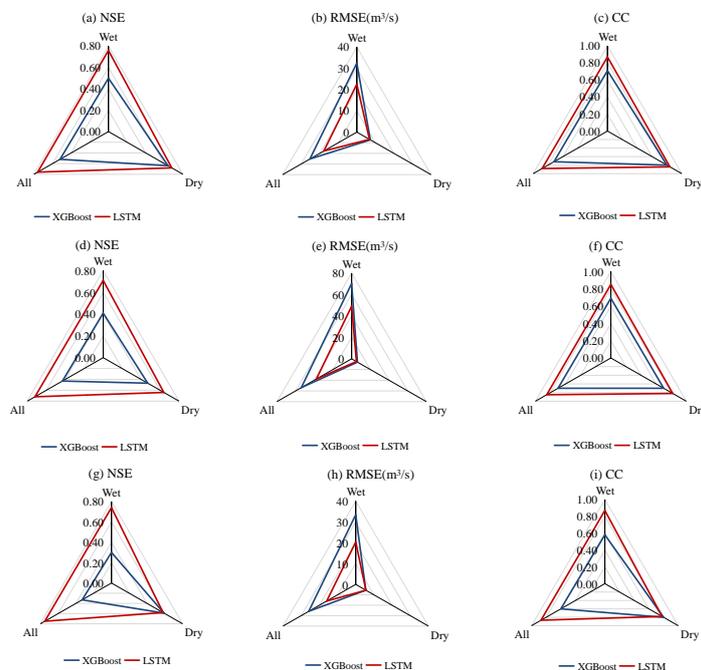


Figure 11. Comparison of LSTM and XGBoost models trained with different datasets (a–c) for training, (d–f) for validation, and (g–i) for testing.

Moreover, two models were selected to be analyzed with the JMS method. The first model was the LSTM model trained with data from all years. The second model was a hybrid model which was combined by the LSTM model trained with wet-season data and the XGBoost model trained with dry-season data. The simulated runoff series of both models passed the verification of multifractality within the range of the temporal resolution of 1–8 days, which was narrower than that of the lumped and distributed models (1–16 days) [40], reflecting that both models performed worse than physically based models do in simulating the long-term autocorrelation of runoff.

Figure 12 displays the multifractal spectra of the runoff series of two models. Table 5 shows the metrics of JMS. Generally, the hybrid model had better spectra for being closer to a 45° line for training, validation, and testing. All spectra indicated that the models performed well in high-flow simulation and performed unsatisfactorily in low-flow simulation, which agrees with the metrics presented in Table 4 and Figure 11. The spectra of the hybrid model for testing were wide. This indicates that the hybrid model was able to catch the magnitude and overall trend, but failed to simulate the fluctuations accurately. All spectra were in the upper-left part, reflecting the overestimation of both models, especially of the LSTM model, for wet seasons.

Figure 13 displays the detailed multifractal spectra of the runoff series of two models with selected $q_{[0]}$ and $q_{[5]}$. The structures of the spectra of the two models for training and validation were similar and indicated satisfactory performances. The multifractal spectra of the hybrid model for testing illustrated the weakness in simply combining the runoff series of two models. Specifically, due to the small area and meteorological features of the ARNT catchment, the runoff was sensitive to rainfall events, and the regression of runoff was similar for wet and dry seasons. By providing totally different simulations of hydrological processes, the hybrid model failed to reproduce consistent fractal characteristics.

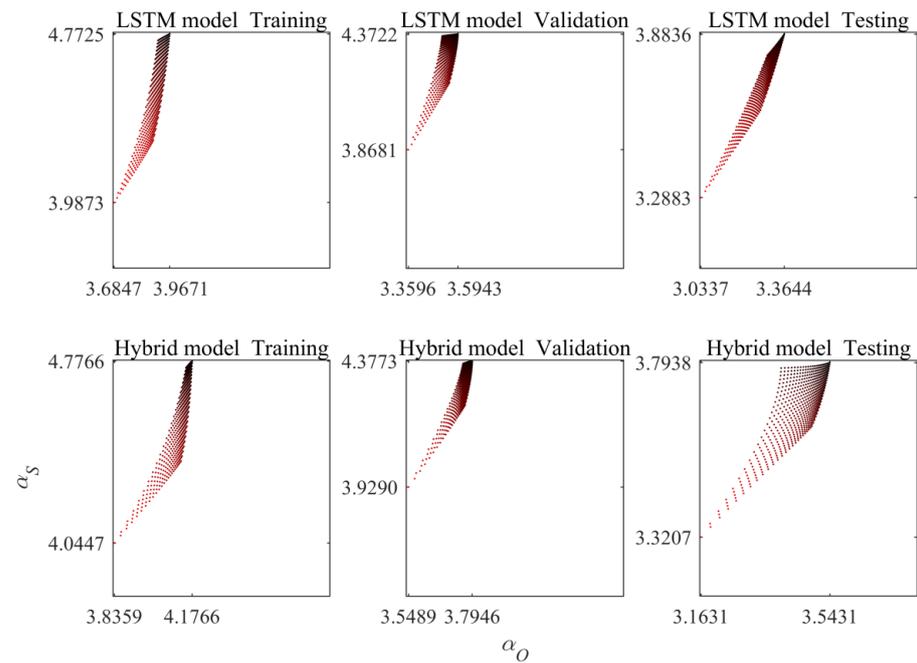


Figure 12. Joint multifractal spectrum of observed and simulated runoff series of two models.

Table 5. Metrics of the joint multifractal spectrum of observed and simulated runoff series in three cases, including the slopes (k) of the fitted line of JMS and the correlation coefficients (r^2) of JMS's $\alpha_{[O]}$ and $\alpha_{[S]}$ calculated from the fitted second-order polynomial used to represent the width of JMS.

k/r^2	Training	Validation	Testing
LSTM	3.29/0.79	1.97/0.64	1.90/0.93
Hybrid model	2.48/0.76	1.94/0.81	1.06/0.65

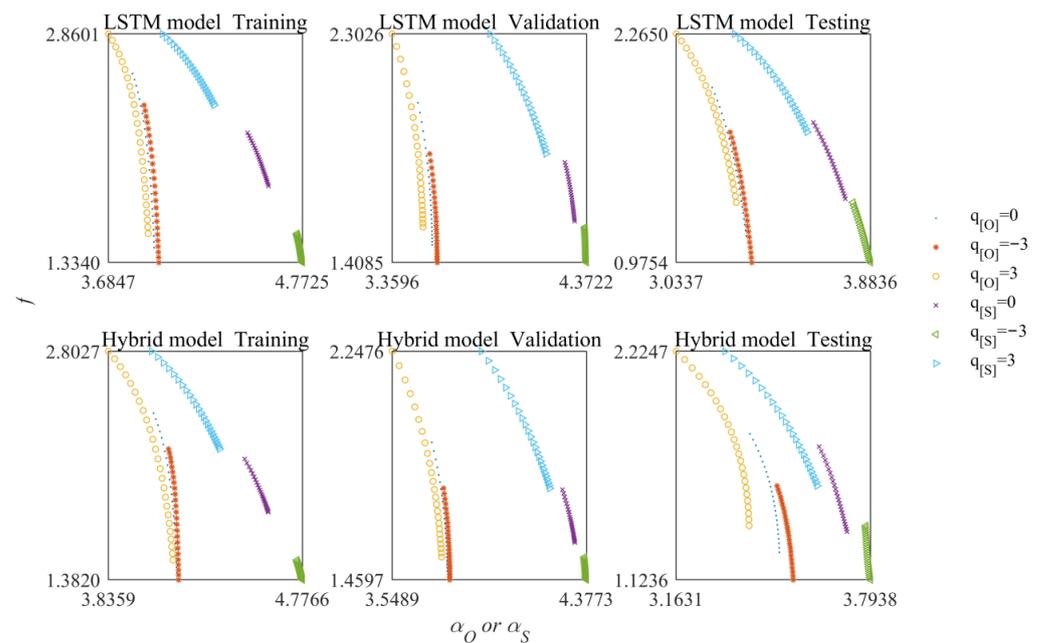


Figure 13. Joint multifractal spectrum of observed and simulated runoff series of two models with selected $q_{[O]}$ and $q_{[S]}$.

4. Conclusions

This study compared three machine learning methods, SVR, XGBoost, and LSTM, for daily runoff simulation in the north tributary of the Ao River (ARNT) catchment, China. Three evaluation criteria, namely NSE, RMSE, and CC, were selected. Five input scenarios, including spatial or areal rainfall, and antecedent runoff, were fed to these models to analyze their sensitivities to runoff fluctuations. Datasets were also divided according to wet and dry seasons for assessing their performances in different rainfall–runoff mechanisms. Furthermore, the JMS method was implemented to analyze the performances of the best models trained with all datasets, and distinct wet and dry season datasets, thoroughly. Several detailed conclusions can be drawn as follows:

- (1) The performance of LSTM models was always better than that of XGBoost, followed by that of SVR. The models with a gauged rainfall and antecedent streamflow input scenario obtained the best accuracy, indicating the roles of the spatial distribution of rainfall and antecedent water storage on streamflow fluctuations.
- (2) The impacts of input variables were different for SVR, XGBoost, and LSTM. The LSTM with only rainfall information as an input, and the XGBoost and SVR models with only antecedent streamflow as an input, performed much better than the LSTM model with only antecedent streamflow as an input and the XGBoost and SVR models with only rainfall information, respectively.
- (3) Although LSTM always yielded better performances, XGBoost showed relatively high accuracy compared with LSTM during dry seasons when trained with all datasets. Moreover, the classification of datasets according to wet and dry seasons improved the performances of LSTM especially for dry seasons. This suggests that different rainfall–runoff mechanisms dominated the runoff processes during wet and dry seasons.
- (4) The LSTM and a hybrid model were analyzed with the JMS method. Overall, the hybrid model outperformed the LSTM model. However, the fractal characteristics of the hybrid model were not consistent throughout the simulation period.

Author Contributions: Conceptualization, methodology, and writing, R.H. and Z.B. All authors have read and agreed to the published version of the manuscript.

Funding: Natural Science Research Project of Anhui Educational Committee (2022AH050832), the Zhejiang Natural Science Foundation (LZJWY22D010001), the Academician Workstation in Anhui Province, Anhui University of Science and Technology (2022-AWAP-06), and Scientific Research Foundation for High-Level Talents of Anhui University of Science and Technology (13190207).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Valizadeh, N.; Mirzaei, M.; Allawi, M.F.; Afan, H.A.; Mohd, N.S.; Hussain, A.; El-Shafie, A. Artificial intelligence and geo-statistical models for stream-flow forecasting in ungauged stations: State of the art. *Nat. Hazards* **2017**, *86*, 1377–1392. [[CrossRef](#)]
2. Yang, M.; Wang, H.; Jiang, Y.; Lu, X.; Xu, Z.; Sun, G. GECA proposed ensemble-KNN method for improved monthly runoff forecasting. *Water Resour. Manag.* **2020**, *34*, 849–863. [[CrossRef](#)]
3. Yassin, M.; Asfaw, A.; Speight, V.; Shucksmith, J.D. Evaluation of Data-Driven and Process-Based Real-Time Flow Forecasting Techniques for Informing Operation of Surface Water Abstraction. *J. Water Resour. Plan. Manag.* **2021**, *147*, 04021037. [[CrossRef](#)]
4. Blöschl, G.; Hall, J.; Viglione, A.; Perdigão, R.A.; Parajka, J.; Merz, B.; Lun, D.; Arheimer, B.; Aronica, G.T.; Bilibashi, A.; et al. Changing climate both increases and decreases European river floods. *Nature* **2019**, *573*, 108–111. [[CrossRef](#)]
5. Lei, X.; Gao, L.; Wei, J.; Ma, M.; Xu, L.; Fan, H.; Li, X.; Gao, J.; Dang, H.; Chen, X.; et al. Contributions of climate change and human activities to runoff variations in the Poyang Lake Basin of China. *Phys. Chem. Earth* **2021**, *123*, 103019. [[CrossRef](#)]
6. Yeditha, P.K.; Kasi, V.; Rathinasamy, M.; Agarwal, A. Forecasting of extreme flood events using different satellite precipitation products and wavelet-based machine learning methods. *Chaos* **2020**, *30*, 63115. [[CrossRef](#)]

7. Xiang, Z.; Jun, Y.; Demir, I. A rainfall-runoff model with LSTM-based sequence-to-sequence learning. *Water Resour. Res.* **2020**, *56*, e2019WR025326. [[CrossRef](#)]
8. Nearing, G.S.; Kratzert, F.; Sampson, A.K.; Pelissier, C.S.; Klotz, D.; Frame, J.M.; Prieto, C.; Gupta, H.V. What role does hydrological science play in the age of machine learning? *Water Resour. Res.* **2021**, *57*, e2020WR028091. [[CrossRef](#)]
9. Mosavi, A.; Ozturk, P.; Chau, K.-w. Flood prediction using machine learning models: Literature review. *Water* **2018**, *10*, 1536. [[CrossRef](#)]
10. Hamitouche, M.; Molina, J. A review of ai methods for the prediction of high-flow extremal hydrology. *Water Resour. Manag.* **2022**, *36*, 3859–3876. [[CrossRef](#)]
11. 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](#)]
12. Li, Y.; Wei, J.; Wang, D.; Li, B.; Huang, H.; Xu, B.; Xu, Y. A Medium and Long-Term Runoff Forecast Method Based on Massive Meteorological Data and Machine Learning Algorithms. *Water* **2021**, *13*, 1308. [[CrossRef](#)]
13. Liu, J.; Ren, K.; Ming, T.; Qu, J.; Guo, W.; Li, H. Investigating the effects of local weather, streamflow lag, and global climate information on 1-month-ahead streamflow forecasting by using XGBoost and SHAP: Two case studies involving the contiguous USA. *Acta Geophys.* **2022**, *71*, 905–925. [[CrossRef](#)]
14. 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](#)]
15. Le, X.; Nguyen, D.; Jung, S.; Yeon, M.; Lee, G. Comparison of deep learning techniques for river streamflow forecasting. *IEEE Access* **2021**, *9*, 71805–71820. [[CrossRef](#)]
16. Rahimzad, M.; Nia, A.M.; Zolfonoon, H.; Soltani, J.; Mehr, A.D.; Kwon, H. Performance comparison of an LSTM-based deep learning model versus conventional machine learning algorithms for streamflow forecasting. *Water Resour. Manag.* **2021**, *35*, 4167–4187. [[CrossRef](#)]
17. Yeditha, P.K.; Rathinasamy, M.; Neelamsetty, S.S.; Bhattacharya, B.; Agarwal, A. Investigation of satellite rainfall-driven rainfall-runoff model using deep learning approaches in two different catchments of India. *J. Hydroinform.* **2022**, *24*, 16–37. [[CrossRef](#)]
18. Feng, D.; Fang, K.; Shen, C. Enhancing streamflow forecast and extracting insights using Long-Short Term Memory Networks with data integration at continental scales. *Water Resour. Res.* **2020**, *56*, e2019WR026793. [[CrossRef](#)]
19. Kim, T.; Yang, T.; Gao, S.; Zhang, L.; Ding, Z.; Wen, X.; Gourley, J.J.; Hong, Y. Can artificial intelligence and data-driven machine learning models match or even replace process-driven hydrologic models for streamflow simulation?: A case study of four watersheds with different hydro-climatic regions across the CONUS. *J. Hydrol.* **2021**, *598*, 126423. [[CrossRef](#)]
20. Moosavi, V.; Fard, Z.G.; Vafakhah, M. Which one is more important in daily runoff forecasting using data driven models: Input data, model type, preprocessing or data length? *J. Hydrol.* **2022**, *606*, 127429. [[CrossRef](#)]
21. Niu, W.; Feng, Z. Evaluating the performances of several artificial intelligence methods in forecasting daily streamflow time series for sustainable water resources management. *Sust. Cities Soc.* **2021**, *64*, 102562. [[CrossRef](#)]
22. 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](#)]
23. Chang, W.; Chen, X. Monthly rainfall-runoff modeling at watershed scale: A comparative study of data-driven and theory-driven approaches. *Water* **2018**, *10*, 1116. [[CrossRef](#)]
24. Xiong, J.; Wang, Z.; Guo, S.; Wu, X.; Yin, J.; Wang, J.; Lai, C.; Gong, Q. High effectiveness of GRACE data in daily-scale food modeling: Case study in the Xijiang River Basin, China. *Nat. Hazards* **2022**, *113*, 507–526. [[CrossRef](#)]
25. Emerton, R.E.; Stephens, E.M.; Cloke, H.L. What is the most useful approach for forecasting hydrological extremes during El Niño? *Environ. Res. Commun.* **2019**, *1*, 031002. [[CrossRef](#)]
26. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer: New York, NY, USA, 1995.
27. Liu, Z.; Zhou, P.; Chen, G.; Guo, L. Evaluating a coupled discrete wavelet transform and support vector regression for daily and monthly streamflow forecasting. *J. Hydrol.* **2014**, *519*, 2822–2831. [[CrossRef](#)]
28. 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. [[CrossRef](#)]
29. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [[CrossRef](#)]
30. 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 (KDD '16), San Francisco, CA, USA, 13–17 August 2016.
31. Ko, C.-M.; Jeong, Y.Y.; Lee, Y.-M.; Kim, B.-S. The development of a quantitative precipitation forecast correction technique based on machine learning for hydrological applications. *Atmosphere* **2020**, *11*, 111. [[CrossRef](#)]
32. Potdar, A.S.; Kirstetter, P.; Woods, D.; Saharia, M. Toward predicting flood event peak discharge in ungauged basins by learning universal hydrological behaviors with machine learning. *J. Hydrometeorol.* **2021**, *22*, 2971–2982.
33. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
34. Le, X.-H.; Ho, H.V.; Lee, G.; Jung, S. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water* **2019**, *11*, 1387. [[CrossRef](#)]
35. Brochu, E.; Cora, V.M.; de Freitas, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv* **2010**, arXiv:1012.2599.

36. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* **2016**, *104*, 148–175. [[CrossRef](#)]
37. Zhang, K.; Zheng, L.; Liu, Z.; Jia, N. A deep learning based multitask model for network-wide traffic speed prediction. *Neurocomputing* **2020**, *396*, 438–450. [[CrossRef](#)]
38. Alizadeh, B.; Bafti, A.G.; Kamangir, H.; Zhang, Y.; Wright, D.B.; Franz, K.J. A novel attention-based LSTM cell post-processor coupled with bayesian optimization for streamflow prediction. *J. Hydrol.* **2021**, *601*, 126526. [[CrossRef](#)]
39. Nash, J.E.; Sutcliffe, J.V. River flow forecasting through conceptual models, part 1: A discussion of principles. *J. Hydrol.* **1970**, *10*, 282–290. [[CrossRef](#)]
40. Bai, Z.; Xu, X.-P.; Pan, S.; Liu, L.; Wang, Z.X. Evaluating the performance of hydrological models with joint multifractal spectra. *Hydrol. Sci. J.* **2022**, *67*, 1771–1789. [[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.