water

MDPI

Article

Combining Multiple Machine Learning Methods Based on CARS Algorithm to Implement Runoff Simulation

Yuyan Fan [1], Xiaodi Fu [2], Guangyuan Kan [3,*], Ke Liang [4] and Haijun Yu [3]

[1] Department of Water Resources Strategy, Institute of Mineral Resources, Chinese Academy of Geological Sciences, Beijing 100037, China; fyyscu@126.com
[2] Beijing Engineering Corporation Limited, Beijing 100024, China; fuxiaodi2006@163.com
[3] State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin, Research Center on Flood & Drought Disaster Prevention and Reduction of the Ministry of Water Resources, China Institute of Water Resources and Hydropower Research, Beijing 100038, China; yuhaijun@iwhr.com
[4] Beijing IWHR Corporation, Beijing 100048, China; liangkepapers@126.com
* Correspondence: kangy@iwhr.com

Abstract: Runoff forecasting is crucial for water resource management and flood safety and remains a central research topic in hydrology. Recent advancements in machine learning provide novel approaches for predicting runoff. This study employs the Competitive Adaptive Reweighted Sampling (CARS) algorithm to integrate various machine learning models into a data-driven rainfall–runoff simulation model. We compare the forecasting performance of different machine learning models to improve rainfall–runoff prediction accuracy. This study uses data from the Maduwang hydrological station in the Bahe river basin, which contain 12 measured flood events from 2000 to 2010. Historical runoff and areal mean rainfall serve as model inputs, while flood data at different lead times are used as model outputs. Among the 12 flood events, 9 are used as the training set, 2 as the validation set, and 1 as the testing set. The results indicate that the CARS-based machine learning model effectively forecasts floods in the Bahe River basin. Under the prediction period of 1 to 6 h, the model achieves high forecasting accuracy, with the average NSE ranging from 0.7509 to 0.9671 and the average R^2 ranging from 0.8397 to 0.9413, though the accuracy declines to some extent as the lead time increases. The model accurately predicts peak flow and performs well in forecasting high flow and recession flows, though peak flows are somewhat underestimated for longer lead times. Compared to other machine learning models, the SVR model has the highest average RMSE of 0.942 for a 1–6 h prediction period. It exhibits the smallest deviation among low-, medium-, and high-flow curves, with the lowest NRMSE values across training, validation, and test sets, demonstrating better simulation performance and generalization capability. Therefore, the machine learning model based on CARS feature selection can serve as an effective method for flood forecasting. The related findings provide a new forecasting method and scientific decision-making basis for basin flood safety.

Keywords: artificial intelligence; machine learning; rainfall–runoff modeling; flood forecasting; CARS algorithm

check for updates

Citation: Fan, Y.; Fu, X.; Kan, G.; Liang, K.; Yu, H. Combining Multiple Machine Learning Methods Based on CARS Algorithm to Implement Runoff Simulation. *Water* **2024**, *16*, 2397. https://doi.org/10.3390/w16172397

Academic Editor: Enedir Ghisi

Received: 18 July 2024
Revised: 4 August 2024
Accepted: 20 August 2024
Published: 26 August 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 runoff forecasting plays a crucial role in mitigating the threats of natural disasters and protecting the safety of people's lives and property. Artificial intelligence-based flood forecasting is an important technological measure to improve flood forecast accuracy, enhancing flood control efficiency and assisting flood disaster defense [1].

Runoff prediction models can be categorized into process-driven and data-driven models [2,3]. Process-driven models forecast future runoff changes based on physical mechanisms by studying the physical processes of the hydrological cycle. Typical models include conceptual hydrological models such as the Xinanjiang model and the Sacramento model, as well as distributed hydrological models such as TOPMODEL [4], VIC [5], and SWAT [6,7].

Although process-driven models have strong physical mechanisms and interpretability, they involve large computations and numerous parameters and require various types of data for model calibration. Parameter tuning is difficult for process-driven models, and their computational efficiency is poor, often resulting in unsatisfactory forecast accuracy, it is necessary to use multiple rainfall events simultaneously for parameter calibration to achieve a more comprehensive system analysis [8]. On the other hand, data-driven models treat runoff as a time series and achieve high-precision predictions by learning the functional relationships between input variables and runoff to describe potential connections between inputs and outputs. These models do not need to consider the actual physical meaning of the hydrological cycle and can achieve good simulation results in data-rich regions.

Data-driven runoff prediction models exploit complex nonlinear relationships between input and output data. By utilizing vast amounts of hydrological data, these models establish high-precision, high-timeliness machine learning models to process unknown samples, thereby achieving the goal of simulating and predicting runoff. Flood simulation and prediction models based on machine learning algorithms can bypass constraints imposed with generalized and assumed hydrological processes of a basin. They also have strong capabilities for discovering the basin's runoff generation and convergence patterns and are easy to implement with efficient and fast computations, significantly enhancing the timeliness of flood simulation and prediction. Commonly used machine learning methods for flood simulation include Regression Analysis (RA), Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), Recurrent Neural Networks (RNNs), Decision Trees (DTs), and Random Forests (RFs). Based on the dimensions of model input factors, runoff prediction models are primarily divided into single-factor driven models [9] based on runoff autoregression and multi-factor-driven models [10] based on hydro-meteorological inputs. Valipour [11] used a Seasonal Autoregressive Integrated Moving Average (SARIMA) model to predict annual runoff for various states of the United States. Duan Yong et al. [12] constructed an improved Backpropagation (BP) neural network model for daily runoff prediction at hydrological stations in the middle reaches of the Yellow river, achieving better prediction accuracy compared to autoregressive models. Multi-factor-driven runoff prediction models start from physical causes, using monitoring data such as precipitation, evaporation, and temperature as inputs and employing machine learning algorithms to predict runoff. Kratzert et al. [13] used an LSTM model to simulate characteristics of 241 catchments in the United States, demonstrating that the LSTM model could effectively represent the correlations and variations between rainfall and runoff sequences, thus better predicting river runoff processes. Chiang et al. [14] constructed an RNN-based multi-step runoff prediction model based on hydrometeorological and numerical weather prediction information, achieving good prediction accuracy within a lead time range of 1–6 h. Kan et al. [15] considered rainfall and antecedent runoff as model input factors, taking into account factors influencing runoff such as rainfall and the wetness of the underlying surface and obtained satisfactory runoff prediction results.

Currently, research on hydrological forecasting based on machine learning primarily focuses on optimizing parameters [16] and frameworks [17] to improve runoff prediction accuracy. However, for multi-factor-driven runoff prediction models, there is still insufficient research on the impact of the lead time of rainfall and antecedent runoff on forecast accuracy and stability. If the order of rainfall and antecedent runoff is too small, it will limit the amount of input information, leading to reduced forecast accuracy, whereas if the order is too large, it may introduce redundant information, resulting in overfitting. Additionally, the complexity, prediction stability, and uncertainty of machine learning runoff models vary under different lead times. Moreover, intercomparison of different machine learning methods is also important for model selection. Therefore, in-depth research into the above-mentioned topics can help select the most suitable model and settings for runoff prediction, thereby enhancing forecast accuracy.

In previous comparative evaluations of different data-driven rainfall–runoff models, the selection of model types was limited and lacked representativeness. The prediction periods of the models were generally short, often using single time steps, and evaluation metrics predominantly focused on overall accuracy, without sufficiently considering aspects such as varying flow conditions and generalization ability. This study aims to explore the optimal modeling approach for machine learning models in data-driven flood runoff forecasting, as well as performance evaluation under different lead times. The research employs Gradient Boosting Decision Tree (GBDT) models (including CatBoost, XgBoost, and LightGBM), neural network models (including LSTM and multilayer BP), and shallow machine learning models (including KNN and SVR) to construct flood runoff prediction models. Due to the complexity and parameterization difficulties of physical models in the Ba River Ma Du Wang sub-basin, accurate rainfall–runoff simulation may be challenging to achieve. Data-driven models, by learning patterns from historical data, can compensate for the shortcomings of traditional physical models. Validating the applicability and performance of data-driven models in this region can promote their application in other similar basins. By comparing the runoff prediction results of different machine learning models for 1 to 6 h forecasting periods in the Ba River Basin, the performance of various data-driven rainfall–runoff models is evaluated based on overall accuracy, event-based flood accuracy, the percentage of low-, medium-, and high-flow duration curves, and generalization ability. This study aims to achieve accurate prediction and timely warning of flood patterns. The results of this research are expected to significantly enhance the reliability of flood risk assessment, provide scientific basis for watershed engineering planning, and optimize the performance of forecasting and warning systems. Additionally, this study will support engineering scheduling and emergency management, contributing to the development of more effective flood response strategies. Through this research, we anticipate making substantial progress in flood management and risk control, offering robust data support and decision-making tools for relevant departments and decision-makers.

2. Methodology

2.1. Runoff Forecasting Models

Considering rainfall and antecedent runoff as the primary driving factors [18,19], using measured antecedent flow, rainfall, and future rainfall as model inputs can achieve single-step or multi-step forecasting of watershed outlet runoff processes. The model is constructed as follows:

$$\left[Q_{t+T}^{Forecast}, \ldots, Q_{t+1}^{Forecast}, Q_{t}^{Forecast}\right] = \mathrm{F}\left(P_t, \ldots, P_{t-np}, Q_{t-1}^{obs}, \ldots, Q_{t-nq}^{obs}\right) \quad (1)$$

where $Q_t^{Forecast}$ represents the predicted watershed outlet flow at time t for one lead time step; $Q_{t+T}^{Forecast}$ represents the predicted watershed outlet flow at time $t + T$ for T lead time steps; F denotes the input–output mapping function implemented via the machine learning models used in this study, including CatBoost, XgBoost, LightGBM, LSTM, multilayer BP, KNN, and SVR; np and nq are integers representing the orders of antecedent rainfall and antecedent flow in the rainfall–runoff relationship, respectively.

2.2. Feature Selection Algorithm Based on CARS

The Competitive Adaptive Reweighted Sampling (CARS) algorithm is a feature selection method that combines Monte Carlo sampling with the regression coefficients of Partial Least Squares (PLS) models [20]. Inspired by Darwin's theory of "survival of the fittest", the algorithm employs an Adaptive Reweighted Sampling (ARS) strategy. After each sampling iteration, the results are retained in the PLS model, and sample points with larger absolute regression coefficient weights are selected as new subsets, while points with smaller weights are removed. Subsequently, a new PLS model is constructed based on this updated subset. Through iterative calculations, the algorithm selects the spectral

bands with the minimum Root Mean Squared Error of Cross-Validation (RMSECV) from the subset with the lowest RMSECV in the PLS model.

The feature selection algorithm based on CARS is widely applied in chemometrics and machine learning, known for its high accuracy and minimal redundancy [21,22]. In data-driven rainfall–runoff modeling, the CARS feature selection algorithm is used to identify the number of lead times for rainfall and antecedent runoff. Initially, based on analysis of the characteristics of historical rainfall–runoff data in the watershed, the lag phases from rainfall centroid to peak flow are estimated. A slightly larger number than this lag period is chosen as the upper limit for the number of lead times for rainfall and antecedent runoff. Following the modeling approach outlined in Section 2.1, candidate input variables are used as inputs, with the corresponding runoff at each time step as the outputs. Iterative calculations using the CARS algorithm's feature selection method are then applied to determine the optimal number of lead times for the candidate input variables.

2.3. Machine Learning Models

2.3.1. Gradient Boosting Decision Tree (GBDT) Model

CatBoost [23], XGBoost [24], and LightGBM are three widely used Gradient Boosting Decision Tree (GBDT) models in machine learning tasks. CatBoost (Categorical Boosting) is a GBDT algorithm developed by Yandex, designed specifically to handle categorical features and to reduce overfitting. XGBoost (Extreme Gradient Boosting) is an optimized GBDT algorithm known for its speed and performance. LightGBM (Light Gradient Boosting Machine) is another GBDT algorithm developed by Microsoft (Redmond, WA, USA), particularly suited for large-scale datasets.

2.3.2. Neural Network Models

The Backpropagation neural network (BP) and Long Short-Term Memory (LSTM) model are widely used neural network models. The BP neural network exhibits good adaptability and nonlinear approximation capabilities, effectively addressing highly nonlinear and dynamically uncertain issues in rainfall–runoff modeling. The LSTM model utilizes memory cell units composed of forget, input, and output gates, providing the model with strong long-term memory capabilities and overcoming the issues of gradient vanishing or exploding present in traditional neural networks [25]. As a type of multi-layer feedforward recurrent neural network composed of neurons, LSTM demonstrates significant structural advantages in handling time series data.

2.3.3. Shallow Machine Learning Models

Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) models are widely used shallow machine learning models due to their simplicity [26], interpretability, low training costs, and high computational efficiency. SVR, as a regression algorithm based on SVM, is built on the principles of risk minimization and VC dimension theory to solve regression problems. Through kernel functions, SVR transforms nonlinear low-dimensional functions into linear high-dimensional functions for regression [27].

The core idea of the KNN algorithm is to calculate the distance between new samples and training samples, find the nearest k neighbors, and predict based on the labels of these neighbors.

SVR can be expressed as follows:

$$f(x) = \sum_{i=1}^{n} (\hat{a}_l - a_i) g(x, x_i) + b \quad (2)$$

In the above equation, $\hat{a}_l$ and a_i are non-negative parameters known as Lagrange multipliers; $g(x, x_i)$ represents the kernel function; b is the model parameter which needs calibration.

3. Study Area and Data Description

3.1. Study Area

The Ba river, historically known as Zishui, is a major tributary on the right bank of the Wei river system, located in the southeastern part of Xi'an City [28], Shaanxi Province, China (see in Figure 1). It spans longitudinally from 109°00′ E to 109°47′ E and latitudinally from 33°50′ N to 34°27′ N. Originating from Jiudaogou in Jiandong, Bayuan township, Lantian county, the river flows through Lantian county, Baqiao district, Chanba ecological district and the international port area, with a total length of 104 km and an average gradient of 6‰. The basin covers an area of 2581 km^2. The region features a semi-humid continental monsoon climate with an annual rainfall ranging from 550 to 900 mm, characterized by highly uneven distribution throughout the year and significant interannual variability. The average annual evaporation is approximately 776 mm. The rainfall pattern results in uneven annual distribution of runoff, significant interannual variability, and pronounced seasonal variations. From July to October, the total runoff accounts for about 57% of the annual total, amounting to 718 million m^3 annually. The Maduwang hydrological station serves as a crucial control station for the Ba river basin [29].

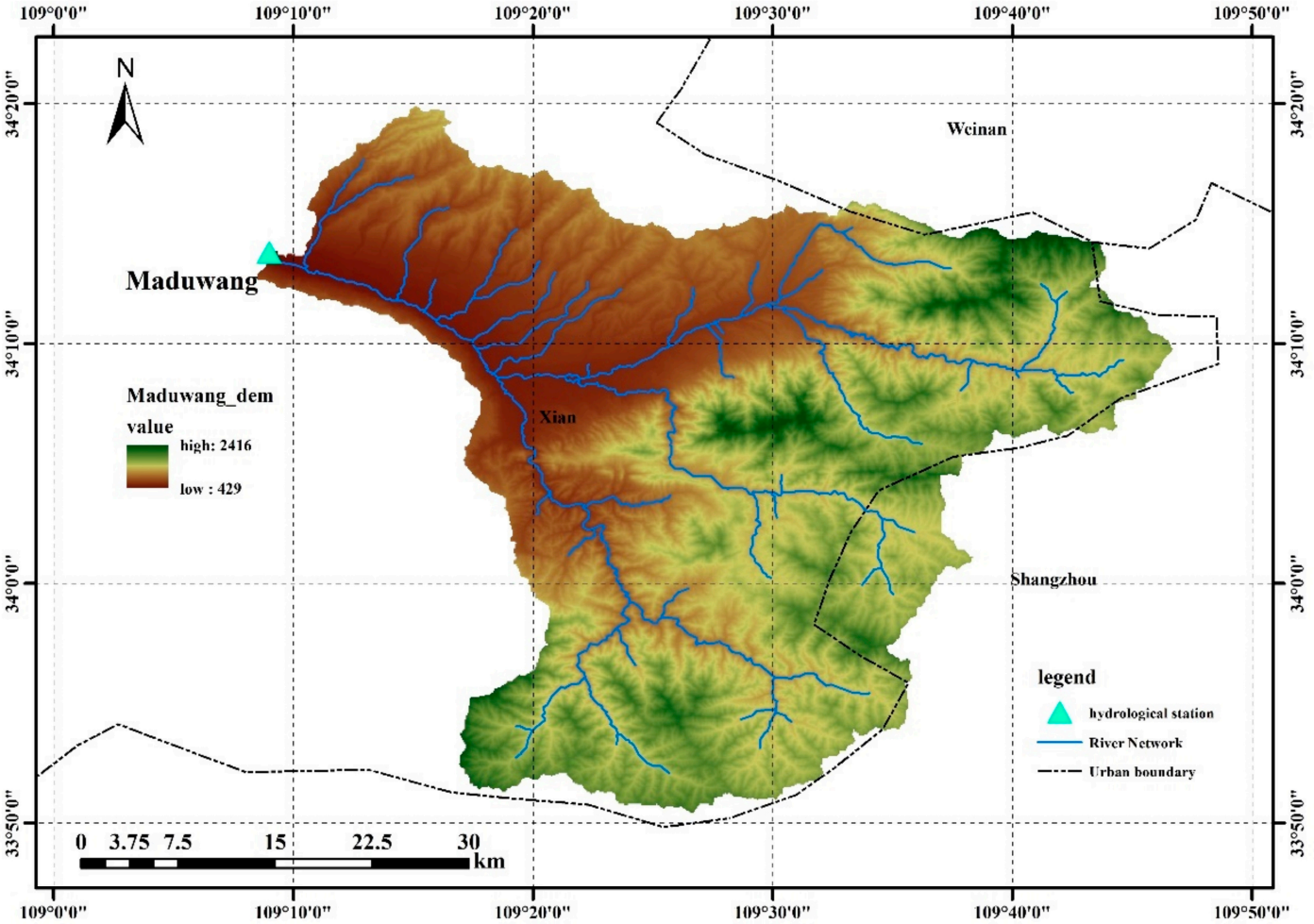


Figure 1. Digital elevation and location map of the Maduwang basin.

3.2. Data Description

Given the difficulty of obtaining observational data, this paper selects 12 flood events from the Maduwang hydrological station in the Ba river basin during the period from 2000 to 2010 for flood simulation and forecasting. The time interval of the observed data is 1 h, The historical continuous year method was applied, where 9 flood events from 2000 to

2007 are used as the training set, 2 flood events from 2008 to 2009 as the validation set, and the flood event from 2010 as the testing set. The peak flow distribution of flood events is shown in Figure 2, which shows that the validation periods include both large and small floods, while the test period consists of a large flood. The peak flow values in the training set adequately reflect the performance of the data-driven model, this indicates that the representativeness of the validation and testing sets meets the requirements.

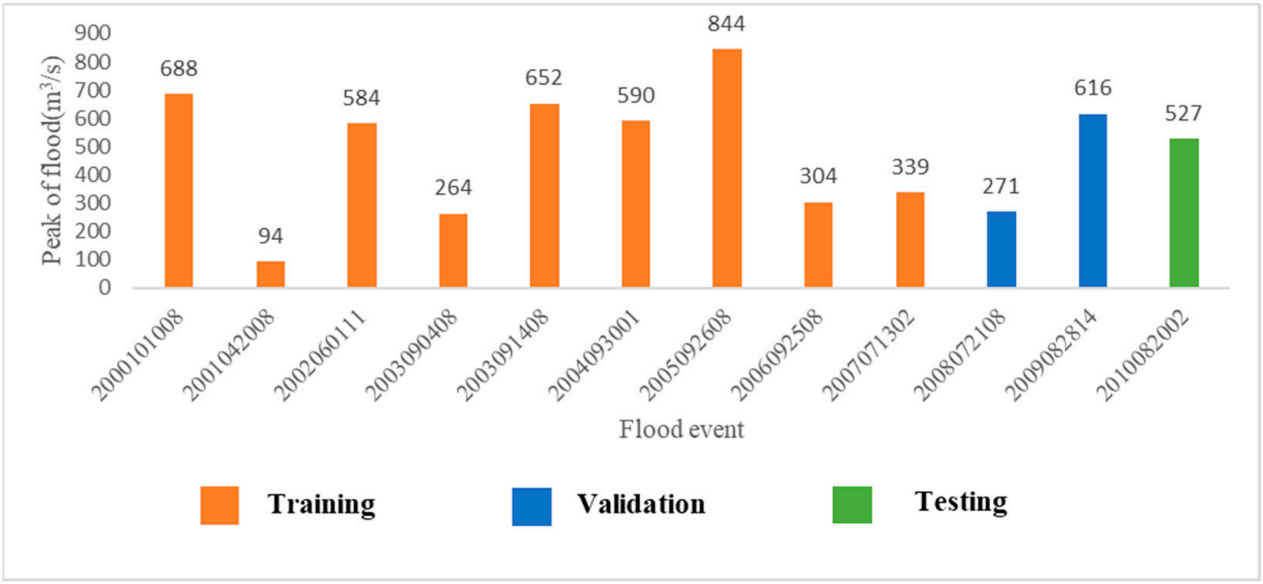


Figure 2. Flood event peak flow distribution.

Due to the large difference in magnitude between rainfall and runoff data, a linear function-based normalization method is employed to reduce the impact of these differences on the weights of runoff input variables. The model output results are denormalized and then compared with the observed data.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3}$$

In the above equation: x' denotes the normalized data, x represents the observed data; x_{max} and x_{min}, respectively, denote the maximum and minimum values in the observed data.

4. Data-Driven Runoff Forecasting Model Based on CARS

4.1. Identification of Model Order

Based on the lag periods between rainfall centroids and peak flows observed in historical flood events, the upper limit for lag periods of rainfall and antecedent runoff is determined to be 20. Rainfall and antecedent runoff series data within this limit are fed into the CARS algorithm. After 45 iterations, the minimum RMSECV is found to be 11.1. The selection process of rainfall and antecedent runoff order based on the CARS algorithm is shown in Figure 3. The rainfall and runoff series selected via CARS are used as input variables for the subsequent models.

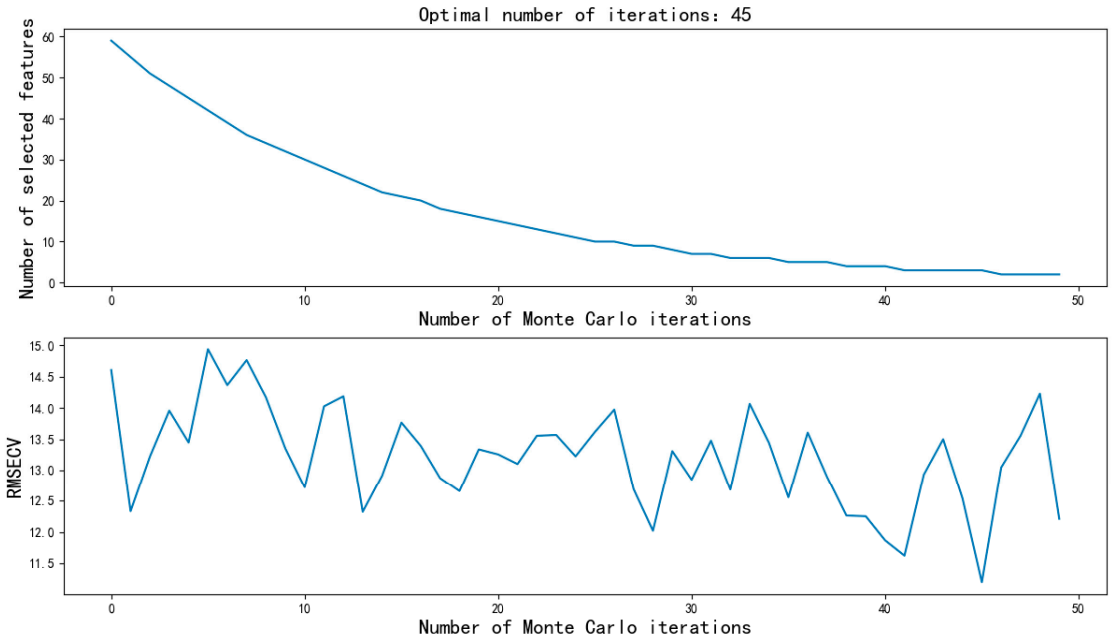


Figure 3. Selection process of rainfall and antecedent runoff order based on the CARS algorithm.

4.2. Settings of Model Parameter

This study constructs a runoff forecasting model using machine learning models based on automated machine learning principles. The hyperparameters that are optimized for each model and their optimization ranges are set using the Bayesian Optimization Algorithm (BOA) [30], combined with a five-fold cross-validation framework for automatic hyperparameter optimization. The Bayesian Optimization Algorithm [31] is grounded in Bayesian theorem, maximizing the acquisition function to find the next most promising evaluation point x_i, evaluating the objective function value y_i, and adding the newly obtained pair (x_i, y_i) to the known evaluation point set. The probability surrogate model is updated iteratively to achieve the optimal solution. Due to its ability to effectively utilize historical information, BOA's efficiency is significantly higher compared to other optimization methods. In this paper, the BOA framework is used to optimize the hyperparameters of the CatBoost, XGBoost, LightGBM, LSTM, BP, KNN, and SVR machine learning models. The objective function is Mean Squared Error (MSE), with 100 iterations for optimization. The training and validation processes converge in approximately 20 iterations. The training loss and validation loss curve diagrams are illustrated in Figure 4, and the optimal parameters for each model are presented in Table 1.



Figure 4. The training and testing loss curves for the Bayesian search algorithms.

Table 1. Optimal parameter configurations for the models.

Model Name	Hyperparameter Name	Range	Optimal Value
CatBoost	Iterations	[50, 300]	184
	Learning rate	[0, 1]	0.15
	Depth	[2, 15]	8
XGBoost	Iterations	[50, 300]	280
	Learning rate	[0, 1]	0.15
	Depth	[2, 15]	10
LightGBM	Iterations	[50, 300]	190
	Learning rate	[0, 1]	0.01
	Depth	[2, 15]	10
BP neural network	Number of neurons in the hidden layer 1	[50, 300]	96
	Number of neurons in the hidden layer 2	[50, 300]	50
	Learning rate	[0, 1]	0.0001
	Epochs	[50, 300]	126
LSTM neural network	Number of neurons in unit1	[50, 300]	32
	Number of neurons in unit2	[50, 300]	59
	Learning rate	[0, 1]	0.006
	Epochs	[50, 300]	254
SVR	C	[0.1, 100]	10
	kernel	['linear', 'rbf', 'poly']	rbf
	epsilon	[0.01, 0.2]	0.01
KNN	K	[1, 9]	6

4.3. Performance Evaluation Metrics

The evaluation metrics are the main criteria for quantitatively assessing model forecasting performance. Evaluation metrics can be categorized into overall error metrics (such as Root Mean Square Error (RMSE)), predictive trend metrics (Nash–Sutcliffe Efficiency (NSE [32]), Coefficient of Determination (R^2)), Feature value error (Relative Error, RE), and normalized Root Mean Square Error (NRMSE), as well as flow duration curve evaluation metrics [33] describing low-, medium-, and high-flow parts. High-flow (top 2%) duration curve percentage bias (%BiasFHV), medium-flow (20–70%) duration curve percentage bias (%BiasFMS), and low-flow (Bottom 20%) duration curve percentage bias (%BiasFLV) are included. In this study, NSE, RMSE, R^2, RE, %BiasFHV, %BiasFMS, and %BiasFLV are selected as evaluation metrics, expressed mathematically as follows:

$$NSE = 1 - \frac{\sum_{i=1}^{n}(Q - Q_i)^2}{\sum_{i=1}^{n}\left(Q - \overline{Q_i}\right)^2} \tag{4}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Q - Q_i)^2}{n}} \tag{5}$$

$$NRMSE = \frac{RMSE}{\overline{Q}} \tag{6}$$

$$R^2 = \frac{\left(\sum_{i=1}^{N}\left(Q - \overline{Q}\right)\left(Q_i - \overline{Q_i}\right)\right)^2}{\sum_{i=1}^{N}\left(Q - \overline{Q}\right)^2 \sum_{i=1}^{n}\left(Q_i - \overline{Q_i}\right)^2} \tag{7}$$

$$RE = \frac{W - W_i}{W} \times 100\% \quad (8)$$

$$\%BiasFHV = \frac{\sum_{h=1}^{H}(\hat{Q}_h - Q_h)}{\sum_{h=1}^{H} Q_h} \times 100\% \quad (9)$$

$$\%BiasFMS = \frac{\left[\log(\hat{Q}_{lower}) - \log(\hat{Q}_{upper})\right] - \left[\log(Q_{lower}) - \log(Q_{upper})\right]}{\left[\log(Q_{lower}) - \log(Q_{upper})\right]} \times 100\% \quad (10)$$

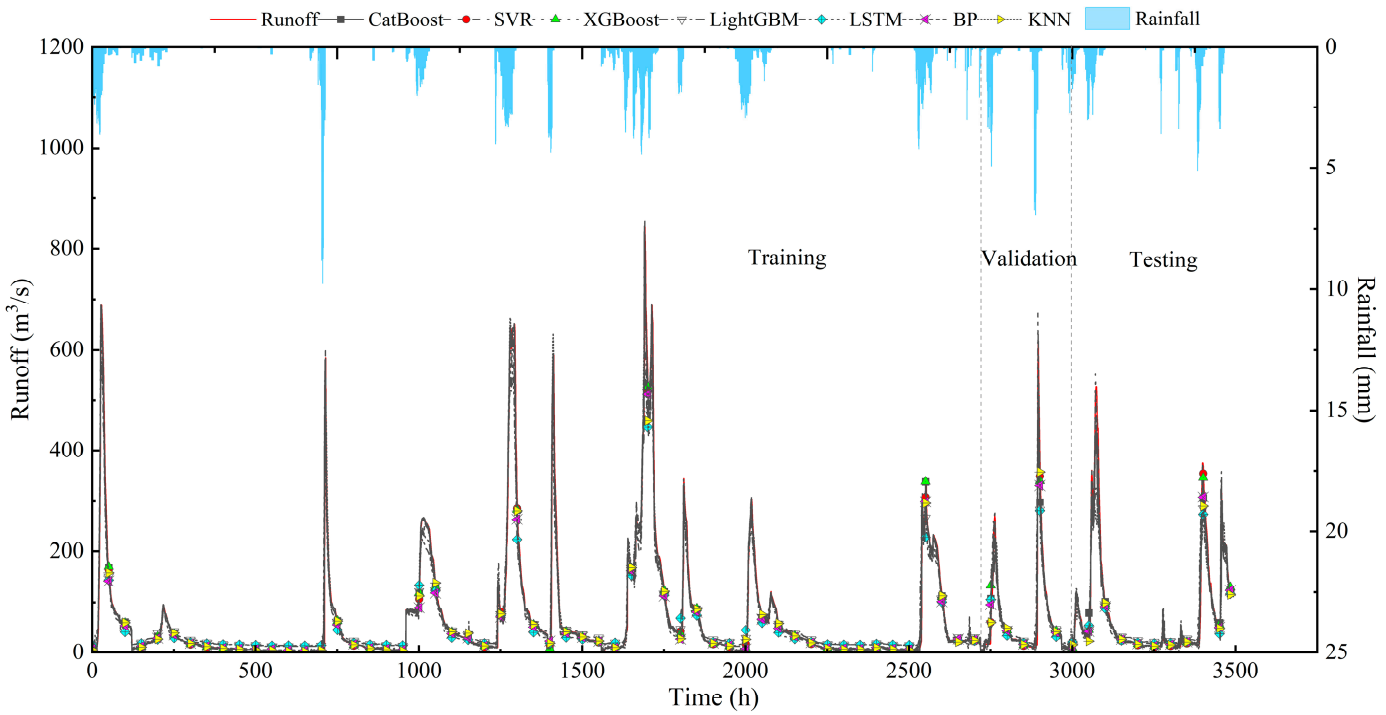
$$\%BiasFLV = -1 \times \frac{\sum_{l=1}^{L}\left[\log(\hat{Q}_l) - \log(\hat{Q}_L)\right] - \sum_{l=1}^{L}\left[\log(Q_l) - \log(Q_L)\right]}{\sum_{l=1}^{L}\left[\log(Q_l) - \log(Q_L)\right]} \times 100\% \quad (11)$$

In the above equations, Q and Q_i represent observed and simulated flow, respectively; W and W_i represent the observed flood volume and the predicted flood volume, respectively; $\overline{Q}$ and $\overline{Q_i}$ represent the average observed and simulated flow; i denotes the i-th time step; and n indicates the number of data points. $\hat{Q}_h$ and Q_h are the simulated and observed flows for the top 2% of flows, with h representing the number of observed values in the top 2%. $\hat{Q}_{lower}$, Q_{lower} and $\hat{Q}_{upper}$, Q_{upper} are the minimum simulated and observed flows, and the maximum simulated and observed flows, respectively, for the middle part (20–70%) of the flow duration curve. $\hat{Q}_l$ and Q_l are the simulated and observed flows for the lower 30% of flows, and $\hat{Q}_L$ and Q_L are the minimum predicted and observed flows. NSE ranges from $-\infty$ to 1, where 1 indicates a perfect fit; $RMSE$ ranges from 0 to $+\infty$, where 0 indicates a perfect fit; R^2 ranges from 0 to 1, where 1 indicates a perfect fit. $NRMSE$ evaluates the generalization ability of the model.

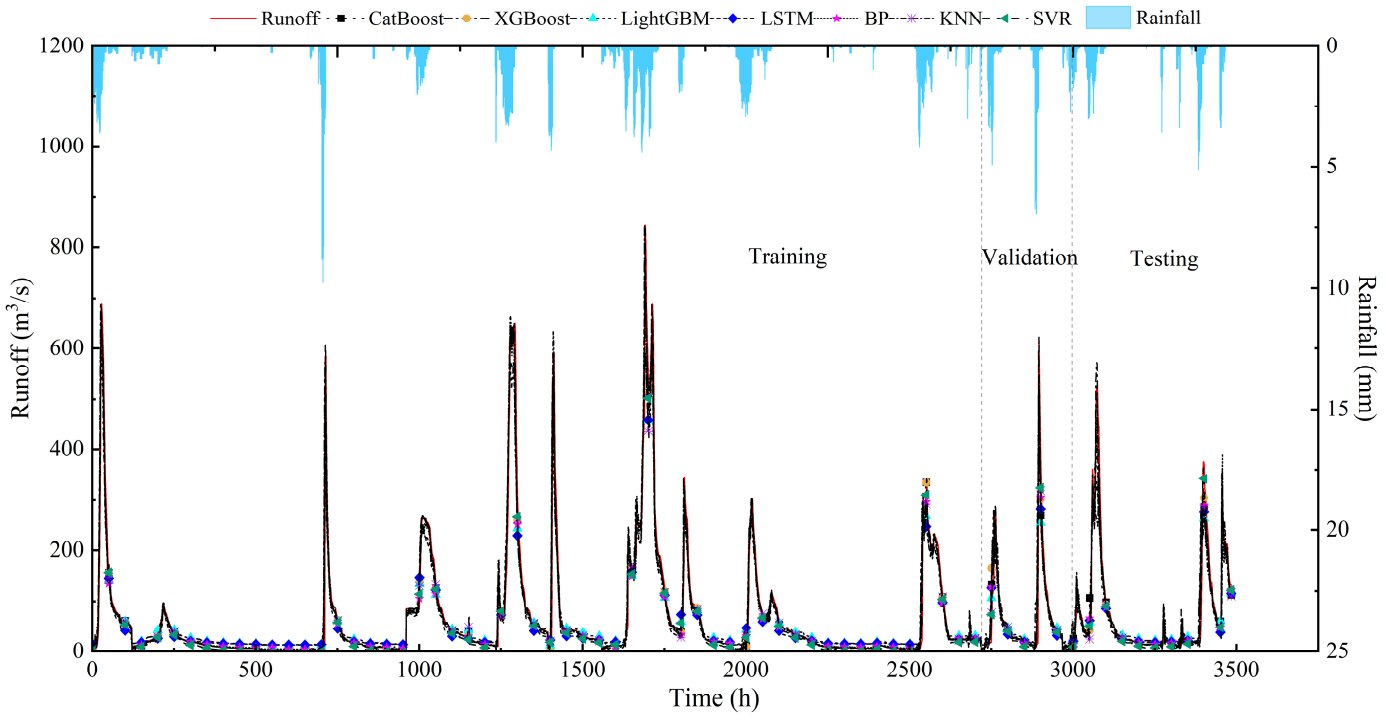
5. Results and Analysis

5.1. Overall Forecasting Performance Evaluation

Figure 5 shows the rainfall–runoff processes of seven models—CatBoost, XGBoost, LightGBM, LSTM, BP, KNN, and SVR—for forecast periods (T) of 1 to 6 h. It can be seen that the forecasted and observed hydrographs are generally in good agreement. Table 2 provides statistical evaluation metrics for the forecast performance of the seven models over forecast periods of 1 to 6 h. From the evaluation metrics, it is evident that for T = 1 h, the NSE ranges from 0.9205 to 0.9965, R^2 ranges from 0.8853 to 0.9741, RMSE ranges from 6.72 to 32.01 m^3/s, %BiasFHV ranges from −18.63% to −0.07%, %BiasFMS ranges from −25.80% to 1.05%, and %BiasFLV ranges from −85.77% to 0.91%. Overall, the machine learning runoff models based on the CARS feature selection algorithm have achieved good simulation results. From the comparison results, it is evident that for T = 1 h, the XGBoost and SVR models perform relatively better. The SVR model has the best NSE and RMSE, at 0.9965 and 6.72 m^3/s, respectively, while the XGBoost model has the best R^2 and %BiasFHV, at 0.9741 and −0.07%, respectively. The forecast results are very close to the observed runoff. At T = 3 h, the accuracy of the machine learning prediction models is lower compared to T = 1 h. The SVR model shows the best performance with NSE and RMSE values of 0.9707 and 19.44 m^3/s, respectively, outperforming other models. At T = 6 h, the forecast accuracy of all seven models declines, with the SVR model still demonstrating the best performance, achieving NSE and RMSE values of 0.8496 and 44.01 m^3/s, respectively.

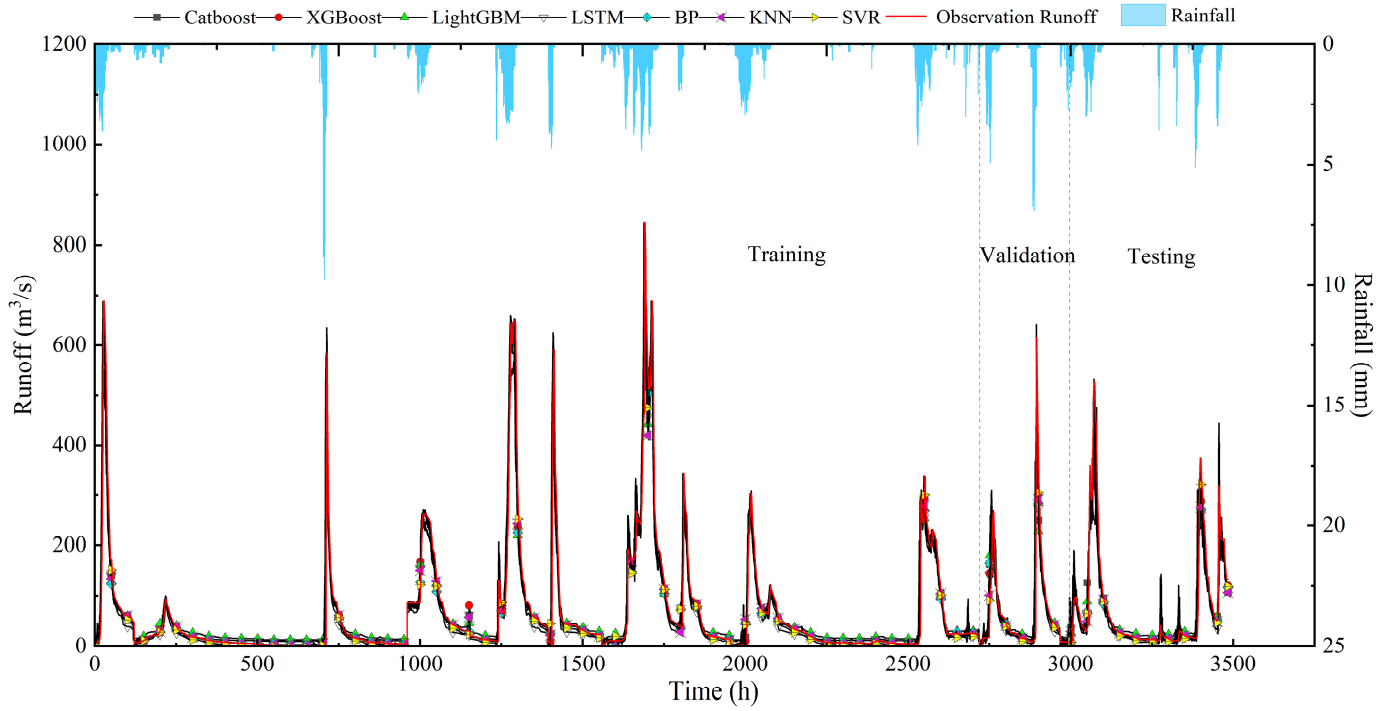


(**a**)

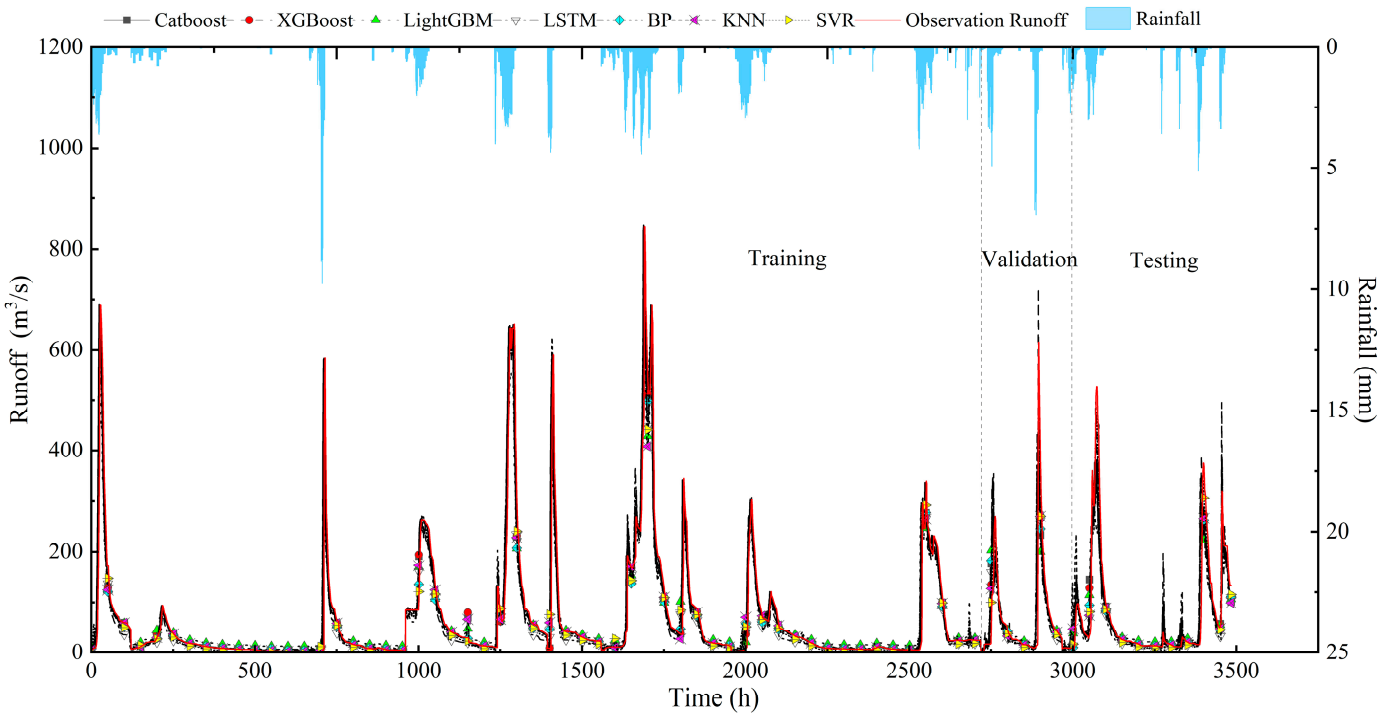


(**b**)

Figure 5. *Cont.*

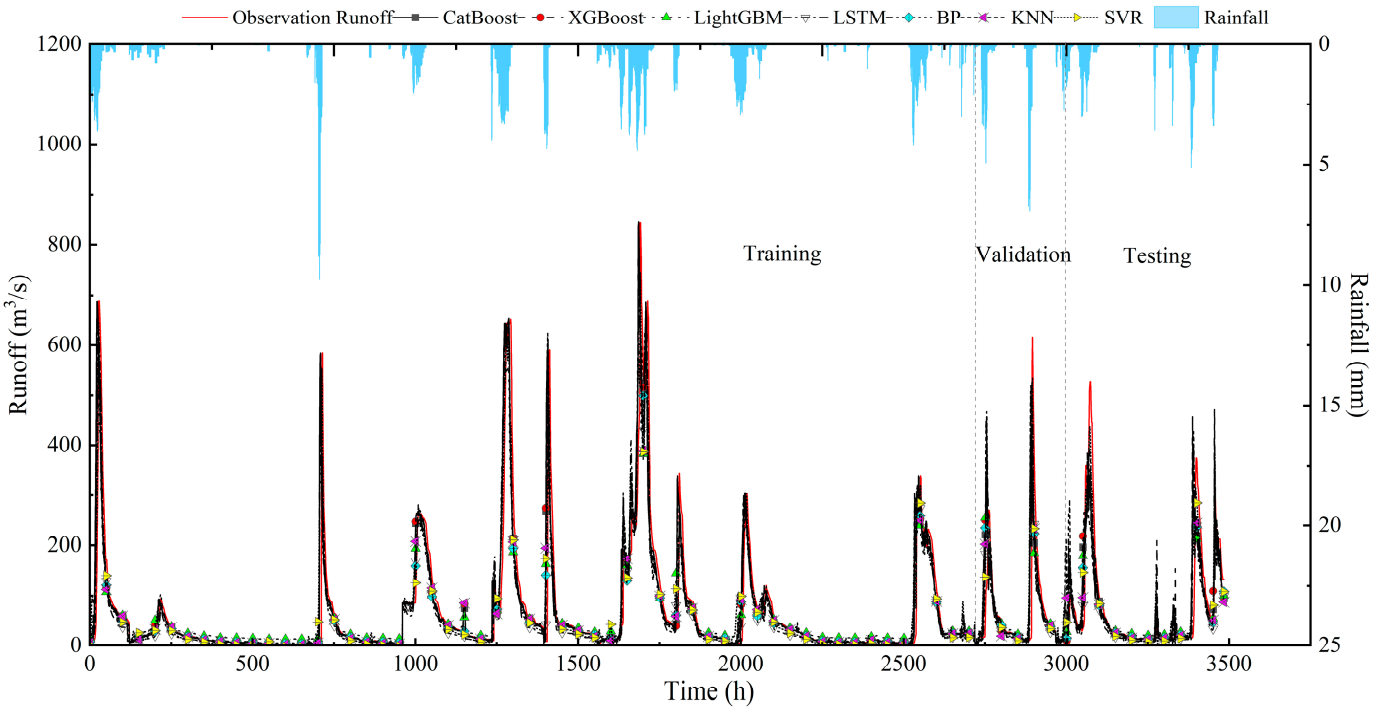


(**c**)

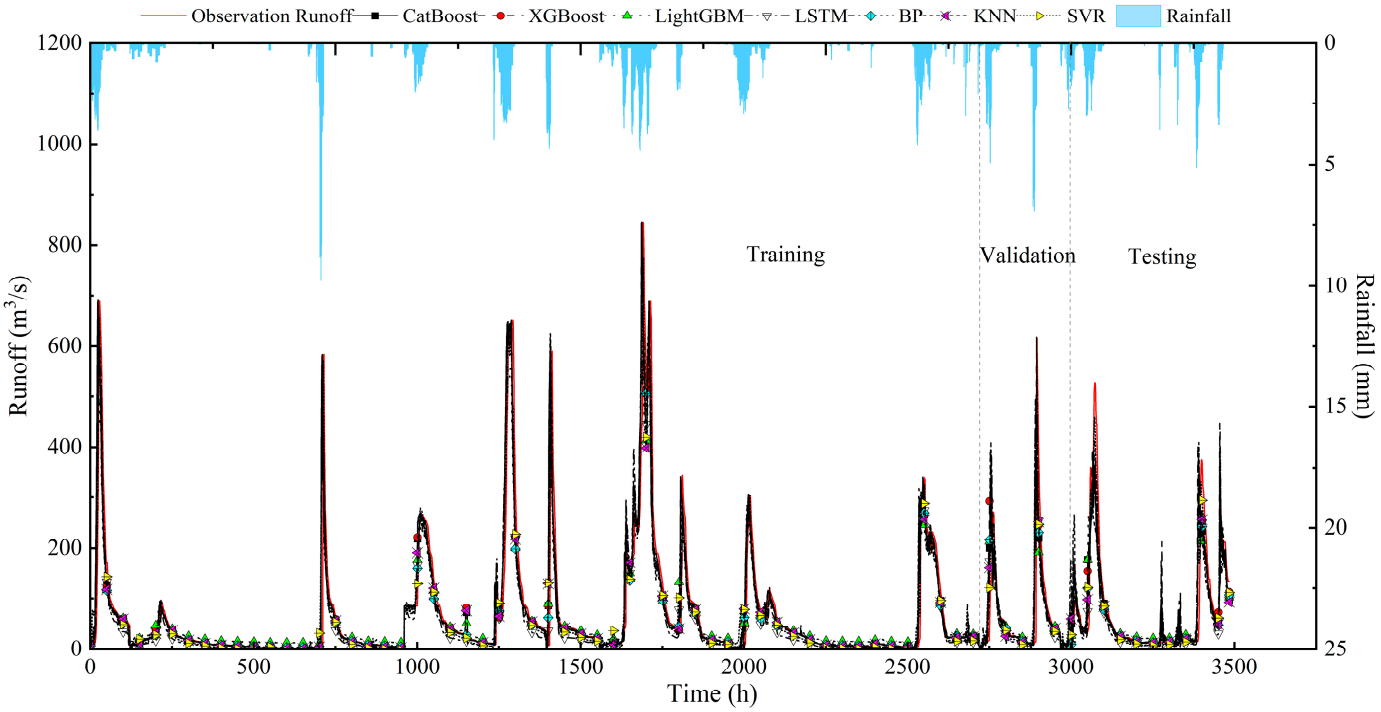


(**d**)

Figure 5. *Cont.*



(**e**)



(**f**)

Figure 5. Comparison of flood forecasting using machine learning models during the training, validation, and testing periods for a prediction period of 1 to 6 h: (**a**) prediction period T = 1 h; (**b**) prediction period T = 2 h; (**c**) prediction period T = 3 h; (**d**) prediction period T = 4 h; (**e**) prediction period T = 5 h; (**f**) prediction period T = 6 h.

Table 2. Statistical evaluation metrics for the forecast performance of the seven models over forecast periods of 1 to 6 h.

Forecast Period	Model Name	NSE	R^2	RMSE	%BiasFHV	%BiasFMS	%BiasFLV
T = 1 h	CatBoost	0.9703	0.9391	19.55	−0.75%	−2.36%	−44.75%
	XGBoost	0.9796	0.9741	16.19	−0.07%	1.05%	−44.38%
	LightGBM	0.9661	0.9415	20.89	−1.11%	−0.81%	−48.00%
	LSTM	0.9205	0.8853	32.01	−18.63%	−25.80%	−43.43%
	BP	0.9689	0.9624	19.99	−7.33%	−8.82%	−85.77%
	KNN	0.9680	0.9139	20.28	−2.51%	−1.32%	0.91%
	SVR	0.9965	0.9730	6.72	−1.58%	2.31%	−7.25%
T = 2 h	CatBoost	0.9311	0.9092	29.77	−0.75%	−2.55%	−53.05%
	XGBoost	0.9361	0.9374	28.66	0.16%	1.54%	−52.73%
	LightGBM	0.9407	0.8969	27.61	−2.07%	0.69%	−54.48%
	LSTM	0.9140	0.8802	33.27	−14.17%	−24.35%	−59.03%
	BP	0.9466	0.9422	26.22	−6.78%	−8.97%	−65.09%
	KNN	0.9419	0.8879	27.33	−2.91%	−2.31%	0.17%
	SVR	0.9874	0.9212	12.72	−3.01%	17.35%	45.23%
T = 3 h	CatBoost	0.8789	0.8921	39.47	−0.75%	−2.30%	−55.05%
	XGBoost	0.8832	0.9033	38.75	−0.27%	0.49%	−54.69%
	LightGBM	0.9011	0.8725	35.66	−2.91%	−1.45%	−65.27%
	LSTM	0.8941	0.8780	36.93	−11.86%	−17.70%	−68.20%
	BP	0.9019	0.9177	35.53	−6.82%	−7.44%	−58.66%
	KNN	0.8996	0.8641	35.93	−3.14%	−3.07%	−0.53%
	SVR	0.9707	0.8826	19.44	−4.13%	11.00%	24.25%
T = 4 h	CatBoost	0.8185	0.8836	48.32	−0.75%	−3.53%	−44.77%
	XGBoost	0.8162	0.8646	48.63	−0.17%	1.27%	−44.19%
	LightGBM	0.8507	0.8588	43.83	−3.59%	−0.09%	−65.57%
	LSTM	0.8597	0.8715	42.55	−9.85%	−6.85%	−83.26%
	BP	0.8659	0.8954	41.55	−8.13%	−9.90%	−48.85%
	KNN	0.8469	0.8456	44.38	−3.47%	−3.64%	−1.09%
	SVR	0.9474	0.8486	26.04	−7.61%	9.28%	32.45%
T = 5 h	CatBoost	0.7563	0.8766	55.99	−0.79%	−3.79%	−61.98%
	XGBoost	0.7522	0.8369	56.45	−0.44%	0.51%	−61.01%
	LightGBM	0.7954	0.8467	51.30	−4.06%	−3.66%	−65.08%
	LSTM	0.8209	0.8586	48.13	−6.92%	14.62%	13.90%
	BP	0.8221	0.8809	47.86	−7.81%	−9.38%	−65.79%
	KNN	0.7881	0.8249	52.22	−3.49%	−2.49%	−1.32%
	SVR	0.9004	0.8411	35.82	−8.92%	11.78%	19.15%
T = 6 h	CatBoost	0.6912	0.8718	63.03	−0.77%	−3.17%	−56.11%
	XGBoost	0.6874	0.8319	63.41	−0.65%	0.68%	−54.42%
	LightGBM	0.7373	0.8293	58.13	−3.85%	−1.38%	−60.46%
	LSTM	0.7983	0.8471	51.04	−7.38%	4.83%	−28.25%
	BP	0.7654	0.8645	54.94	−8.07%	−9.59%	−64.61%
	KNN	0.7274	0.8019	59.22	−3.54%	−2.69%	−1.31%
	SVR	0.8496	0.8315	44.01	−10.92%	12.81%	21.85%

Figure 6 shows scatter plots of the observed and predicted flow rates during the training, validation, and testing periods for seven models at forecast horizons of T = 1 h, T = 3 h, and T = 6 h. As the forecast horizon increases, the scatter points of the seven models diverge to varying degrees, gradually deviating from the 1:1 line. However, the scatter plots of the SVR model are closer to the 1:1 line compared to the other six machine learning models at each forecast horizon, indicating a weaker degree of deviation. Figure 7 presents the R^2 radar charts for the validation and testing periods of the seven models at forecast horizons of T = 1~6 h. During the validation and testing periods, the R^2 values of the seven models decrease to varying degrees as the forecast horizon increases, with the SVR model showing a smaller decrease in R^2 values. Specifically, during the testing period, the R^2

values of the SVR model at forecast horizons of T = 1 h, 3 h, and 6 h are 0.9854, 0.9144, and 0.8238, respectively, outperforming the other models at the same forecast horizons. This indicates that the SVR model can better capture the relationship between simulated and observed flow rates. The results show that the SVR model slightly outperforms the other six models, which can be attributed to the SVR model's superior capability in handling small-scale nonlinear data [34]. The prediction accuracy of the seven models decreases with an increasing forecast horizon due to the larger time gap between inputs and outputs in the training set, resulting in reduced data correlation and consequently lower prediction accuracy of the machine learning models.

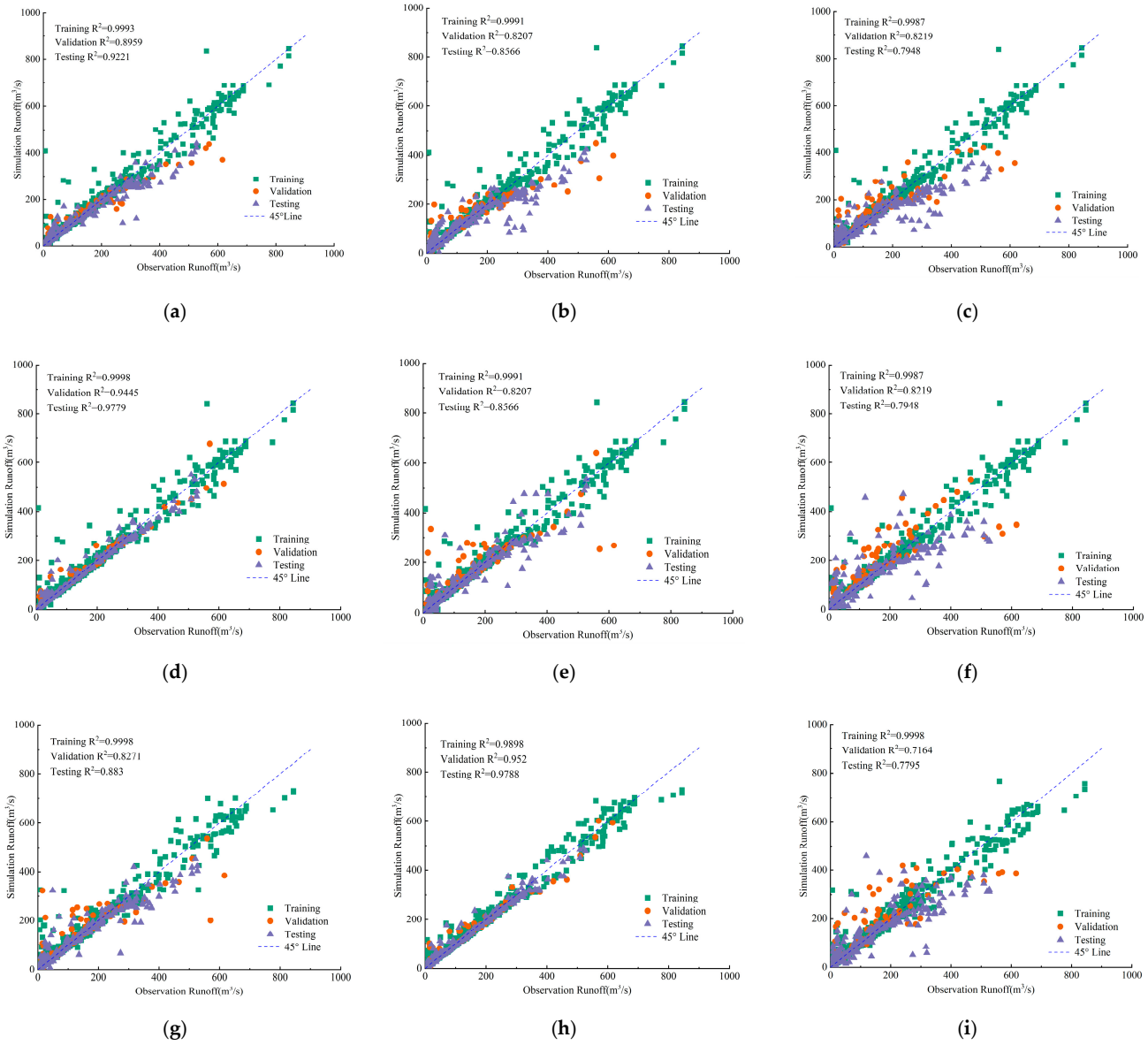


Figure 6. *Cont.*

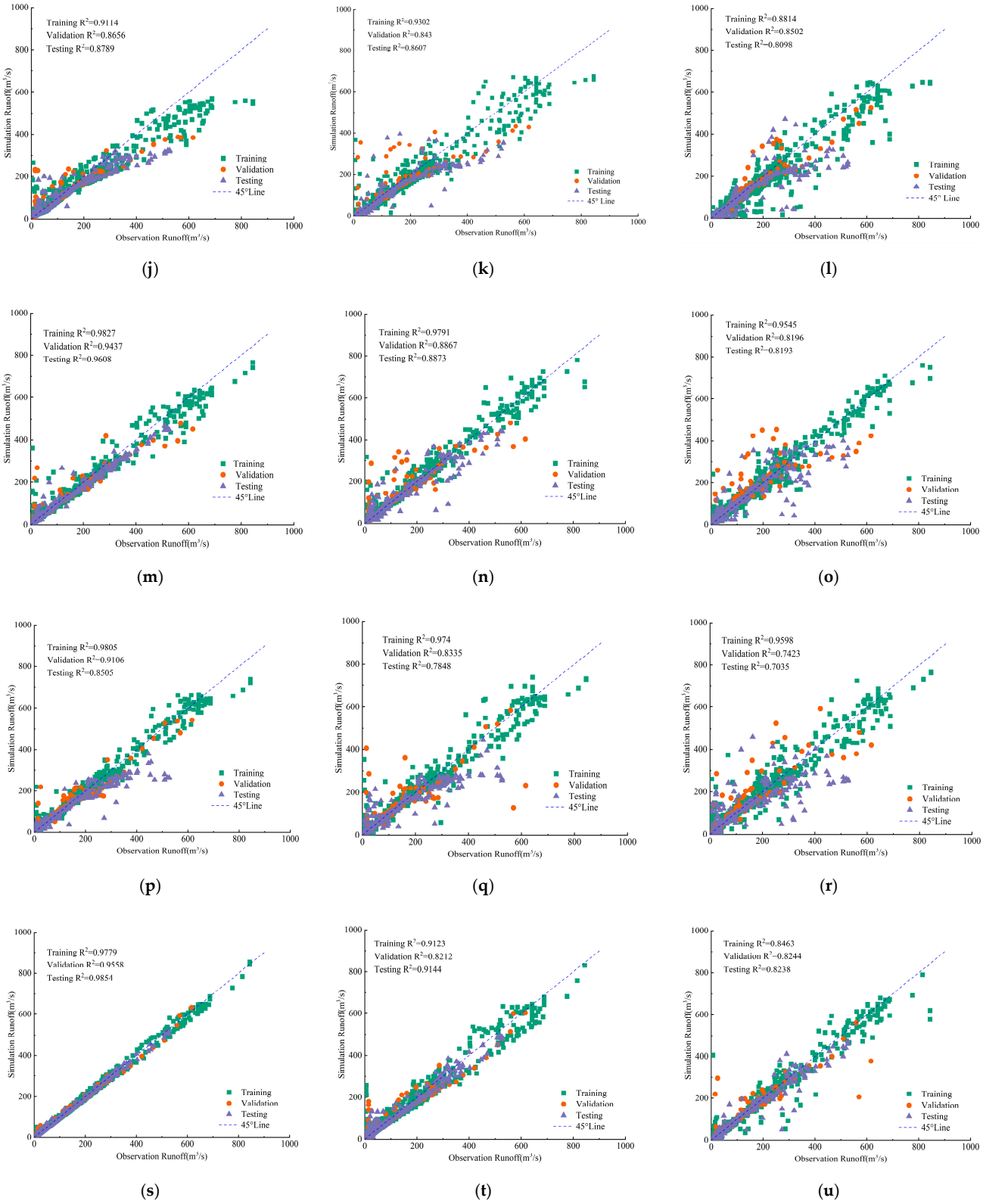


Figure 6. Scatter plots of observed and simulated flows for seven different models during the training, validation, and testing periods under various prediction periods: (**a**) CatBoost, T = 1 h; (**b**) CatBoost, T = 3 h; (**c**) CatBoost, T = 6 h; (**d**) XGBoost, T = 1 h; (**e**) XGBoost, T = 3 h; (**f**) XGBoost, T = 6 h; (**g**) LigthGBM, T = 1 h; (**h**) LigthGBM, T = 3 h; (**i**) LigthGBM, T = 6 h; (**j**) LSTM, T = 1 h; (**k**) LSTM, T = 3 h; (**l**) LSTM, T = 6 h; (**m**) BP, T = 1 h; (**n**) BP, T = 3 h; (**o**) BP, T = 6 h; (**p**) KNN, T = 1 h; (**q**) KNN, T = 3 h; (**r**) KNN, T = 6 h; (**s**) SVR, T = 1 h; (**t**) SVR, T = 3 h; (**u**) SVR, T = 6 h.

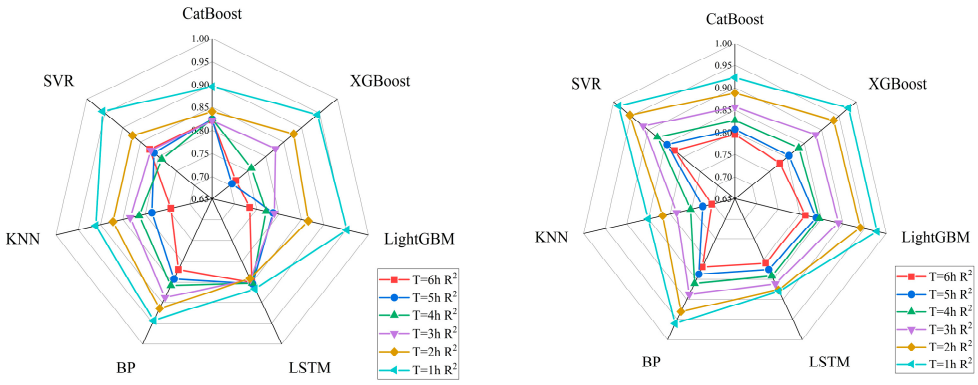


Figure 7. Radar charts of R^2 for the validation and testing periods with T ranging from 1 to 6 h: (**a**) validation; (**b**) testing.

5.2. Effectiveness of Flood Forecasting for Different Events

In order to further understand the effectiveness of the models in forecasting flood events, we analyzed the hydrograph comparison for two typical flood events during the validation and testing periods at T = 1 h, 3 h, and 6 h. The two typical flood events selected are the '20090828' and the '20100820' events, as shown in Figure 8. Table 3 is the RE of flood volumes for forecast horizons T = 1, 3, and 6 h. As seen from Table 3 and Figure 8, among the simulated hydrograph of flood events generated with the seven models, there are fluctuations in the pre-peak stage, and the start of the rise occurs earlier than the actual measured flood. However, the peak flow present time and the recession stage show good forecasting performance. The SVR model's forecasted flow is closest to the actual measured flow, and it is able to predict the peak flow well, with a minimum RE of flood volumes for flood events, with the most stable overall forecasting performance and good flood forecasting capabilities. Additionally, as the prediction period increases, the forecasting effectiveness of the seven models decreases, and the RE of flood volumes increases. For the T = 6 h prediction period, most REs are positive, indicating that the models tend to underestimate peak flows and flood volumes, with the underestimation becoming more evident with a longer prediction period, but the peak flow present time does not show significant delay. This is because, as the prediction period increases, the correlation between the input data and the forecasted flow data decreases, making it more challenging to accurately learn and extract flood data features.

Table 3. RE of flood volumes for forecast horizons T = 1, 3, and 6 h.

Flood Event Number	Model Name	1 h	3 h	6 h
20090828	svr	0.30%	1.96%	2.65%
	knn	−4.60%	−2.14%	7.98%
	bp	0.57%	3.26%	5.23%
	lightGBM	−4.21%	−1.12%	7.64%
	lstm	0.70%	2.67%	16.43%
	Xgboost	−1.87%	−1.97%	4.76%
	catboost	−0.20%	0.30%	10.50%
20100820	svr	−0.10%	1.73%	3.08%
	knn	0.11%	2.52%	8.23%
	bp	−1.75%	2.65%	3.21%
	lightGBM	−2.32%	0.50%	3.36%
	lstm	−1.35%	3.60%	23.36%
	Xgboost	−0.90%	3.25%	4.90%
	catboost	−3.30%	−3.05%	11.50%

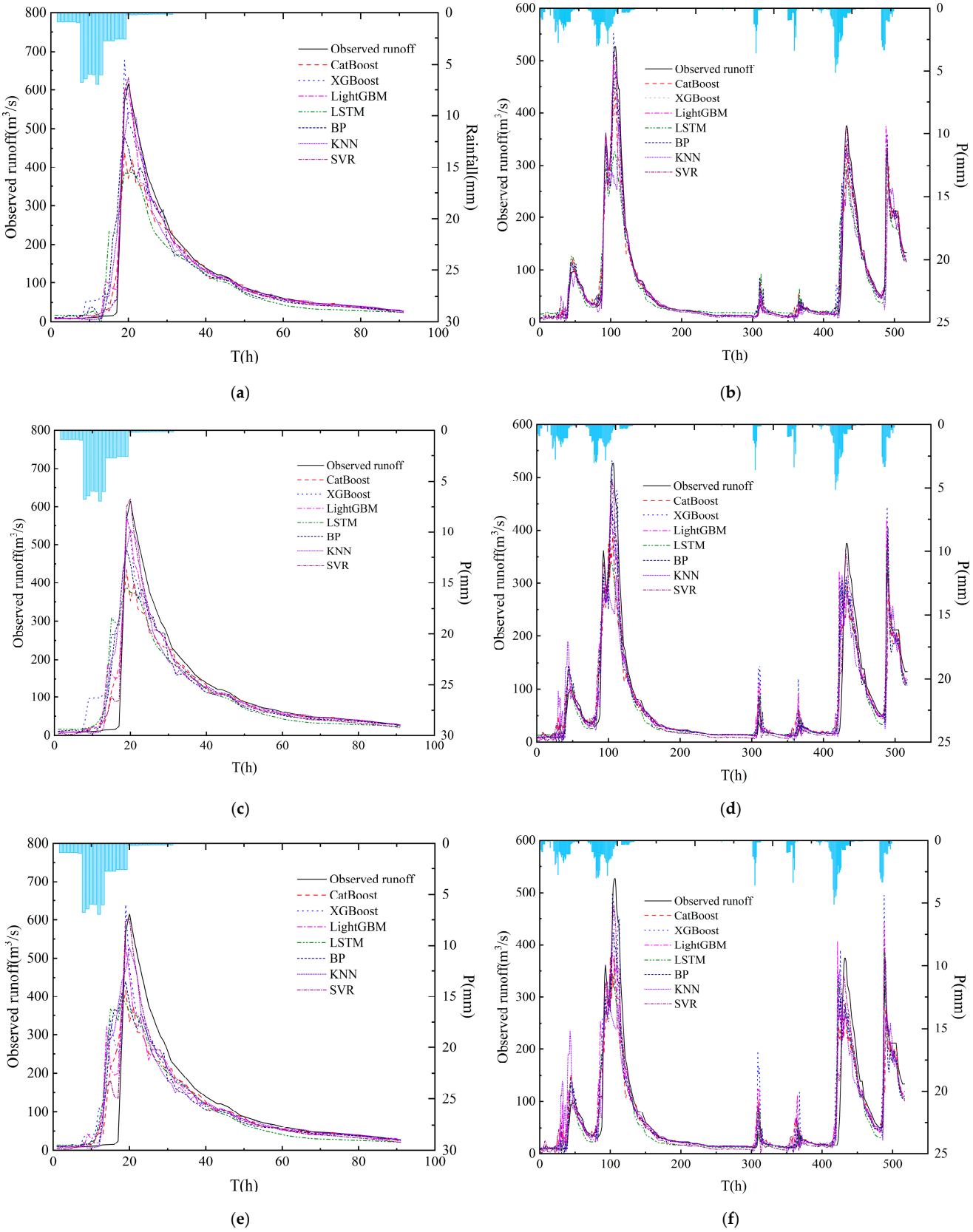


Figure 8. Comparison of observed flow and predictions from seven models for the '20090828' and '20100820' flood events: (**a**) "20090828", T = 1 h; (**b**) "20100820", T = 1 h; (**c**) "20090828", T = 3 h; (**d**) "20100820", T = 3 h; (**e**) "20090828", T = 6 h; (**f**) "20100820", T = 6 h.

5.3. Percentage Analysis of Duration Curves for Low, Medium, and High Flows

Analyzing the percentage deviations of historical duration curves for different flow segments can identify systematic biases in forecasting models and help optimize runoff models. The percentages of duration curves for low, medium, and high flows under different prediction periods are shown in Figure 9.

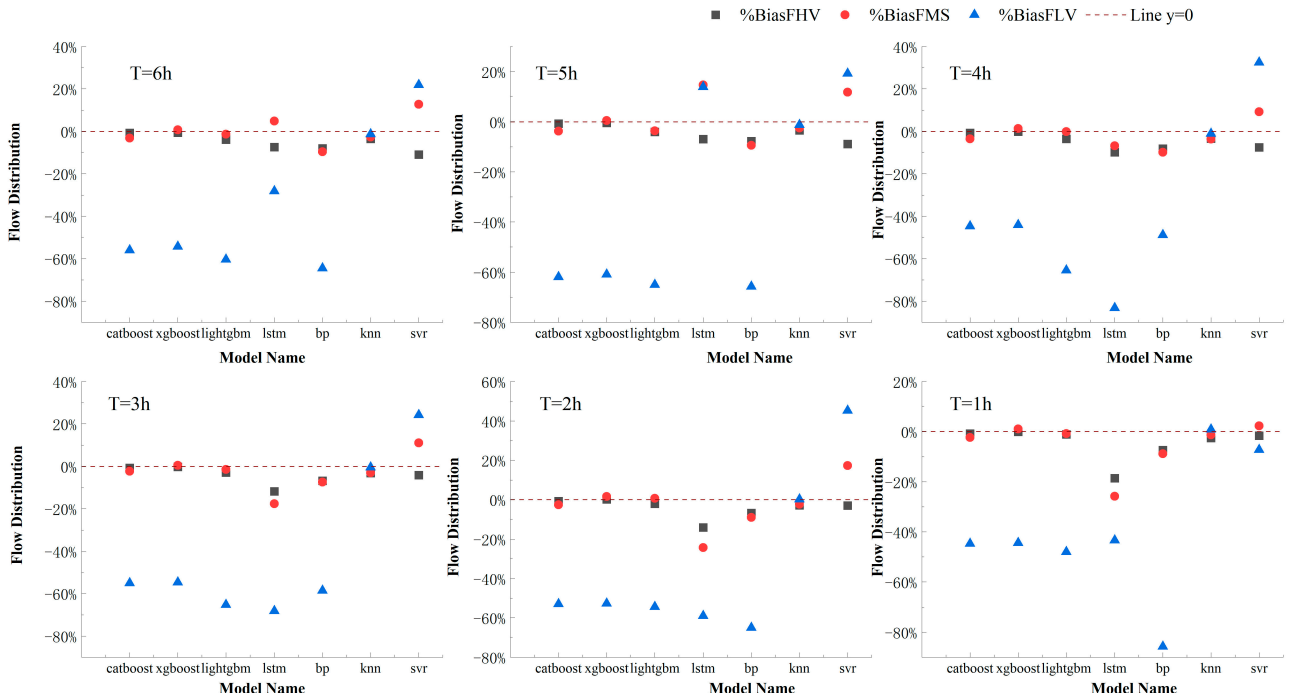


Figure 9. Percentages of duration curves for low, medium, and high flows under different prediction periods.

Based on the analysis of forecasting results for different runoff flow segments, the percentage deviations (%BiasFHV) of high flow duration curves under different models show little difference. Most of the %BiasFHV values for the seven models based on gradient boosting decision trees, neural networks, and shallow learning models are negative. The %BiasFHV values for the CatBoost, XGBoost, LightGB, KNN, and SVR models are close to 0%, with small fluctuations, indicating good forecasting performance for high runoff flows, although the forecasts are slightly lower. The %BiasFHV values for the LSTM and BP models fluctuate significantly, with the predictions being relatively lower, and the underestimation is greater for the LSTM model compared to the BP model. For medium flow, except for the LSTM and BP models, the %BiasFHV values for the historical duration curves of the other five models are similar and close to 0%, indicating similar forecasting performance under normal flow conditions. For low flow, except for the SVR model, the %BiasFLV values for the historical duration curves of the other six models are negative, with the forecasts being relatively lower. The %BiasFLV value for the SVR model is positive, indicating higher forecast results for lower runoff flows and showing that the SVR model does not underestimate runoff flow during flood forecasting. There is no obvious trend change in the low, medium, and high flow predictions with varying forecast periods for each model. Considering the three evaluation indicators—%BiasFHV, %BiasFMS, and %BiasFLV—the SVR model's simulated runoff shows higher values in high-flow segments, lower values in medium–high-flow segments, and lower forecast accuracy in low-flow segments compared to medium–high-flow segments. Overall, the SVR model has the smallest bias and performs the best among the models.

5.4. Analysis of Model Generalization Ability

The generalization ability of machine learning models refers to their capability to predict unknown data after training. It is a key factor in evaluating machine learning forecasting performance. Based on the prediction results from the training, validation, and testing datasets of the study basin, the generalization ability of different machine learning models is compared using RMSE. The NRMSE results for each forecasting scenario across the three datasets are shown in Table 4. The distribution of NRMSE for different models at various forecast horizons is shown in Figure 10.

Table 4. RMSE of seven models during the training, validation, and testing periods under different lead times.

Forecast Period	Model	Training	Validation	Testing
T = 1 h	CatBoost	0.2341	0.4302	0.4021
	XGBoost	0.2366	0.2555	0.2099
	LightGBM	0.3009	0.2862	0.3153
	LSTM	0.4323	0.6051	0.5346
	BP	0.2699	0.4682	0.2746
	KNN	0.1983	0.3595	0.5575
	SVR	0.0963	0.1182	0.0908
T = 2 h	CatBoost	0.4084	0.5407	0.4819
	XGBoost	0.4097	0.5568	0.3567
	LightGBM	0.3904	0.4405	0.4245
	LSTM	0.4385	0.7263	0.5439
	BP	0.3597	0.5863	0.3523
	KNN	0.3213	0.5329	0.6115
	SVR	0.1816	0.2340	0.1689
T = 3 h	CatBoost	0.5589	0.7006	0.5647
	XGBoost	0.5598	0.6944	0.4878
	LightGBM	0.4942	0.6297	0.5639
	LSTM	0.4827	0.8591	0.5859
	BP	0.4887	0.8054	0.4612
	KNN	0.4554	0.7313	0.6878
	SVR	0.2733	0.3907	0.2584
T = 4 h	CatBoost	0.6908	0.8687	0.6487
	XGBoost	0.6922	0.8745	0.6687
	LightGBM	0.5990	0.8277	0.7015
	LSTM	0.5552	1.0233	0.6548
	BP	0.5767	0.9180	0.5296
	KNN	0.5790	0.9202	0.7797
	SVR	0.3651	0.5269	0.3487
T = 5 h	CatBoost	0.8029	1.0217	0.7268
	XGBoost	0.8045	0.9835	0.7907
	LightGBM	0.6978	0.9894	0.8246
	LSTM	0.6286	1.1786	0.7220
	BP	0.6655	1.0571	0.6036
	KNN	0.6865	1.1190	0.8743
	SVR	0.4993	0.7531	0.4748
T = 6 h	CatBoost	0.9025	1.1709	0.8119
	XGBoost	0.9048	1.0853	0.8936
	LightGBM	0.7976	1.1246	0.9009
	LSTM	0.6660	1.2568	0.7629
	BP	0.7662	1.2017	0.6904
	KNN	0.7811	1.2928	0.9665
	SVR	0.6132	0.9273	0.5832

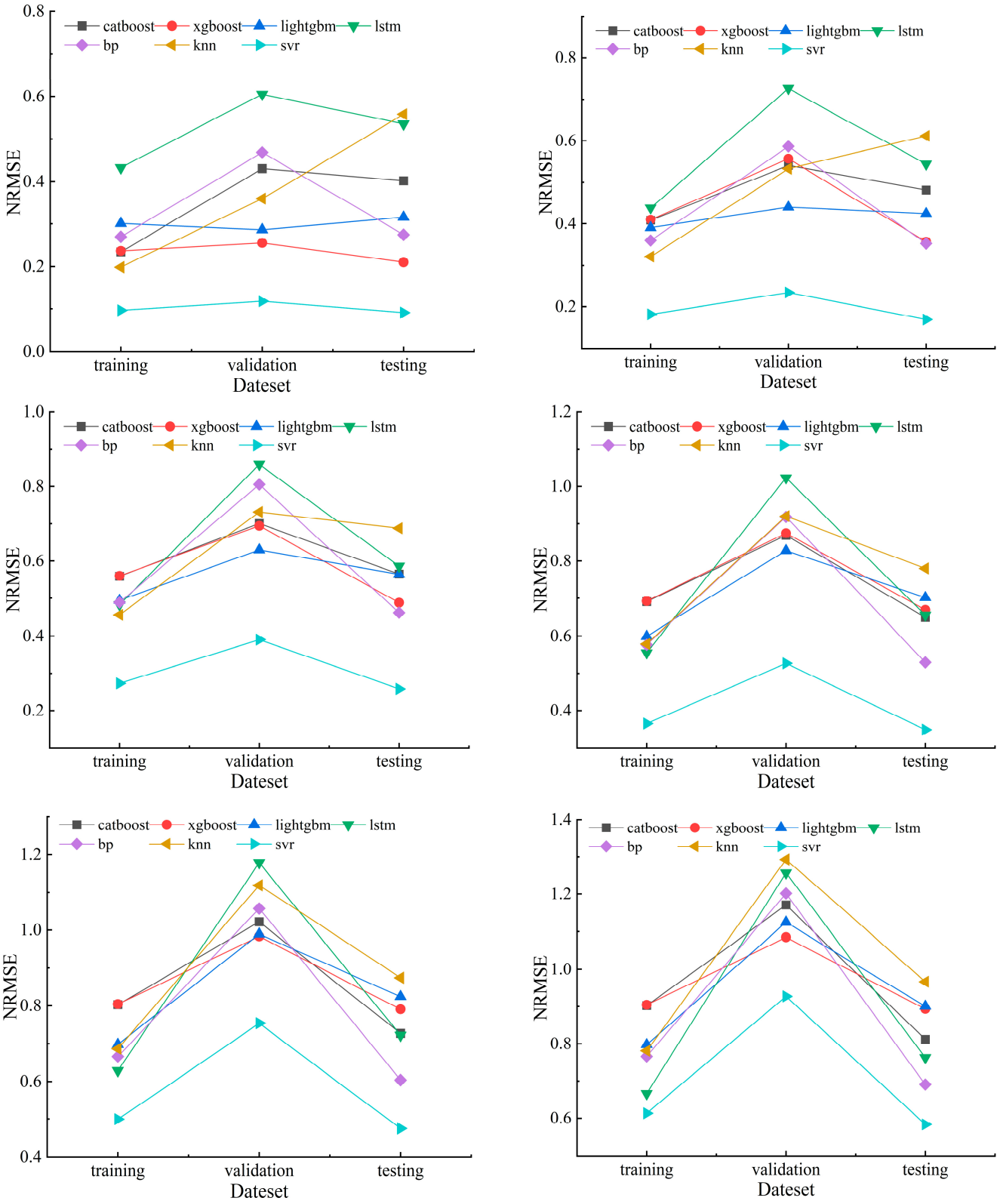


Figure 10. Distribution of NRMSE for seven models during the training, validation, and testing periods under different lead times.

Based on the analysis of Table 3 and Figure 10, comparing the generalization ability of different models for runoff prediction, at the forecast horizon of T = 1 h, the NRMSE values of the LightGBM and KNN models on the test set are higher than those on the validation

and training sets. For the other five models, the test set NRMSE values are lower than those on the validation set but higher than those on the training set; At the forecast horizon of T = 2 h, the KNN model's test set NRMSE values are higher than those on the validation and training sets. For the other six models, the test set NRMSE values are lower than those on the validation set but higher than those on the training set. At T = 3, 4, 5, and 6 h, the test set NRMSE values for all seven models are lower than those on the validation set but higher than those on the training set. This indicates that the generalization ability of the machine learning models is relatively good across different forecast horizons, with the best performance at T = 1 h. As the forecast horizon increases, the NRMSE values for the training, validation, and test sets gradually decrease, and generalization ability diminishes.

Among the models, LSTM shows lower prediction accuracy with higher NRMSE values for both the training and test sets, indicating an underfitting condition. At T = 1 and 2 h, the KNN model exhibits significantly lower NRMSE values for the training set compared to the test set, with the test set NRMSE being notably higher than that of other models, showing clear overfitting. The SVR model demonstrates the best predictive performance, with lower NRMSE values across training, validation, and test sets compared to other models.

6. Conclusions

This research proposed the application of machine learning models based on CARS feature selection for flood forecasting in the Ba river basin, establishing prediction models for lead times of 1 to 6 h. By comparing the simulation results of different machine learning models, the following conclusions can be drawn:

(1) Machine learning runoff models based on the CARS feature selection algorithm demonstrate good applicability in flood forecasting. For a lead time of T = 1 h, the NSE of the seven models ranges from 0.9205 to 0.9965, the R^2 ranges from 0.8853 to 0.9741, and the RMSE ranges from 6.72 m^3/s to 32.01 m^3/s. Forecast accuracy decreases to some extent with increasing lead time. Overall, the machine learning runoff models based on the CARS feature selection algorithm achieve satisfactory simulation results, with the SVR model showing relatively high forecasting accuracy;

(2) Through the study of flood forecasting for different flood events, during the simulation process, the seven models often exhibit fluctuations in the pre-peak stage, with the onset of rising occurring earlier than the actual measured flood. However, they demonstrate good forecasting performance in terms of peak flow present time and recession stage. The SVR model predicts flow closest to the measured flow, performs well in predicting peak flow, and exhibits the most stable overall forecasting performance, indicating good flood forecasting capabilities. As the forecast lead time increases, the NSE and R^2 of the seven models decrease to varying extents, resulting in reduced forecasting accuracy. Additionally, the predicted peak flow for flood events tends to decrease with longer lead times, leading to instances where the models underestimate the peak flow;

(3) Considering the percentage indicators of flow historical curves (%BiasFHV, %BiasFMS, and %BiasFLV), the SVR model's simulated runoff shows overestimation in high-flow segments, underestimation in medium–high-flow segments, and lower accuracy in low-flow segments compared to medium–high-flow segments. Overall, it exhibits the smallest deviations compared to other models and performs the best;

(4) The NRMSE values of the seven models on the testing dataset are sometimes higher than those on the validation dataset but lower than those on the training dataset, indicating good generalization ability. As the lead time increases, the NRMSE values gradually decrease on the training, validation, and testing datasets, suggesting a gradual decline in generalization ability.

Data-driven models improve the accuracy of rainfall–runoff simulations by learning complex nonlinear relationships in historical data. This study compares the performance of various data-driven models (such as regression and neural networks), enhancing the

theoretical foundation of rainfall–runoff modeling. The data-driven models used in this study achieve high accuracy in flood forecasting, optimizing water resource management and emergency response, and providing scientific decision support for scheduling. They also improve the prediction capability for extreme weather events, helping to reduce disaster losses and enhance the resilience and response capacity of watersheds.

Although the data-driven rainfall–runoff simulations in this study excel in improving prediction accuracy and supporting flood forecasting, there are still several limitations and shortcomings. Firstly, the predictive capability of the models heavily relies on the quality and completeness of the input data. Missing data, measurement errors, or data noise can significantly affect the accuracy of the model, making high-quality and complete data a prerequisite for successful application. Secondly, while data-driven models can capture the complex nonlinear relationships in rainfall–runoff processes, their generalization ability may be limited. The models may perform poorly under extreme weather events or rapidly changing environmental conditions, indicating a need for further optimization and adjustment in future applications. Selecting a normalization method that suits the data characteristics has a significant impact on model performance and evaluation accuracy. Therefore, related research can be pursued in the future. When applying this model to other regions, it is essential to ensure that the new area's data are of sufficient quality and completeness. Understand the hydrological characteristics of the target watershed, including topography, soil types, and land use. Adjust and optimize the model according to the specific conditions of the new region. Conduct thorough validation before applying the model to ensure its applicability and predictive accuracy in the new area.

Author Contributions: Conceptualization, Y.F. and G.K.; methodology, Y.F. and X.F.; software, X.F.; validation, G.K. and H.Y.; formal analysis, Y.F. and X.F.; investigation, G.K. and K.L; resources, X.F. and K.L.; data curation, Y.F.; writing—original draft preparation, Y.F., X.F. and K.L.; writing—review and editing, Y.F. and X.F.; visualization, Y.F.; supervision, G.K.; project administration, H.Y.; funding acquisition, G.K. and H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key R&D Program of China (2023YFC3010704, 2023YFC3209202); the IWHR Research & Development Support Program (JZ0199A032021); GHFUND A (No. ghfund202407012809); and the Significant Science and Technology Project of the Ministry of Water Resources (SKR-2022056). We gratefully acknowledge the support from the Key Laboratory of Water Safety for the Beijing-Tianjin-Hebei Region of the Ministry of Water Resources. We gratefully acknowledge the support of the NVIDIA Corporation with the donation of the Tesla K40 and TITAN V GPUs used for this research.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author Xiaodi Fu was employed by the company Beijing Engineering Corporation Limited. Author Ke Liang was employed by the company Beijing IWHR Corporation. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Kan, G.; Hong, Y.; Liang, K. Flood Forecasting Research Based on Coupled Machine Learning Models. *China Rural Water Hydrol.* **2018**, *10*, 165–169, 176. (In Chinese)
2. Kan, G. Study on Application and Comparative of Data-Driven Model and Semi-Data-Driven Model for Rainfall-Runoff Simulation. *Acta Geod. Cartogr. Sin.* **2017**, *46*, 265. (In Chinese)
3. Liang, K.; Kan, G.; Li, Z. Application of A New Coupled Data-driven Model in Rainfall-Runoff Simulation. *J. China Hydrol.* **2016**, *36*, 1–7. (In Chinese)
4. Beven, K.J.; Kirkby, M.J. A physically based, variable contributing area model of basin hydrology/Un modèle à base physique de zone d'appel variable de l'hydrologie du bassin versant. *Hydrol. Sci. J.* **1979**, *24*, 43–69. [CrossRef]
5. Du, T.; Guo, M.; Zhang, J.; Tian, S. Hydrological simulation and application of VIC model in the Xijiang River Basin. *Res. Soil Water Conserv.* **2021**, *28*, 121–127. (In Chinese) [CrossRef]
6. Zhao, A.Z.; Liu, X.F.; Zhu, X.F.; Pan, Y.Z.; Li, Y.Z. Spatial and temporal distribution of drought in the Weihe River Basin based on SWAT model. *Prog. Geogr.* **2015**, *34*, 1156–1166. (In Chinese)

7. Baker, T.J.; Miller, S.N. Using the Soil and Water Assessment Tool (SWAT) to assess land use impact on water resources in an East African watershed. *J. Hydrol.* **2013**, *486*, 100–111. [CrossRef]
8. Assaf, M.N.; Manenti, S.; Creaco, E.; Giudicianni, C.; Tamellini, L.; Todeschini, S. New optimization strategies for SWMM modeling of stormwater quality applications in urban area. *J. Environ. Manag.* **2024**, *361*, 121244. [CrossRef]
9. Zhang, L.; Wang, H.; Guo, N.; Xu, Y.; Li, L.; Xie, J. Ensemble modeling of non-stationary runoff series based on time series decomposition and machine learning. *Prog. Water Sci.* **2023**, *34*, 42–52. (In Chinese)
10. Li, B.; Tian, F.; Li, Y.; Ni, G. Deep learning hydrological model integrating spatial-temporal characteristics of meteorological elements. *Prog. Water Sci.* **2022**, *33*, 904–913. (In Chinese)
11. Valipour, M. Long-term runoff study using SARIMA and ARIMA models in the United States. *Meteorol. Appl.* **2015**, *22*, 592–598. [CrossRef]
12. Duan, Y.; Ren, L. Research on daily runoff prediction of the middle reaches of the Yellow River based on BP neural network. *People's Yellow River* **2020**, *42*, 5–8. (In Chinese)
13. Kratzert, F.; Klotz, D.; Brenner, C.; Schulz, K.; Herrnegger, M. Rainfall-runoff modelling using long short-term memory (LSTM) networks. *Hydrol. Earth Syst. Sci.* **2018**, *22*, 6005–6022. [CrossRef]
14. Chiang, Y.-M.; Chang, F.-J. Integrating hydrometeorological information for rainfall-runoff modelling by artificial neural networks. *Hydrol. Process. Int. J.* **2009**, *23*, 1650–1659. [CrossRef]
15. Kan, G.; Yao, C.; Li, Q.; Li, Z.; Yu, Z.; Liu, Z.; Ding, L.; He, X.; Liang, K. Improving event-based rainfall-runoff simulation using an ensemble artificial neural network based hybrid data-driven model. *Stoch. Environ. Res. Risk Assess.* **2015**, *29*, 1345–1370. [CrossRef]
16. Guo, T.; Song, S.; Zhang, T.; Wang, H. A new step-by-step decomposition integrated runoff prediction model based on two-stage particle swarm optimization algorithm. *J. Hydraul. Eng.* **2022**, *53*, 1456–1466. (In Chinese)
17. Xiong, Y.; Zhou, J.; Sun, N.; Zhang, J.; Zhu, S. Monthly runoff forecast based on adaptive variational mode decomposition and long short-term memory network. *J. Hydraul. Eng.* **2023**, *54*, 172–183, 198. (In Chinese)
18. Li, C.; Jiao, Y.; Kan, G.; Fu, X.; Chai, F.; Yu, H.; Liang, K. Comparisons of Different Machine Learning-Based Rainfall–Runoff Simulations under Changing Environments. *Water* **2024**, *16*, 302. [CrossRef]
19. Kan, G.; Li, J.; Zhang, X.; Ding, L.; He, X.; Liang, K.; Jiang, X.; Ren, M.; Li, H.; Wang, F.; et al. A new hybrid data-driven model for event-based rainfall–runoff simulation. *Neural Comput. Appl.* **2017**, *28*, 2519–2534. [CrossRef]
20. Ding, Q.; Wang, Y.; Zhang, J.; Jia, K.; Huang, H. Joint estimation of soil moisture and organic matter content in saline-alkali farmland using the CARS algorithm. *Chin. J. Appl. Ecol.* **2024**, *35*, 1321–1330. [CrossRef]
21. Wang, Y.; Ding, Q.; Zhang, J.; Chen, R.; Jia, K.; Li, X. Retrieval of soil water and salt information based on UAV hyperspectral remote sensing and machine learning. *Appl. Ecol. J.* **2023**, *34*, 3045–3052.
22. Shang, T.; Chen, R.; Zhang, J.; Wang, Y. Estimation of soil organic matter content in Yinchuan Plain based on fractional differential combined spectral index. *J. Appl. Ecol.* **2023**, *34*, 717–725.
23. Szczepanek, R. Daily Streamflow Forecasting in Mountainous Catchment Using XGBoost, LightGBM and CatBoost. *Hydrology* **2022**, *9*, 226. [CrossRef]
24. Hao, R.; Bai, Z. Comparative Study for Daily Streamflow Simulation with Different Machine Learning Methods. *Water* **2023**, *15*, 1179. [CrossRef]
25. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
26. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
27. Chang, C.-C.; Lin, C.-J. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27. [CrossRef]
28. Shi, J.; Jin, S.; Hu, C. Analysis of peak flow characteristics change in the Bahe River Basin. *Shaanxi Water Resour.* **2023**, 58–61. [CrossRef]
29. Ke, X.; Wang, N. Comparative study on runoff variation laws in typical north and south Qinling basins. *J. Xi'an Univ. Technol.* **2019**, *35*, 452–458. [CrossRef]
30. Kotthoff, L.; Thornton, C.; Hoos, H.H.; Hutter, F.; Vanschoren, J. *Automated Machine Learning: Methods, Systems, Challenges*; Springer: Cham, Switzerland, 2019.
31. Pelikan, M.; Pelikan, M. Bayesian optimization algorithm. In *Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 31–48.
32. Gupta, H.V.; Kling, H.; Yilmaz, K.K.; Martinez, G.F. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *J. Hydrol.* **2009**, *377*, 80–91. [CrossRef]
33. Yilmaz, K.K.; Gupta, H.V.; Wagener, T. A process-based diagnostic approach to model evaluation. Application to the NWS distributed hydrologic model. *Water Resour. Res.* **2008**, *44*. [CrossRef]
34. Fu, X.; Kan, G.; Liu, R.; Liang, K.; He, X.; Ding, L. Research on Rain Pattern Classification Based on Machine Learning: A Case Study in Pi River Basin. *Water* **2023**, *15*, 1570. [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.