



Article

Application of a New Hybrid Deep Learning Model That Considers Temporal and Feature Dependencies in Rainfall–Runoff Simulation

Feng Zhou, Yangbo Chen * and Jun Liu

School of Geography and Planning, Sun Yat-sen University, Guangzhou 510275, China

* Correspondence: eescyb@mail.sysu.edu.cn

Abstract: Runoff forecasting is important for water resource management. Although deep learning models have substantially improved the accuracy of runoff prediction, the temporal and feature dependencies between rainfall–runoff time series elements have not been effectively exploited. In this work, we propose a new hybrid deep learning model to predict hourly streamflow: SA-CNN-LSTM (self-attention, convolutional neural network, and long short-term memory network). The advantages of CNN and LSTM in terms of data extraction from time series data are combined with the self-attention mechanism. By considering interdependences of the rainfall–runoff sequence between timesteps and between features, the prediction performance of the model is enhanced. We explored the performance of the model in the Mazhou Basin, China; we compared its performance with the performances of LSTM, CNN, ANN (artificial neural network), RF (random forest), SA-LSTM, and SA-CNN. Our analysis demonstrated that SA-CNN-LSTM demonstrated robust prediction with different flood magnitudes and different lead times; it was particularly effective within lead times of 1–5 h. Additionally, the performance of the self-attention mechanism with LSTM and CNN alone, respectively, was improved at some lead times; however, the overall performance was unstable. In contrast, the hybrid model integrating CNN, LSTM, and the self-attention mechanism exhibited better model performance and robustness. Overall, this study considers the importance of temporal and feature dependencies in hourly runoff prediction, then proposes a hybrid deep learning model to improve the performances of conventional models in runoff prediction.



Citation: Zhou, F.; Chen, Y.; Liu, J. Application of a New Hybrid Deep Learning Model That Considers Temporal and Feature Dependencies in Rainfall–Runoff Simulation. *Remote Sens.* **2023**, *15*, 1395.

Academic Editors: Pingping Luo, Van-Thanh-Van Nguyen, Binaya Kumar Mishra, Ahmed Elbeltagi, Reza Hassanzadeh and Baofu Li

Received: 7 January 2023

Revised: 23 February 2023

Accepted: 28 February 2023

Published: 1 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

Floods are among the most common natural disasters, and extreme precipitation is often the main cause of flood disasters. Thus, the rainfall–runoff model is important for studies of basin hydrological processes with the goal of preventing flood disasters [1–3]. However, runoff prediction is highly complex, dynamic, and unstable; thus, it remains challenging for hydrologists to achieve efficient and accurate runoff prediction [4–6]. Models for runoff prediction can be divided into two categories: physical process-based and data-driven [7,8].

Physical process-based models are mathematical models that systematically describe the hydrological processes of a basin. They can be subdivided into lumped and distributed types. Lumped models generalize and assume the hydrological processes of a basin. Such models perform calculations for the entire basin but they do not consider spatial heterogeneity with respect to climate and underlying surface factors. Examples of lumped models include HEC-HMS [9,10], HBV [11], and XAJ [12]. Distributed models solve the problem of homogenization of watershed physical characteristics that exists in lumped models. The watershed is divided into fine meshes for calculations of runoff generation and concentration that can fully reflect the watershed physical characteristics and rainfall spatial distribution. Examples of distributed models include SWAT [13,14], SHE [15,16],

and Liuxihe [17–19]. Although distributed hydrological models have been applied to watershed hydrological forecasting, they have some limitations. These models attempt to simulate physical mechanisms of the hydrological process, thus improving interpretability; however, the increase in model complexity creates a need for additional input data and physical parameters, which leads to poor model portability and an increasing need for high-precision physical property data [20]. Additionally, some processes of the hydrological cycle cannot be accurately described, leading to necessary simplifications in the hydrological model calculation process and the inevitable introduction of simulation error [21,22].

Compared with process-based hydrological models, data-driven models are completely based on internal relationships between the data fitting input and the output; because such models lack a clear understanding of the internal physical mechanisms, they avoid generalization of the hydrological runoff generation and concentration processes [23–25]. These models have been widely used for runoff prediction. In early research regarding data-driven hydrological models, ARMA and its improved models were often used to predict runoff time series [26,27]. However, because these models are linear, they have considerable limitations in complex hydrological applications. Nonetheless, advances in machine learning allowed some nonlinear models to become popular. In 1986, Rumelhart proposed a backpropagation algorithm to train the network, which solved the problem of artificial neural network (ANN) parameter training [28]. Thus, ANN models have been widely used in hydrological forecasting [29,30]. However, ANNs have problems such as overfitting, gradient disappearance, and sequence information loss; accordingly, these simple models are inappropriate for complex hydrological processes [31]. With the exponential growth of sample data and improvements in computing power, complex deep learning models (e.g., convolutional neural network [CNN] and long short-term memory network [LSTM]) have demonstrated strong capabilities and robust performance in the field of hydrology. Because of their excellent feature extraction abilities, CNNs can extract repetitive patterns hidden in hydrology time series [32,33]. Recurrent neural networks (RNNs) transform hidden nodes into a cyclic structure, serialize the input data when learning nonlinear relationships between input and output, and adequately capture temporal dynamics [34]. However, RNNs are susceptible to problems of gradient disappearance or gradient explosion during backpropagation. A variant of the RNN approach, LSTM models use a gate mechanism and introduce memory cells to store long-term memory, thereby solving the RNN bottleneck involving long-term dependence; accordingly, LSTMs have been studied in the field of hydrology [35–37]. Kratzert et al. tested the effect of LSTM in ungauged watersheds using K-fold cross-validation in the CAMELS dataset. The experiment showed that the data-driven model had better performance than the traditional physical model under out-of-sample conditions [38]. Mao et al. proved the importance of hydrological hysteresis to LSTM runoff simulation by experiments [39].

In recent years, with the continuous development of research on rainfall–runoff data-driven models, some ensemble models or hybrid models began to appear. Francesco et al. proposed a novel simpler model based on the stacking of the Random Forest and Multilayer Perceptron algorithms, which performed similar to the excellent performance of the bidirectional LSTM, but had the notable advantage of much shorter computation times than the bidirectional LSTM [40]. Kao et al. proposed a Long Short-Term Memory based Encoder-Decoder (LSTM-ED), and compared it with a feed forward neural network-based Encoder-Decoder (FFNN-ED) model to prove the superiority of the model for multi-step-ahead flood forecasting [41]. Cui et al. developed a data-driven model by integrating singular spectrum analysis (SSA) and a light gradient boosting machine (LightGBM) to achieve the high-accuracy, real-time prediction of regional urban runoff [42]. Yin et al. proposed a novel data-driven model named LSTM-based multi-state-vector sequence-to-sequence (LSTM-MSV-S2S) rainfall–runoff model, which contained m multiple state vectors for m -step-ahead runoff predictions, and compared it with two LSTM-S2S models by testing them on CAMELS data set. The results showed that LSTM-MSV-S2S model has better performance [43]. Chang et al. proposed a novel urban flood forecast methodology

framework by integrating the advantages of Principal Component Analysis (PCA), Self-Organizing Map (SOM), and Nonlinear Autoregressive with Exogenous Inputs (NARX) to mine the spatial-temporal features between rainfall patterns and inundation maps for making multi-step-ahead regional flood inundation forecasts [44].

Although the new proposed hybrid rainfall-runoff models constantly has greatly improved the accuracy of runoff prediction, previous research regarding data-driven models has not effectively used temporal and feature dependencies between rainfall-runoff time series elements. Temporal dependencies reflect the importance of variation in time series data according to time frequency. Additionally, rainfall station observations, an important input feature, often indicate that rainfall frequency is closer when the distance is smaller. To improve model accuracy, temporal and feature dependencies should be exploited, rather than ignored [45–47].

In this study, we consider temporal and feature dependencies. CNN and LSTM models are used to extract the characteristics of rainfall-runoff, and explore the temporal and feature dependencies of rainfall-runoff input with the self-attention mechanism. We propose a new hybrid streamflow prediction model, namely SA-CNN-LSTM, that considers interdependences between timesteps and features of the rainfall-runoff series; this model can improve the performance of hourly predicted runoff.

2. Methodologies

2.1. Self-Attention Mechanism

The attention mechanism is currently widely used in various deep learning tasks, such as natural language processing, image recognition, and speech recognition [48–50]. In accordance with the selective attention aspect of human vision, it considers extensive available information, then selects information that is more relevant to the current task goal by assigning different weights [51]. In this study, the self-attention mechanism was used in the output matrix of LSTM and CNN to extract interdependences between timesteps and features. The attention mechanism can be regarded as the contribution of source to target, which represents the mapping of a query vector to a series of key-value pair vectors [52]. The process can be divided into three steps: (1) calculation of similarity between query and key vectors to obtain target weight; (2) normalization of weight; and (3) multiplication of the value vector by the weight vector to assign weight. The special case in which the source equals the target constitutes the self-attention mechanism.

Suppose that an output vector $h \in R^{m \times k}$ of LSTM is selected as the input of the self-attention mechanism. In the expression, m represents the number of features in a single timestep, and k represents the number of timesteps.

$$h_{m \times k} = [h_m^{(1)}, h_m^{(2)}, \dots, h_m^{(i)}, \dots, h_m^{(k)}] \quad (1)$$

The attention weight α of each hidden layer output vector is computed by

$$S_i = F(W_q h_m^{(i)}, W_k h_m^{(i)}) \quad (2)$$

$$\alpha_i = \text{Softmax}(S_i) = \frac{\exp(S_i)}{\sum_{j=1}^k \exp(S_j)} \quad (3)$$

$$V = [W_v h_m^{(1)} * \alpha_1, W_v h_m^{(2)} * \alpha_2, \dots, W_v h_m^{(i)} * \alpha_i, \dots, W_v h_m^{(k)} * \alpha_k] \quad (4)$$

where W_q , W_k , and W_v are the query, key, and value weight matrix, respectively (all obtained by training); F is usually a scaled dot product; α_i is the weight of timestep i ; and V represents the result of assigning weight to each timestep.

2.2. Neural Network

2.2.1. Long Short-Term Memory Network

LSTM models are RNN variants that add or delete time series memory information by adjusting gate states [53]. The LSTM unit calculation process contains three inputs: the memory cell state (C_{t-1}) and hidden state (h_{t-1}), passed down from the previous timestep, respectively, which represent long-term memory and short-term memory, as well as the current timestep, X_t . According to X_t and h_{t-1} , there are four internal states, z_i , z_f , z_o , and z , which are estimated as follows:

$$z = \tanh(wx_t + wh_{t-1} + b) \quad (5)$$

$$z_i = \text{Sigmoid}(w_i x_t + w_i h_{t-1} + b_i) \quad (6)$$

$$z_f = \text{Sigmoid}(w_f x_t + w_f h_{t-1} + b_f) \quad (7)$$

$$z_o = \text{Sigmoid}(w_o x_t + w_o h_{t-1} + b_o) \quad (8)$$

In Equations (5)–(8), w and b represent the weight and the deviation, respectively; \tanh and sigmoid represent the activation function; and z_i , z_f , z_o , and z represent the input gate, forget gate, output gate, and input information, respectively. Figure 1 shows the internal structure of the LSTM, which mainly progresses through three stages:

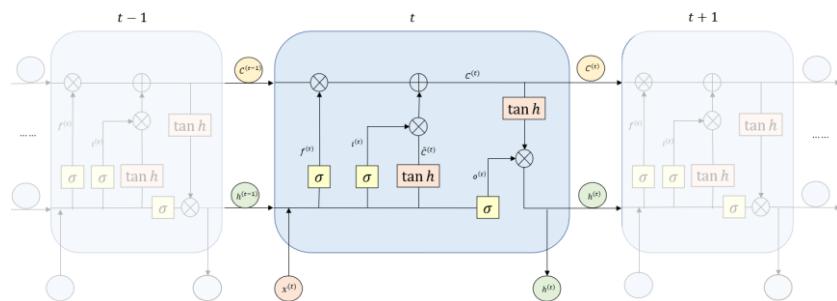


Figure 1. LSTM structure.

Forget stage: z_f , as the forget gate signal, controls the C_{t-1} passed down from the last timestep to perform selective forgetting.

Memory stage: z_i , as the input gate signal, selectively remembers from the current timestep x_t .

Output stage: z_o , as the output gate signal, selectively outputs from the updated memory cell state C_t to obtain the hidden state h_t at the current timestep.

The mathematical expressions of processes in the three stages are as follows:

$$C_t = z_f \odot C_{t-1} + z_i \odot z \quad (9)$$

$$h_t = z_o \odot \tanh C_t \quad (10)$$

$$y_t = \text{sigmoid}(w_t h_t) \quad (11)$$

where C_t and h_t represent the updated memory cell state and updated hidden state in the current timestep, respectively; y_t is an output of the current timestep. \odot represents matrix multiplication.

2.2.2. Convolutional Neural Network

CNNs are deep neural network models based on the cognitive mechanism of natural biological vision; they are particularly successful in image recognition [54] and object detection [55]. Sparse connection and parameter sharing are the most prominent features of CNNs, which have excellent feature extraction capabilities that distinguish them from con-

ventional ANN models [56]. CNNs extract features through convolution operations, which usually scan entire datasets according to input order, as well as preset convolution core size, convolution stride size, and padding size. At each step of scanning, the convolution kernel and the input data of corresponding rows and columns are used for matrix multiplication and matrix addition. CNN-based time series prediction methods all use one-dimensional convolution operations to study the trend of the time series itself. The one-dimensional convolution operation process focused on time series data is shown in Figure 2.

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p
q	r	s	t

 \times

α	β	γ	δ
ε	ϵ	ζ	η

 $=$

$aa\alpha + b\beta + c\gamma + d\delta + e\varepsilon + f\epsilon + g\zeta + h\eta$
$e\alpha + f\beta + g\gamma + h\delta + i\varepsilon + j\epsilon + k\zeta + l\eta$
$i\alpha + j\beta + k\gamma + l\delta + m\varepsilon + n\epsilon + o\zeta + p\eta$
$m\alpha + n\beta + o\gamma + p\delta + q\varepsilon + r\epsilon + s\zeta + t\eta$

5×4 2×4 4×1

Figure 2. One-dimensional convolution operation process.

2.3. SA-CNN-LSTM Model

The SA-CNN-LSTM model integrates the excellent abilities of CNNs and LSTMs to extract feature patterns. The self-attention mechanism is used to learn temporal and feature dependencies from the output feature matrix, then assign weights. The model mainly progresses through two stages of processing from data input to output. In the first stage, the input matrix is sent to the LSTM to obtain information regarding each timestep; the transposed input matrix (timestep dimension and feature dimension transposition) is sent to the CNN for a one-dimensional convolution operation that yields information regarding each input feature. The self-attention mechanism is used to assign weights to features extracted by the two models. This mechanism assigns weights to the output matrices of the two models in their respective timesteps and feature dimensions. In the second stage, in addition to the above nonlinear relationship, the model adds an autoregressive layer to consider the linear relationship with respect to runoff [57,58]. Finally, the output results of the two stages are summed and input into the feedforward neural network to yield the final prediction results. A flowchart of the SA-CNN-LSTM is shown in Figure 3. In the Figure, the input matrix is a batch of data samples. A single input sample is $Input^{(1)} = [X^{(1)}, X^{(2)}, \dots, X^{(t)}, \dots, X^{(T)}, Q^{(1)}]$, where X and Q represent meteorological and hydrological stations data, respectively, and T is the timestep. $X = [X_{(1)}, X_{(2)}, X_{(3)}, \dots, X_{(T)}]$ is the historical rainfall sequence of T timestep, and $Q = [Q_{(1)}, Q_{(2)}, Q_{(3)}, \dots, Q_{(T)}]$ is the historical runoff sequence of T timestep. The output matrix is a batch of prediction results. A single output sample is $Output^{(1)} = [Y^{(1)}, Y^{(2)}, \dots, Y^{(m)}]$, where $Y^{(m)}$ represents the result of predicted runoff in the next m hours.

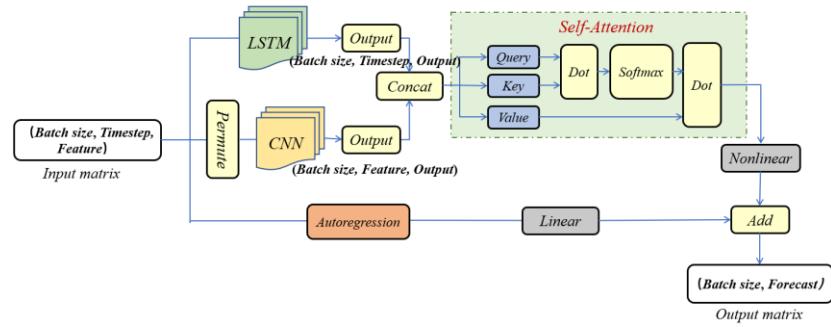


Figure 3. Flowchart of SA-CNN-LSTM model.

2.4. Network Structure Optimization

The time of confluence at the outlet point is closely related to the physical characteristics of the basin. To determine the timestep of the input model, the Pearson correlation coefficient is used to calculate the correlation between rainfall and runoff. The calculation formula is as follows:

$$R(\delta) = \frac{\sum_i^N [(F_{i+\delta} - \bar{F}) * (R_i - \bar{R})]}{\sqrt{\sum_i^N (F_{i+\delta} - \bar{F})^2} * \sqrt{\sum_i^N (R_i - \bar{R})^2}} \quad (12)$$

where F , \bar{F} , R , and \bar{R} represent runoff, average runoff, precipitation, and average precipitation, respectively; δ represents the offset between runoff and rainfall time. The correlation coefficient curves of different offsets are shown in Figure 4. The impact of hourly rainfall on change in runoff first increases and then decreases; it eventually reaches the same magnitude as the starting point after 60 h, indicating that hourly rainfall can effectively influence the change in runoff in the basin over the next 60 h.

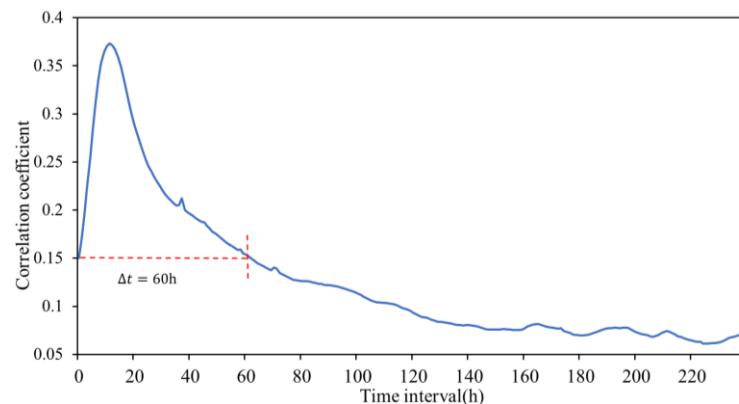


Figure 4. Correlation coefficients between rainfall and runoff.

Based on the results of the correlation coefficient curve calculated above, we set the timestep to 60 h for prediction of runoff at the outlet of the basin in the next 7 h. The Adam optimizer was chosen as the model optimization algorithm [59,60], and mean square error was regarded as the loss function. Other super-parameters of the model (e.g., LSTM hidden layer dimension, CNN layer dimension, autoregression layer dimension, attention layer dimension, and learning rate) need to be set in advance before training. The parameter optimization of the deep learning model has always been a very complex problem. In this experiment, we choose to use the Hyperopt library provided by python to help optimize the super-parameters, where `fmin()` is the function to call the parameter optimization. This function mainly contains four parameters. The first parameter is to set the target function as MSE, the second parameter is to set the parameter optimization range, as shown in Table 1,

the third parameter is to set the number of iterations as 1000, and the fourth parameter is to set the parameter optimization method as random search method.

Table 1. Ranges of model super-parameters used in optimization.

Super-Parameter	Range
LSTM layer	8–256
Autoregressive layer	8–256
CNN layer	8–256
Attention layer	8–256
Learning rate	[0.1, 0.01, 0.001, 0.0001, 0.00001]

2.5. Evaluation Statistics

In this study, the Nash efficiency coefficient (*NSE*), root mean square error (*RMSE*), mean absolute error (*MAE*), mean relative error (*MRE*), and peak error (*PE*) were used to evaluate the performance of the proposed model.

$$NSE = 1 - \frac{\sum_{i=1}^N (O_i - P_i)^2}{\sum_{i=1}^N (O_i - \bar{O})^2} \quad (13)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (O_i - P_i)^2} \quad (14)$$

$$MAE = \frac{\sum_{i=1}^N |O_i - P_i|}{N} \quad (15)$$

$$MRE = \frac{1}{N} \sum_{i=1}^N \frac{|(O_i - P_i)|}{O_i} \quad (16)$$

$$PE = \frac{|O_f - P_f|}{O_f} \quad (17)$$

where O_i , \bar{O} , P_i , O_f and P_f represent the runoff observation value, observation average value, prediction value, observation peak value, and prediction peak value, respectively. *NSE* is in the range of $-\infty$ to one. When the value of *NSE* is >0.9 , the model fit is perfect. An *NSE* value in the range of 0.8–0.9 is considered indicative of a fairly good model, whereas a value < 0.8 indicates an unsatisfactory model [61]. The *RMSE* represents the deviation between observation data and simulation data; the *MAE* and *MRE* represent the average absolute error and relative error between observation data and simulation data, respectively. When these values are closer to 0, the model simulation value is more consistent with the observed value [62]. *PE* is used to evaluate the prediction accuracy of peak flow in actual flood performance.

3. Case Study

3.1. Study Area and Data

We tested the performance of the SA-CNN-LSTM model in the Mazhou Basin, Jiangxi Province, southeast China, which has a subtropical monsoon climate with high temperature and humidity. From March to June, frontal rain, sometimes persisting for >2 weeks, often occurs because of the influence of cold air moving from north to south. Typhoons, with heavy rainfall intensity, may affect the region from July to September. The Mazhou hydrological station ($25^{\circ}30'54''N$, $115^{\circ}47'00''E$) was established in January 1958. It controls a basin area of 1758 km^2 , and the gradient of the main river is 1.53% . The average annual flow at the Mazhou station is $47.3 \text{ m}^3/\text{s}$, the average annual rainfall is 1560 mm , and the average annual evaporation is 1069 mm . The observation data used in this study include hourly precipitation data of 35 rainfall stations and the hourly runoff data of

the Mazhou hydrological station, both acquired during the period of 2013–2021. The geographical location of the Mazhou watershed and the distribution of each station are shown in Figure 5.

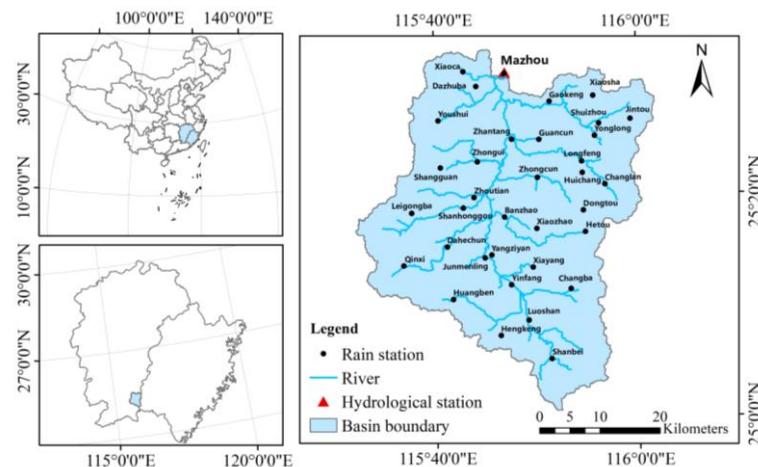


Figure 5. Location of the Mazhou Basin (China) and distribution of stations in the basin.

The dataset was obtained by sliding the window in chronological order through the whole rainfall-runoff sequence. The size of the sliding window was set according to the timestep. The process of dataset preparation is shown in Figure 6. The dataset was divided into training and testing sets, which constituted 80% and 20% of the total records, respectively. The training set was used to calibrate the model parameters, whereas the testing set was used to evaluate model performance. To ensure consistent model input and enhance convergence speed during model training [63], the rainfall and streamflow were normalized before the training process:

$$D_i^{Norm} = \frac{D_i^{Org} - D_{MIN}^{Org}}{D_{MAX}^{Org} - D_{MIN}^{Org}} \quad (18)$$

where D_i^{Org} is the original input data; D^{MAX} and D^{MIN} are the maximum and minimum values of the input data, respectively; and D_i^{Norm} is the normalized input data.

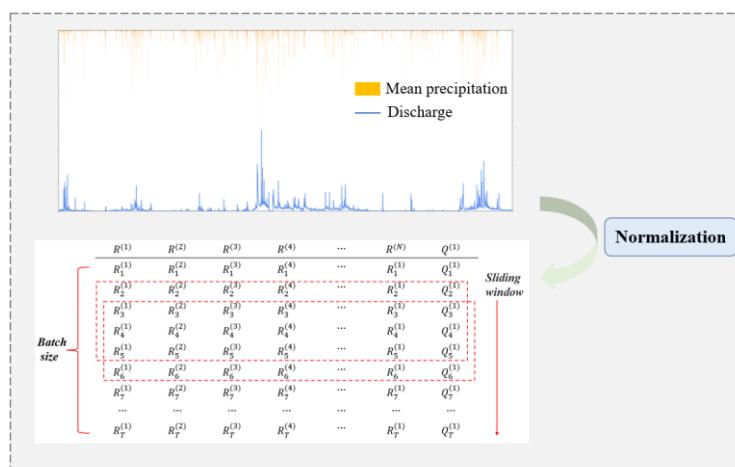


Figure 6. Process of dataset preparation.

3.2. Open-Source Software

This study relied on open-source libraries, including sklearn, NumPy, Math, and Matplotlib. Keras 2.1.5 and Python 3.7 were used to implement deep learning methods. The sklearn library was used to implement the random forest method. Matplotlib was used to draw figures. NumPy and Math were used for matrix operations. All experiments were conducted on a workstation equipped with an Intel E5-2680V3 CPU, 32 GB of RAM, and a NVIDIA GTX Geforce 2080 Ti GPU.

4. Results and Discussion

To explore the performance of the SA-CNN-LSTM model, four forecast models were compared: LSTM, CNN, ANN, and RF (random forest), along with the SA-LSTM and SA-CNN models. Additionally, SA-LSTM and SA-CNN models were used to discuss the impact of temporal and feature dependencies on runoff prediction respectively, and LSTM, CNN, ANN and RF were used as the benchmark models. Among these, the super-parameters of the CNN, LSTM, ANN, SA-LSTM, and SA-CNN models were optimized through the random search method; RF was set as the default recommended parameter.

In Section 4.1, the performance of the SA-CNN-LSTM model is compared with the performances of other models on the basis of different evaluation indicators. Comparisons of scatter plots and streamflow prediction in three categories are shown in Sections 4.2 and 4.3, respectively. In Section 4.4, six selected flood events are used to evaluate the actual performance of the SA-CNN-LSTM model, compared with the performances of other models.

4.1. Comparisons of Evaluation Indicators

Comparisons of model performance according to evaluation indicator are shown in Tables 2 and 3. The testing set result revealed that the prediction accuracy of each model decreased with increasing lead time. The SA-CNN-LSTM exhibited the best performance according to all evaluation indicators at the lead times of 1–5 h. Its performance at the lead times of 6–7 h was similar to the performances of the SA-LSTM and SA-CNN, although it remained better than the performance of the four benchmark models. From the results, the proposed model had better performance with short lead times, and the difference of indicators became smaller as the lead time increased. Notably, at the lead time of 1 h, the RMSE of the SA-CNN-LSTM was 51%, 55%, 61%, and 83% lower than the respective RMSE of the CNN, LSTM, ANN, and RF.

Comparison of performance among the four data-driven models revealed that RF performed the worst at each lead time, followed by ANN. This result indicated that the LSTM and CNN have a stronger ability to extract time series features, compared with ANN and RF, because of their specific network structures. At the lead times of 1–2 h, the predictive ability of the CNN was better than the predictive ability of the LSTM; subsequently, the performance of the LSTM was better. Tian et al. showed that whether a neural network model or a hydrological model is used for hydrological simulation, the basin with high station density will have more abundant data information and more accurate simulation results [64]. These above results may have occurred because the dense rainfall stations ($50.25 \text{ km}^2/\text{station}$) in the Mazhou watershed provided the CNN with rich rainfall features, which enhanced the prediction ability for the short forecast period. The LSTM exhibited outstanding forecast ability in the long forecast period because of its memory advantage in time series data.

Table 2. Performances of each model in the training set.

Table 3. Performances of each model in the testing set.

In respective comparisons with the CNN and LSTM, the SA-CNN and SA-LSTM exhibited varying degrees of improved prediction accuracy with respect to the indicators. The RMSE of the SA-LSTM was 45% lower than the RMSE of the LSTM at the lead time of 1 h; the RMSE of the SA-CNN was 38% lower than the RMSE of the CNN at the lead time of 1 h. As indicated by the performances at different lead times, the SA-LSTM was always better than the LSTM as measured by the evaluation indicators; at the lead time of 7 h, it exhibited the best performance among all models. Notably, temporal dependencies improved the prediction performance of the LSTM, which was consistent with the conclusion of the articles proposed by Gao et al. [46,65] and Liu et al. [66]. However, their paper did not consider the impact of feature dependencies on runoff prediction. The SA-CNN had better prediction performance than the CNN only at the lead times of 1–2 h; it was less effective than the CNN at 3–7 h lead times. These results indicated that feature correlations help the CNN to perform better when predictions are made at a short lead time, but they have a negative effect on the CNN at longer lead times. This result may be related to the lack of memory ability of the CNN in time series, which leads to poor performance in more advanced prediction. However, the introduction of the self-attention mechanism increases model complexity and leads to reduced predictive ability. Based on the above results, temporal and feature dependencies can provide varying degrees of improvement in the predictive performances of conventional data-driven models. The self-attention mechanism has a more robust improvement effect on the LSTM, particularly in longer lead time forecasts; when combined with the self-attention mechanism, the CNN is more appropriate for short lead-time prediction, where it has better performance.

Chen et al. [46] developed the Self-Attentive Long Short-Term Memory (SA-LSTM), which achieved the best performance by comparing the four benchmark models, but the model only considered the temporal dependencies in the rainfall–runoff relationship and ignored importance of the feature dependencies. However, the SA-CNN-LSTM proposed by this study exhibited better comprehensive performance among all models in the prediction evaluation. This finding indicates that feature dependencies can further improve the prediction accuracy of the improved model considering only temporal dependencies, although its best performance only was observed with short lead times.

4.2. Comparison of Scatter Regression Plots

Figure 7 shows the scatter distributions of predicted and observed hourly runoff, along with the respective regression lines of each model, at lead times of 1, 3, 5, and 7 h. The regression line of the SA-CNN-LSTM is nearest to the ideal line (1:1) at different lead times, whereas RF is the model with the greatest deviation from the ideal line. At the lead time of 1 h, the ranking according to the distance from the ideal line is as follows: SA-CNN-LSTM < SA-CNN < SA-LSTM < LSTM < CNN < ANN < RF. This result indicates that the SA-CNN-LSTM model has the best forecasting ability, followed by the SA-CNN and SA-LSTM. However, the regression lines of the SA-CNN and SA-LSTM gradually fluctuate from the ideal line, indicating a lack of stability with increasing lead time. In contrast, the hybrid model integrating the CNN, LSTM, and the self-attention mechanism exhibits more stable performance. This result confirms that, after considering feature and temporal correlation, the combination of CNN, LSTM, and the self-attention mechanism has better robustness in terms of runoff prediction performance. The comparison of regression results further proves the importance of temporal and feature dependencies in improving runoff prediction.

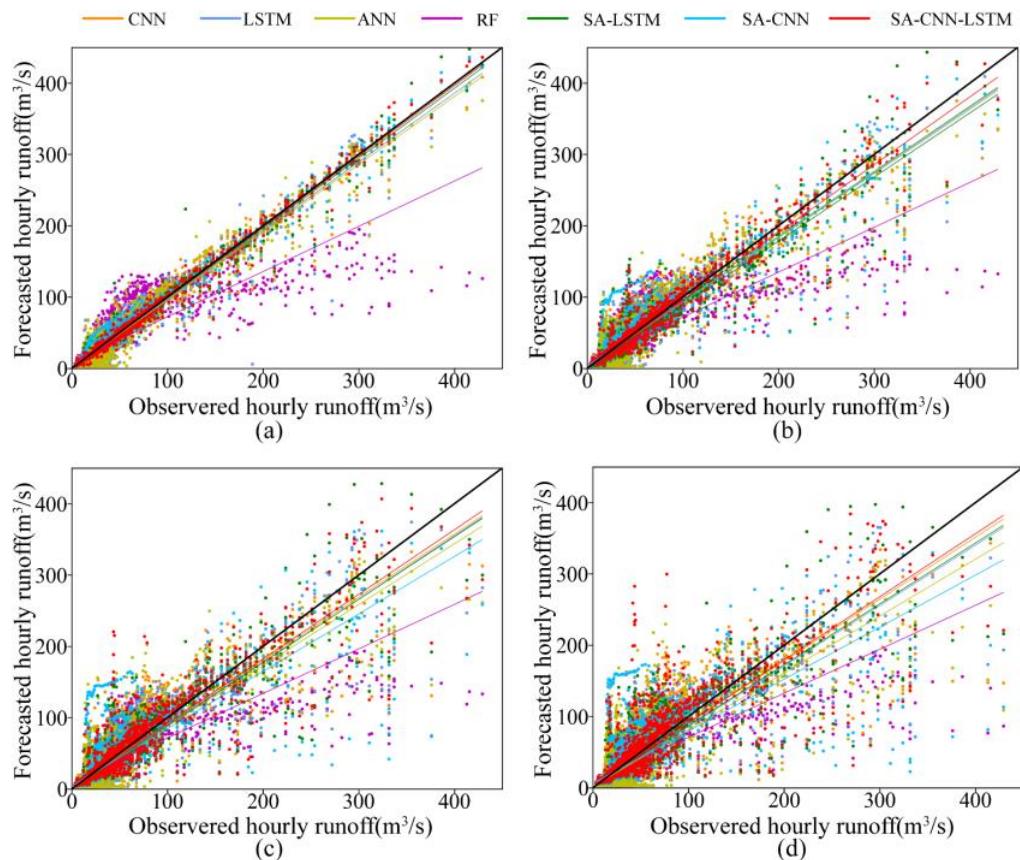


Figure 7. Scatter plots and fitting lines of prediction results for each model at lead times of 1 (a), 3 (b), 5 (c), and 7 (d) h.

4.3. Comparisons of Streamflow in Three Categories

In terms of hydrology, high flow is important for flood forecasting, which can predict the impacts of potential floods and reduce the risks of economic loss. Low flow can affect the sustainable water supply for agricultural departments or groundwater recharge [67]. We divided streamflow into three categories according to flow value: low ($0\text{--}50\text{ m}^3/\text{s}$), medium ($50\text{--}250\text{ m}^3/\text{s}$), and high ($>250\text{ m}^3/\text{s}$). We evaluated the performance of the SA-CNN-LSTM model with respect to prediction of streamflow in the different categories. Figure 8 shows the comparison of prediction results among models, according to RMSE, for the prediction periods of 1, 3, 5, and 7 h. The SA-CNN-LSTM exhibited excellent performance at various streamflow values. With increasing lead time, the model exhibited some degradation at low and medium flow values, but it maintained good performance at high flow values. These results indicate that the model is more stable with increasing lead time at high runoff flow values.

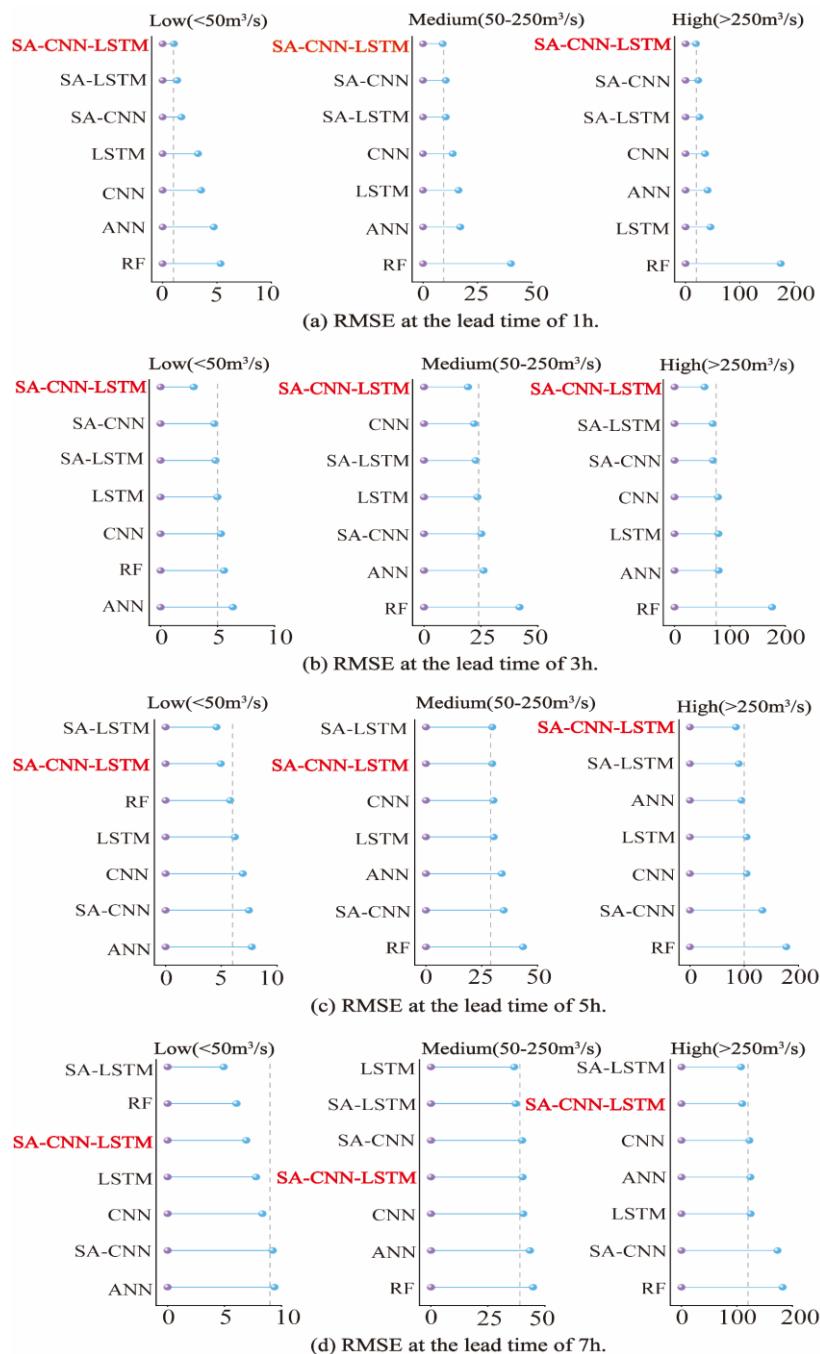


Figure 8. Comparisons of streamflow predictions in three categories of flow for each model at lead times of 1, 3, 5, and 7 h.

4.4. Comparisons of Performance during Actual Flood Events

As shown in Figure 9, six flood events were selected for evaluation of the SA-CNN-LSTM model's performance during actual flood events; these events were 2020040200, 2020042100, 2020060800, 2020081109, 2021050900, and 2021051612. In addition, the evaluation indicators of these six floods were shown in Table 4, where PE was used to evaluate the performance of the flood peak predicted by the model. SA-CNN-LSTM showed the best performance in six flood events, and NSE of five floods was above 0.9. For the prediction of runoff peak, the PEs of SA-CNN-LSTM in six flood events were all within 3%, indicating that the flood peak prediction was also effective.

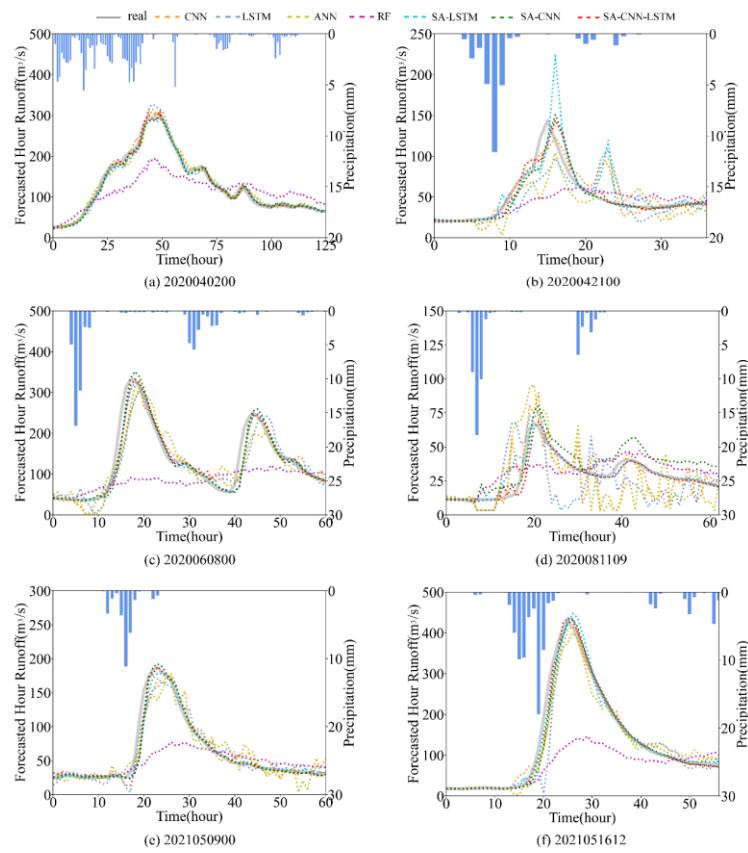


Figure 9. Comparisons of the predictions from each model with a lead time of 1 h.

Table 4. Comparisons of evaluation indicators of six flood events from each model at the lead time of 1 h.

Flood	SA-CNN-LSTM	SA-LSTM	SA-CNN	LSTM	CNN	ANN	RF
2020040200	NSE	0.99	0.99	0.99	0.98	0.98	0.55
	MAE	4.86	5.98	4.88	7.36	6.42	7.71
	MRE	0.04	0.05	0.04	0.06	0.05	0.29
	RMSE	6.49	7.88	6.22	9.72	8.84	10.31
	PE (305.6 m³/s)	1.7% (311.04)	5% (290.14)	2.2% (298.82)	7% (327.15)	1.6% (310.54)	3% (314.78)
	NSE	0.87	0.31	0.85	0.45	0.75	0.21
2020042100	MAE	5.97	11.98	5.92	13.22	10.04	15.72
	MRE	0.09	0.19	0.08	0.23	0.22	0.29
	RMSE	10.56	24.63	11.57	22.02	14.90	23.89
	PE (144 m³/s)	0.1% (144.21)	55% (223.7)	4.6% (150.76)	24.6% (108.49)	14.4% (123.2)	27.7% (104.01)
	NSE	0.94	0.93	0.93	0.80	0.87	0.79
	MAE	11.80	12.32	12.65	23.54	18.55	26.68
2020060800	MRE	0.08	0.08	0.09	0.20	0.17	0.26
	RMSE	21.07	22.04	22.46	37.25	29.97	37.97
	PE (331.6 m³/s)	0.2% (332.55)	1.9% (325.18)	5.5% (351.1)	8.2% (304.35)	1% (328.1)	5.7% (312.42)
	NSE	0.94	0.93	0.93	0.80	0.87	-0.15
	MAE	11.80	12.32	12.65	23.54	18.55	26.68
	MRE	0.08	0.08	0.09	0.20	0.17	0.38

Table 4. Cont.

Flood	SA-CNN-LSTM	SA-LSTM	SA-CNN	LSTM	CNN	ANN	RF
2020081109	NSE	0.91	0.83	0.49	-0.87	0.08	-0.83
	MAE	2.14	2.77	8.94	14.82	10.26	14.89
	MRE	0.08	0.13	0.34	0.58	0.40	0.58
	RMSE	4.36	5.86	10.17	19.43	13.61	19.22
	PE	2.1%	0.02%	18.9%	12.8%	33.5%	42%
	(67.33 m ³ /s)	(68.76)	(67.31)	(80.11)	(77.29)	(89.9)	(95.65)
2021050900	NSE	0.96	0.95	0.96	0.94	0.91	0.89
	MAE	4.07	4.77	4.25	8.46	9.94	11.76
	MRE	0.06	0.07	0.06	0.16	0.18	0.22
	RMSE	9.65	10.77	9.91	11.87	14.52	16.38
	PE	1%	4.2%	3.1%	10.9%	4.8%	6.9%
	(186.6 m ³ /s)	(188.58)	(178.61)	(192.41)	(166.17)	(177.52)	(173.74)
2021051612	NSE	0.99	0.98	0.98	0.90	0.96	0.95
	MAE	7.06	8.70	9.27	19.12	13.88	18.65
	MRE	0.06	0.06	0.07	0.15	0.15	0.16
	RMSE	11.77	16.74	17.83	39.45	23.33	28.41
	PE	1.7%	4.3%	1%	1.6%	3%	6.7%
	(429.1 m ³ /s)	(436.48)	(447.88)	(433.61)	(436.55)	(415.83)	(400.31)

According to the flood process in Figure 9, the flood events 2021050900 and 2021051612 revealed that data-driven models exhibited some lag with respect to the observed peak time. However, the lag intervals of the SA-CNN-LSTM, SA-LSTM, and SA-CNN were smaller than the lag intervals of the other models. This finding indicates that the self-attention mechanism can alleviate the time lag in terms of flood peak prediction, which is a problem for conventional data-driven models. Among the models, the SA-CNN-LSTM had the best performance in terms of predicting peak time, further supporting its superiority. The flood event of 2020081109 was a small flood; the prediction results of the SA-CNN-LSTM in terms of flood rise and fall almost fully conformed to the observed values. Other comparison models exhibited poor performance. For the flood event of 2021051612, a high peak-value flood, the SA-CNN-LSTM performed better than other models in terms of flood peak time and peak value prediction. The performances for different magnitudes of flood indicated that the proposed model has good prediction performance in terms of peak line time, peak prediction, and flood rise and fall. In addition to flood peak prediction, base flow prediction is also important. For the flood events of 2021051612, 2021050900, and 2021051612, the SA-CNN-LSTM demonstrated good prediction of base flow, whereas other models exhibited some fluctuation. In summary, the proposed model was superior to other models in the prediction of actual flood events.

In order to verify the flood performance under different lead times, we also selected a new flood event (2021053000) to compare the model performance under different lead time, as shown in Figure 10. SA-CNN-LSTM performed well in simulating the flood peak in four different lead times, while the performance of other models on the peak value decreased with the increase of lead time. Table 5 showed the flood (2021053000) evaluation indicators of each model under different lead times. All the evaluation indicators of SA-CNN-LSTM was the best among all models at the lead times of 1, 3, 5, and 7 h. Especially in the lead time of 7 h, the PE of SA-CNN-LSTM was still within 3%. However, the PE of other comparison models all exceeded 10%. It can be predicted from the results that the proposed model also has certain potential in long lead times.

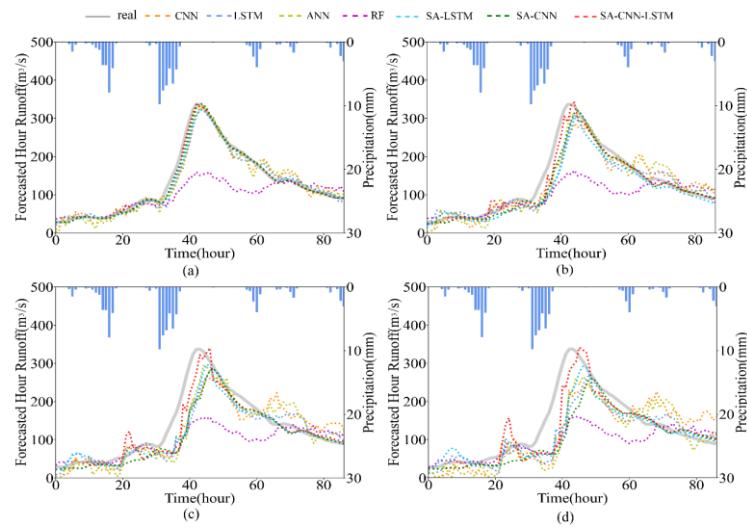


Figure 10. Comparisons of the predictions of the flood event (2021053000) from each model at the lead times of 1 (a), 3 (b), 5 (c), and 7 (d) h.

Table 5. Comparisons of evaluation indicators of flood event (2021053000) from each model at the lead times of 1, 3, 5, and 7 h.

Lead Time	SA-CNN-LSTM	SA-LSTM	SA-CNN	LSTM	CNN	ANN	RF
1 h	NSE	0.99	0.98	0.99	0.96	0.96	0.95
	MAE	4.40	7.89	4.67	11.55	13.49	15.39
	MRE	0.04	0.06	0.04	0.10	0.13	0.16
	RMSE	7.32	12.83	8.10	17.80	18.45	20.18
	PE (336.8 m³/s)	0.01% (336.85)	4.4% (321.98)	0.7% (339.38)	4.1% (322.78)	1.1% (332.89)	1.1% (332.88)
3 h	NSE	0.95	0.83	0.91	0.86	0.85	0.86
	MAE	12.28	23.63	14.26	20.93	24.58	24.67
	MRE	0.11	0.19	0.11	0.17	0.21	0.24
	RMSE	19.56	35.85	25.70	33.07	34.21	32.78
	PE (336.8 m³/s)	1.5% (342.01)	9.8% (303.75)	5% (319.87)	8.9% (306.63)	15% (286.23)	7.7% (310.86)
5 h	NSE	0.86	0.78	0.74	0.71	0.69	0.70
	MAE	20.96	25.13	25.48	29.62	35.92	33.48
	MRE	0.19	0.20	0.19	0.23	0.30	0.29
	RMSE	32.21	40.96	44.70	47.16	48.17	47.69
	PE (336.8 m³/s)	1% (340.22)	11.8% (296.76)	15.1% (285.69)	15.1% (285.90)	21.8% (263.25)	12.6% (294.19)
7 h	NSE	0.77	0.69	0.50	0.55	0.48	0.49
	MAE	27.43	30.08	34.98	35.70	48.89	44.38
	MRE	0.24	0.24	0.24	0.26	0.41	0.39
	RMSE	42.19	48.81	61.41	58.56	63.02	62.06
	PE (336.8 m³/s)	1.3% (341.42)	13.4% (291.59)	22.5% (260.80)	19.8% (269.93)	27.6% (243.62)	21.4% (264.51)

5. Conclusions

Previous research regarding data-driven models did not effectively exploit temporal and feature dependencies between time series elements. Therefore, we proposed a hybrid model, SA-CNN-LSTM, to improve streamflow prediction by considering temporal and feature dependencies. Our model combines the respective advantages of the CNN and LSTM to extract the characteristics of rainfall runoff; it uses a self-attention mechanism to improve performance and robustness by considering interdependence among timesteps

and features. We tested the performance of the proposed model in the Mazhou Basin of China. To explore the performance of the SA-CNN-LSTM model, the LSTM, CNN, ANN, RF, SA-LSTM, and SA-CNN models were compared at lead times of 1–7 h. The SA-LSTM and SA-CNN models were used to discuss the impact of temporal and feature dependencies on runoff prediction respectively, and LSTM, CNN, ANN, and RF were used as the benchmark models.

The experimental results indicated that the SA-CNN-LSTM was better than the four benchmark models within the lead times of 1–7 h, especially in the lead times of 1–5 h, it had the best performance among all the comparison models, which is measured by various evaluation indicators. In addition, the base flow, peak value, and peak time flood prediction performances were partially improved. It can be proved that the combination of CNN, LSTM, and self-attention mechanism can more fully extract the abilities of input features, thereby improving prediction performance and model robustness. Respectively, temporal and feature dependencies can provide some degree of improvement regarding the predictive performances of conventional data-driven models (LSTM, CNN). The self-attention mechanism has a more robust improvement effect on the LSTM, particularly in forecasts with a longer lead time; when combined with the self-attention mechanism, the CNN is more appropriate for short lead-time prediction, where it has better performance. However, this study did not include an evaluation of model interpretability and cannot confirm the advantages of the model with respect to interpretability theory. In future research, we plan to focus on the combination of interpretability and the rainfall-runoff model.

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

Funding: This research was funded by the Science and Technology Program of Guangdong Province (no. 2020B1515120079), and the National Natural Science Foundation of China (NSFC) (no. 51961125206).

Data Availability Statement: Data sharing is not applicable.

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

References

1. Liu, J.; Yuan, X.; Zeng, J.; Jiao, Y.; Li, Y.; Zhong, L.; Yao, L. Ensemble streamflow forecasting over a cascade reservoir catchment with integrated hydrometeorological modeling and machine learning. *Hydrol. Earth Syst. Sci.* **2022**, *26*, 265–278. [[CrossRef](#)]
2. Yin, H.; Wang, F.; Zhang, X.; Zhang, Y.; Chen, J.; Xia, R.; Jin, J. Rainfall-runoff modeling using long short-term memory based step-sequence framework. *J. Hydrol.* **2022**, *610*, 127901. [[CrossRef](#)]
3. Luo, P.; Liu, L.; Wang, S.; Ren, B.; He, B.; Nover, D. Influence assessment of new Inner Tube Porous Brick with absorbent concrete on urban floods control. *Case Stud. Constr. Mater.* **2022**, *17*, e01236. [[CrossRef](#)]
4. Nourani, V.; Baghanam, A.H.; Adamowski, J.; Kisi, O. Applications of hybrid wavelet–Artificial Intelligence models in hydrology: A review. *J. Hydrol.* **2014**, *514*, 358–377. [[CrossRef](#)]
5. Hu, R.; Fang, F.; Pain, C.; Navon, I. Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method. *J. Hydrol.* **2019**, *575*, 911–920. [[CrossRef](#)]
6. Luo, P.; Luo, M.; Li, F.; Qi, X.; Huo, A.; Wang, Z.; He, B.; Takara, K.; Nover, D.; Wang, Y. Urban flood numerical simulation: Research, methods and future perspectives. *Environ. Model. Softw.* **2022**, *156*, 105478. [[CrossRef](#)]
7. Lin, Y.; Wang, D.; Wang, G.; Qiu, J.; Long, K.; Du, Y.; Xie, H.; Wei, Z.; Shangguan, W.; Dai, Y. A hybrid deep learning algorithm and its application to streamflow prediction. *J. Hydrol.* **2021**, *601*, 126636. [[CrossRef](#)]
8. Lv, N.; Liang, X.; Chen, C.; Zhou, Y.; Li, J.; Wei, H.; Wang, H. A long Short-Term memory cyclic model with mutual information for hydrology forecasting: A Case study in the xixian basin. *Adv. Water Resour.* **2020**, *141*, 103622. [[CrossRef](#)]
9. Scharffenberg, W.; Harris, J. Hydrologic Engineering Center Hydrologic Modeling System, HEC-HMS: Interior Flood Modeling. In Proceedings of the World Environmental and Water Resources Congress 2008, Honolulu, HI, USA, 12–16 May 2008; pp. 1–3.
10. Chu, X.; Steinman, A. Event and Continuous Hydrologic Modeling with HEC-HMS. *J. Irrig. Drain. Eng.* **2009**, *135*, 119–124. [[CrossRef](#)]
11. Zhang, S.; Zhou, Y. The analysis and application of an HBV model. *Appl. Math. Model.* **2012**, *36*, 1302–1312. [[CrossRef](#)]
12. Ren-Jun, Z. The Xinanjiang model applied in China. *J. Hydrol.* **1992**, *135*, 371–381. [[CrossRef](#)]

13. 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](#)]
14. Zhou, F.; Xu, Y.; Chen, Y.; Xu, C.-Y.; Gao, Y.; Du, J. Hydrological response to urbanization at different spatio-temporal scales simulated by coupling of CLUE-S and the SWAT model in the Yangtze River Delta region. *J. Hydrol.* **2013**, *485*, 113–125. [[CrossRef](#)]
15. Abbott, M.; Bathurst, J.; Cunge, J.; O’Connell, P.; Rasmussen, J. An introduction to the European Hydrological System—Système Hydrologique Européen, “SHE”, 1: History and philosophy of a physically-based, distributed modelling system. *J. Hydrol.* **1986**, *87*, 45–59. [[CrossRef](#)]
16. Abbott, M.B.; Bathurst, J.C.; Cunge, J.A.; O’Connell, P.E.; Rasmussen, J. An introduction to the European Hydrological System—Système Hydrologique Européen, “SHE”, 2: Structure of a physically-based, distributed modelling system. *J. Hydrol.* **1986**, *87*, 61–77. [[CrossRef](#)]
17. Xu, S.; Chen, Y.; Xing, L.; Li, C. Baipenzhu Reservoir Inflow Flood Forecasting Based on a Distributed Hydrological Model. *Water* **2021**, *13*, 272. [[CrossRef](#)]
18. Zhou, F.; Chen, Y.; Wang, L.; Wu, S.; Shao, G. Flood forecasting scheme of Nanshui reservoir based on Liuxihe model. *Trop. Cyclone Res. Rev.* **2021**, *10*, 106–115. [[CrossRef](#)]
19. Chen, Y.; Li, J.; Xu, H. Improving flood forecasting capability of physically based distributed hydrological models by parameter optimization. *Hydrol. Earth Syst. Sci.* **2016**, *20*, 375–392. [[CrossRef](#)]
20. Lees, T.; Buechel, M.; Anderson, B.; Slater, L.; Reece, S.; Coxon, G.; Dadson, S.J. Benchmarking data-driven rainfall-runoff models in Great Britain: A comparison of long short-term memory (LSTM)-based models with four lumped conceptual models. *Hydrol. Earth Syst. Sci.* **2021**, *25*, 5517–5534. [[CrossRef](#)]
21. Yaseen, Z.M.; Sulaiman, S.O.; Deo, R.C.; Chau, K.-W. An enhanced extreme learning machine model for river flow forecasting: State-of-the-art, practical applications in water resource engineering area and future research direction. *J. Hydrol.* **2018**, *569*, 387–408. [[CrossRef](#)]
22. 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](#)]
23. Xu, Y.; Hu, C.; Wu, Q.; Jian, S.; Li, Z.; Chen, Y.; Zhang, G.; Zhang, Z.; Wang, S. Research on particle swarm optimization in LSTM neural networks for rainfall-runoff simulation. *J. Hydrol.* **2022**, *608*, 127553. [[CrossRef](#)]
24. Yokoo, K.; Ishida, K.; Ercan, A.; Tu, T.; Nagasato, T.; Kiyama, M.; Amagasaki, M. Capabilities of deep learning models on learning physical relationships: Case of rainfall-runoff modeling with LSTM. *Sci. Total. Environ.* **2021**, *802*, 149876. [[CrossRef](#)] [[PubMed](#)]
25. Xie, K.; Liu, P.; Zhang, J.; Han, D.; Wang, G.; Sheri, C. Physics-guided deep learning for rainfall-runoff modeling by considering extreme events and monotonic relationships. *J. Hydrol.* **2021**, *603*, 127043. [[CrossRef](#)]
26. Montanari, A.; Rosso, R.; Taqqu, M.S. A seasonal fractional ARIMA Model applied to the Nile River monthly flows at Aswan. *Water Resour. Res.* **2000**, *36*, 1249–1259. [[CrossRef](#)]
27. Wang, W.-C.; Chau, K.-W.; Xu, D.-M.; Chen, X.-Y. Improving Forecasting Accuracy of Annual Runoff Time Series Using ARIMA Based on EEMD Decomposition. *Water Resour. Manag.* **2015**, *29*, 2655–2675. [[CrossRef](#)]
28. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [[CrossRef](#)]
29. Riad, S.; Mania, J.; Bouchaou, L.; Najjar, Y. Predicting catchment flow in a semi-arid region via an artificial neural network technique. *Hydrol. Process.* **2004**, *18*, 2387–2393. [[CrossRef](#)]
30. Khalil, A.F.; McKee, M.; Kembowski, M.; Asefa, T. Basin scale water management and forecasting using artificial neural networks. *JAWRA J. Am. Water Resour. Assoc.* **2005**, *41*, 195–208. [[CrossRef](#)]
31. Gao, S.; Huang, Y.; Zhang, S.; Han, J.; Wang, G.; Zhang, M.; Lin, Q. Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation. *J. Hydrol.* **2020**, *589*, 125188. [[CrossRef](#)]
32. Van, S.P.; Le, H.M.; Thanh, D.V.; Dang, T.D.; Loc, H.H.; Anh, D.T. Deep learning convolutional neural network in rainfall-runoff modelling. *J. Hydroinformatics* **2020**, *22*, 541–561. [[CrossRef](#)]
33. Chen, C.; Hui, Q.; Xie, W.; Wan, S.; Zhou, Y.; Pei, Q. Convolutional Neural Networks for forecasting flood process in Internet-of-Things enabled smart city. *Comput. Networks* **2020**, *186*, 107744. [[CrossRef](#)]
34. Onan, A. Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 2098–2117. [[CrossRef](#)]
35. Liu, D.; Jiang, W.; Mu, L.; Wang, S. Streamflow Prediction Using Deep Learning Neural Network: Case Study of Yangtze River. *IEEE Access* **2020**, *8*, 90069–90086. [[CrossRef](#)]
36. 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](#)]
37. Xiang, Z.; Yan, J.; Demir, I. A Rainfall-Runoff Model With LSTM-Based Sequence-to-Sequence Learning. *Water Resour. Res.* **2020**, *56*, e2019wr025326. [[CrossRef](#)]
38. Kratzert, F.; Klotz, D.; Herrnegger, M.; Sampson, A.K.; Hochreiter, S.; Nearing, G.S. Toward Improved Predictions in Ungauged Basins: Exploiting the Power of Machine Learning. *Water Resour. Res.* **2019**, *55*, 11344–11354. [[CrossRef](#)]
39. Mao, G.; Wang, M.; Liu, J.; Wang, Z.; Wang, K.; Meng, Y.; Zhong, R.; Wang, H.; Li, Y. Comprehensive comparison of artificial neural networks and long short-term memory networks for rainfall-runoff simulation. *Phys. Chem. Earth Parts A/B/C* **2021**, *123*, 103026. [[CrossRef](#)]

40. Granata, F.; Di Nunno, F.; de Marinis, G. Stacked machine learning algorithms and bidirectional long short-term memory networks for multi-step ahead streamflow forecasting: A comparative study. *J. Hydrol.* **2022**, *613*, 128431. [[CrossRef](#)]
41. Kao, I.F.; Zhou, Y.; Chang, L.C.; Chang, F.J. Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting. *J. Hydrol.* **2020**, *583*, 124631. [[CrossRef](#)]
42. Cui, Z.; Qing, X.; Chai, H.; Yang, S.; Zhu, Y.; Wang, F. Real-time rainfall-runoff prediction using light gradient boosting machine coupled with singular spectrum analysis. *J. Hydrol.* **2021**, *603*, 127124. [[CrossRef](#)]
43. Yin, H.; Zhang, X.; Wang, F.; Zhang, Y.; Xia, R.; Jin, J. Rainfall-runoff modeling using LSTM-based multi-state-vector sequence-to-sequence model. *J. Hydrol.* **2021**, *598*, 126378. [[CrossRef](#)]
44. Chang, L.-C.; Liou, J.-Y.; Chang, F.-J. Spatial-temporal flood inundation nowcasts by fusing machine learning methods and principal component analysis. *J. Hydrol.* **2022**, *612*, 128086. [[CrossRef](#)]
45. Ding, Y.; Zhu, Y.; Feng, J.; Zhang, P.; Cheng, Z. Interpretable spatio-temporal attention LSTM model for flood forecasting. *Neurocomputing* **2020**, *403*, 348–359. [[CrossRef](#)]
46. Chen, X.; Huang, J.; Han, Z.; Gao, H.; Liu, M.; Li, Z.; Liu, X.; Li, Q.; Qi, H.; Huang, Y. The importance of short lag-time in the runoff forecasting model based on long short-term memory. *J. Hydrol.* **2020**, *589*, 125359. [[CrossRef](#)]
47. Wang, Y.K.; Hassan, A. Impact of Spatial Distribution Information of Rainfall in Runoff Simulation Using Deep-Learning Methods. *Hydrol. Earth Syst. Sci.* **2021**, *26*, 2387–2403. [[CrossRef](#)]
48. Zheng, Z.; Huang, S.; Weng, R.; Dai, X.-Y.; Chen, J. Improving Self-Attention Networks With Sequential Relations. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 1707–1716. [[CrossRef](#)]
49. Guo, M.-H.; Xu, T.-X.; Liu, J.-J.; Liu, Z.-N.; Jiang, P.-T.; Mu, T.-J.; Zhang, S.-H.; Martin, R.R.; Cheng, M.-M.; Hu, S.-M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [[CrossRef](#)]
50. Yan, C.; Hao, Y.; Li, L.; Yin, J.; Liu, A.; Mao, Z.; Chen, Z.; Gao, X. Task-Adaptive Attention for Image Captioning. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 43–51. [[CrossRef](#)]
51. Huang, B.; Liang, Y.; Qiu, X. Wind Power Forecasting Using Attention-Based Recurrent Neural Networks: A Comparative Study. *IEEE Access* **2021**, *9*, 40432–40444. [[CrossRef](#)]
52. 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](#)]
53. Sherstinsky, A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [[CrossRef](#)]
54. Mei, S.; Ji, J.; Hou, J.; Li, X.; Du, Q. Learning Sensor-Specific Spatial-Spectral Features of Hyperspectral Images via Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4520–4533. [[CrossRef](#)]
55. Li, Q.; Chen, Y.; Zeng, Y. Transformer with Transfer CNN for Remote-Sensing-Image Object Detection. *Remote. Sens.* **2022**, *14*, 984. [[CrossRef](#)]
56. Yao, G.; Lei, T.; Zhong, J. A review of Convolutional-Neural-Network-based action recognition. *Pattern Recognit. Lett.* **2018**, *118*, 14–22. [[CrossRef](#)]
57. Wei, X.; Lei, B.; Ouyang, H.; Wu, Q. Stock Index Prices Prediction via Temporal Pattern Attention and Long-Short-Term Memory. *Adv. Multimedia* **2020**, *2020*, 8831893. [[CrossRef](#)]
58. Shih, S.-Y.; Sun, F.-K.; Lee, H.-Y. Temporal pattern attention for multivariate time series forecasting. *Mach. Learn.* **2019**, *108*, 1421–1441. [[CrossRef](#)]
59. Wilson, A.C.; Roelofs, R.; Stern, M.; Srebro, N.; Recht, B. The Marginal Value of Adaptive Gradient Methods in Machine Learning. *arXiv* **2014**. [[CrossRef](#)]
60. Leslie, N.; Smith, N.T. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. *arXiv* **2017**. [[CrossRef](#)]
61. Shamseldin, A.Y. Application of a neural network technique to rainfall-runoff modelling. *J. Hydrol.* **1997**, *199*, 272–294. [[CrossRef](#)]
62. Krause, P.; Boyle, D.P.; Bäse, F. Comparison of different efficiency criteria for hydrological model assessment. *Adv. Geosci.* **2005**, *5*, 89–97. [[CrossRef](#)]
63. Sergey Ioffe, C.S. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**. [[CrossRef](#)]
64. Tian, Y.; Xu, Y.-P.; Yang, Z.; Wang, G.; Zhu, Q. Integration of a Parsimonious Hydrological Model with Recurrent Neural Networks for Improved Streamflow Forecasting. *Water* **2018**, *10*, 1655. [[CrossRef](#)]
65. Gao, S.; Zhang, S.; Huang, Y.; Han, J.; Luo, H.; Zhang, Y.; Wang, G. A new seq2seq architecture for hourly runoff prediction using historical rainfall and runoff as input. *J. Hydrol.* **2022**, *612*, 128099. [[CrossRef](#)]
66. Liu, Y.; Zhang, T.; Kang, A.; Li, J.; Lei, X. Research on Runoff Simulations Using Deep-Learning Methods. *Sustainability* **2021**, *13*, 1336. [[CrossRef](#)]
67. Li, Y.; Wang, G.; Liu, C.; Lin, S.; Guan, M.; Zhao, X. Improving Runoff Simulation and Forecasting with Segmenting Delay of Baseflow from Fast Surface Flow in Montane High-Vegetation-Covered Catchments. *Water* **2021**, *13*, 196. [[CrossRef](#)]