

Article

Is the LSTM Model Better than RNN for Flood Forecasting Tasks? A Case Study of HuaYuankou Station and LouDe Station in the Lower Yellow River Basin

Yiyang Wang, Wenchuan Wang , Hongfei Zang and Dongmei Xu

College of Water Resources, North China University of Water Resources and Electric Power, Zhengzhou 450046, China; wangyang145623@outlook.com (Y.W.); zh6344@163.com (H.Z.); xudongmei@ncwu.edu.cn (D.X.)

* Correspondence: wangwenchuan@ncwu.edu.cn

Abstract: The long short-term memory network (LSTM) model alleviates the gradient vanishing or exploding problem of the recurrent neural network (RNN) model with gated unit architecture. It has been applied to flood forecasting work. However, flood data have the characteristic of unidirectional sequence transmission, and the gated unit architecture of the LSTM model establishes connections across different time steps which may not capture the physical mechanisms or be easily interpreted for this kind of data. Therefore, this paper investigates whether the gated unit architecture has a positive impact and whether LSTM is still better than RNN in flood forecasting work. We establish LSTM and RNN models, analyze the structural differences and impacts of the two models in transmitting flood data, and compare their performance in flood forecasting work. We also apply hyperparameter optimization and attention mechanism coupling techniques to improve the models, and establish an RNN model for optimizing hyperparameters using BOA (BOA-RNN), an LSTM model for optimizing hyperparameters using BOA (BOA-LSTM), an RNN model with MHAM in the hidden layer (MHAM-RNN), and an LSTM model with MHAM in the hidden layer (MHAM-LSTM) using the Bayesian optimization algorithm (BOA) and the multi-head attention mechanism (MHAM), respectively, to further examine the effects of RNN and LSTM as the underlying models and of cross-time scale bridging for flood forecasting. We use the measured flood process data of LouDe and HuaYuankou stations in the Yellow River basin to evaluate the models. The results show that compared with the LSTM model, under the 1 h forecast period of the LouDe station, the RNN model with the same structure and hyperparameters improves the four performance indicators of the Nash–Sutcliffe efficiency coefficient (NSE), the Kling–Gupta efficiency coefficient (KGE), the mean absolute error (MAE), and the root mean square error (RMSE) by 1.72%, 4.43%, 35.52% and 25.34%, respectively, and the model performance of the HuaYuankou station also improves significantly. In addition, under different situations, the RNN model outperforms the LSTM model in most cases. The experimental results suggest that the simple internal structure of the RNN model is more suitable for flood forecasting work, while the cross-time bridging methods such as gated unit architecture may not match well with the flood propagation process and may have a negative impact on the flood forecasting accuracy. Overall, the paper analyzes the impact of model architecture on flood forecasting from multiple perspectives and provides a reference for subsequent flood forecasting modeling.



Citation: Wang, Y.; Wang, W.; Zang, H.; Xu, D. Is the LSTM Model Better than RNN for Flood Forecasting Tasks? A Case Study of HuaYuankou Station and LouDe Station in the Lower Yellow River Basin. *Water* **2023**, *15*, 3928. <https://doi.org/10.3390/w15223928>

Academic Editor: Marco Franchini

Received: 11 October 2023

Revised: 28 October 2023

Accepted: 30 October 2023

Published: 10 November 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/>).

Keywords: flood forecasting; RNN; LSTM; model interpretability

1. Introduction

Floods are a major global issue that exposes over a billion people around the world to the risk of disasters [1]. For hydrology, reducing the losses caused by flood disasters by improving the accuracy of flood forecasting is a crucial challenge, and hydrological models play a key role in it [2–4].

Hydrological models can be classified into two categories based on their driving mode: data-driven and process-driven [5–7]. Traditional process-driven hydrological models suffer from a limited understanding of the flood process and rely on idealized assumptions and approximations in their construction, which results in drawbacks such as excessive state equations and parameters, discrepancies between the model and reality, and complex and challenging computations. On the other hand, data-driven deep learning models leverage enhanced computing power to achieve powerful fitting ability and can produce accurate predictions of the flood process [8].

Based on their underlying architectures, existing deep learning models can be categorized into convolutional neural network (CNN) models, recurrent neural network (RNN) models, and attention mechanism neural network (AMNN) models [9–11]. Recurrent neural network models in particular are widely adopted in flood forecasting tasks due to their sequential architecture that matches the spatio-temporal distribution characteristics of floods [12,13].

However, upon further reviewing the related research work, we find that most of the current flood forecasting task modeling relies on the LSTM model, which is a variant of the RNN model, and there is a lack of research on the prediction performance of the basic RNN model [14–16]. The LSTM model, as a variant of the RNN model, addresses the gradient issue of the RNN model in long sequence data by employing a gated unit, but the physical mapping mechanism of the gated unit is hard to interpret, which attracts a lot of criticism [2]. In contrast, the basic architecture of the RNN model facilitates the physical interpretation of spatio-temporal units.

Furthermore, model selection should be based on fundamental evaluation measures rather than on the complexity of the model [17]. Beven [18] demonstrated with a case study that models with complex parameters can fit the observed values well in the training and validation periods but encounter over-parameterization issues in the test period. On the other hand, models with few parameters tend to maintain consistent prediction performance across different periods [19,20]. However, regrettably, in flood forecasting tasks, direct comparison studies between the RNN model and the LSTM model are scarce; many scholars assume that LSTM is better than RNN and opt for the LSTM model for modeling [21–23]. However, the advantages and disadvantages of the RNN model and the LSTM model in hydrology are not clear-cut, and this arbitrary selection results in the paucity of relevant research on the RNN model in the flood forecasting direction [24].

The notion that the LSTM model is better than the RNN model originates from fields such as Natural Language Processing (NLP), which claim that the LSTM model excels in complex sequential tasks [25–27]. However, compared with these fields, a flood forecasting task has its differences, and its sequence complexity is much lower than that of tasks such as sentiment analysis. Hochreiter and Schmidhuber [28] proposed in the article the LSTM model that can learn to bridge minimal time lags over 1000 discrete-time steps by enforcing constant error flow through constant error carousels within special units. However, flood forecasting task does not require such a long-time memory at all, which also implies that the long-term memory advantage of the LSTM model is not evident in the flood forecasting task. From this standpoint, it is critical for hydrological research to explicitly compare the performance of the two models in flood forecasting accuracy, which has implications for the underlying model selection in hydrological research.

In order to evaluate the performance of the two models in flood forecasting tasks, we analyze them from various perspectives. First, we use the hydrological information flow within the model architecture to reveal the underlying information mechanism behind the model prediction results. Then, we analyze the specific performance of the model prediction results straightforwardly. We also extend the forecast period time to evaluate the scalability of the model prediction performance. Moreover, taking into account the popular hyperparameter optimization and attention mechanism coupling techniques, we further evaluate the performance of the hyperparameter optimization models and attention mechanism coupling models based on RNN-type models from these two aspects [29,30].

The paper concentrates on the flood forecasting task and assesses the differences between using the RNN model and the LSTM model for flood forecasting. The main contributions of the paper are as follows:

(1) The paper begins with the perspective of the underlying architecture of the model and elucidates the uniqueness of the information flow of the model in the flood forecasting task and its physical level mapping, which offers a structured basis for the subsequent model research selection. At the same time, based on the underlying architecture analysis, the paper evaluates the performance of the model under different forecast periods.

(2) The paper further extends the model performance comparison experiments in two directions of hyperparameter optimization and attention mechanism coupling by employing the Bayesian optimization algorithm and the multi-head attention mechanism coupling model; it also enhances the generality of the research conclusion and furnishes reference for multi-directional flood forecasting modeling.

(3) The analysis of the models in the paper can assist in the selection of basic models for flood forecasting tasks, further improving the accuracy of flood forecasting tasks while saving experimental costs.

The remainder of the paper is organized as follows. In Section 2, the paper describes the structural differences between RNN and LSTM models, introduces the algorithms and attention mechanism structures required by other parts of coupling models, and determines the indicators used to evaluate the model. In Section 3, the paper introduces the basin data used to verify the model conclusion and the methods used to process data. In Section 4, the paper analyzes and discusses the material from different perspectives such as model structure and performance. Finally, the whole paper is summarized.

2. Research Object

2.1. Research Area and Data

To enhance the universality of the research conclusions of the paper, we selected two stations with different underlying surface properties and runoff conditions as the research objects, namely the HuaYuankou station in the mainstream of the Yellow River and the LouDe station in the downstream tributary Dawen River. The data were obtained from the Henan River Bureau and the Shandong Hydrology and Water Resources Bureau of the Yellow River Water Conservancy Commission. The Yellow River is 5464 km long and flows through nine provinces in China. The drainage area is 795,000 km². The downstream Henan and Shandong provinces are flat, and the sediment accumulation forms a suspended river on the ground. The river channel safety directly affects more than 300 million people in the North China Plain. Therefore, it is necessary to establish an accurate flood forecasting model for the river channel [31].

The HuaYuankou station is located in the lower reaches of the mainstream of the Yellow River, 4696 km away from the source. It is a major control point in the Yellow River basin, a drainage area of 730,000 km², accounting for 92% of the Yellow River basin area. The water supply of the HuaYuankou station mainly comes from the upstream river channel, with three inflow stations: XiaoLangdi, WuZhi, and HeiShiguan. In addition, due to the flat river channel and sediment accumulation in the lower reaches of the Yellow River, the riverbed elevation increases, and the rainfall in the station interval makes it difficult to converge into the main river channel of the Yellow River. Therefore, this station no longer takes the interval rainfall as a model input factor. The topography and river conditions of the HuaYuankou station are shown in Figure 1.

Due to the large catchment area and the influence of the upstream reservoir regulation, the flood process of the HuaYuankou station lasts for a long time. The station selected a total of 31,043 h of flood process data from 2015 to 2022 with peak flows greater than 1000 m³/s. All data were divided into a training set, a validation set, and a test set according to the ratio of 70:25:5 (since the flood lasted for a long time, the last flood was chosen as the test set). The data division is shown in Figure 2.

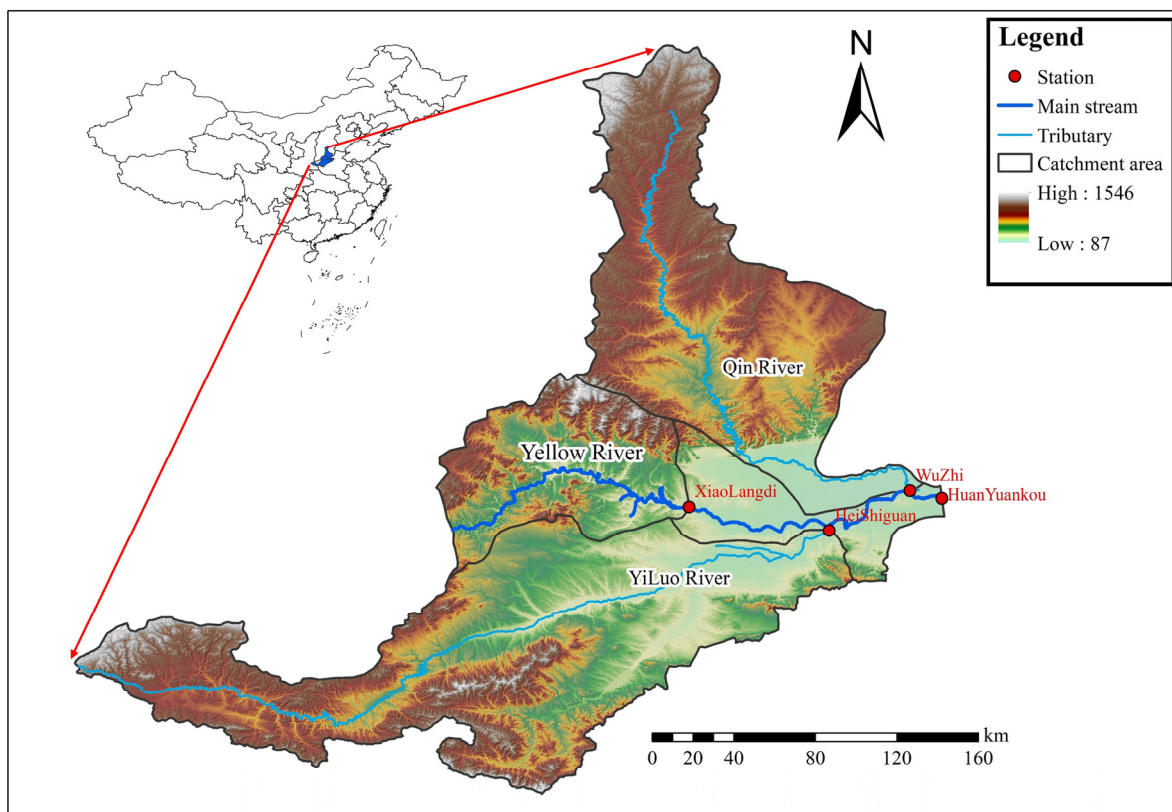


Figure 1. Watershed situation of HuaYuankou station.

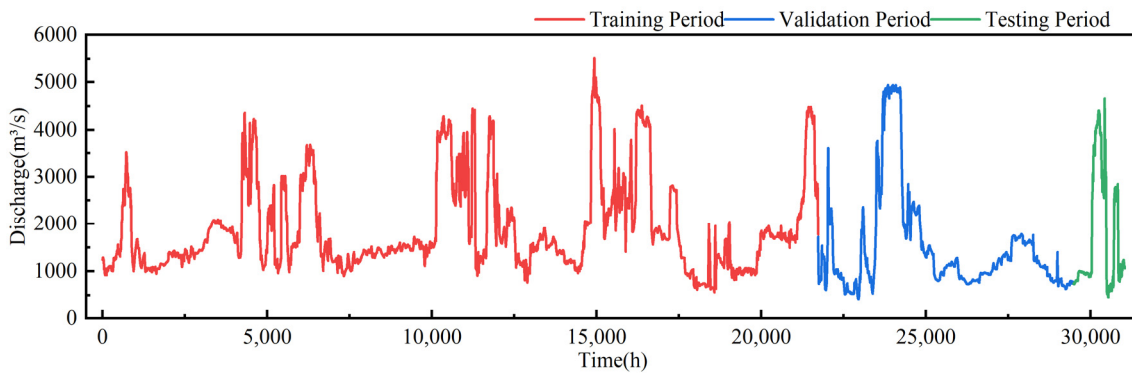


Figure 2. Division of the HuaYuankou Station Dataset.

The LouDe station is located on the Dawen River, the only tributary of the lower Yellow River below the HuaYuankou station (the JinDi River and the natural Wenyan Canal need to be lifted into the Yellow River when necessary due to the elevation effect). It is the control station of the south branch of the Dawen River. The Dawen River originates from the north of the XuanGu Mountain in Shandong Province, with a total length of 209 km and a drainage area of 9098 km². It flows into the Dongping Lake from east to west and then into the Yellow River. Affected by the monsoon climate, the precipitation in the flood season accounts for more than 70% of the year, and the river flow changes are greatly affected by rainfall. Seasonal floods are likely to occur, and the flood confluence and the mainstream overlap may even affect the safety of the main river channel of the Yellow River. The Loude station has two inflow stations: the GuangMing Reservoir and the DongZhou Reservoir, and there are 16 rainfall stations such as XiaFeng and MengYinzhai in the station interval. The topography and river conditions of the HuaYuankou station are shown in Figure 3.

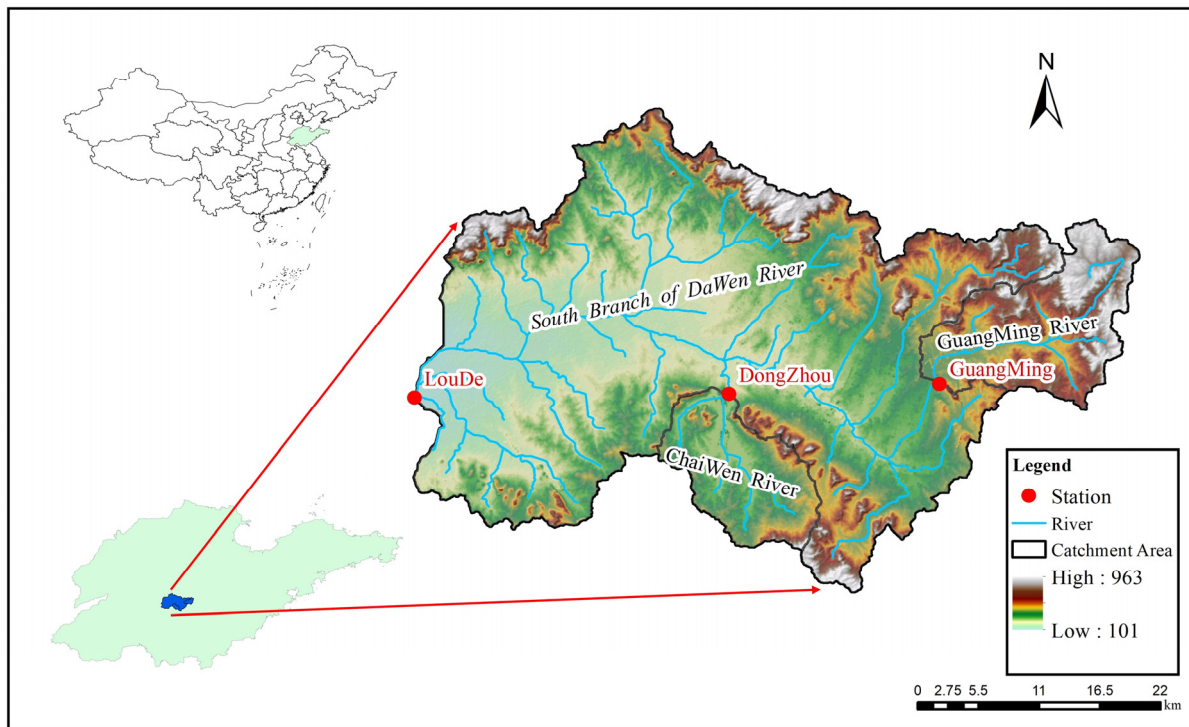


Figure 3. Watershed situation of the LouDe station.

The catchment area of the LouDe station is relatively small, and the flood process of the station mostly comes from the flash flood confluence caused by short-term heavy rainfall, so it lasts for a short time and rises and falls sharply. A total of 4684 h of flood process data were selected, with 22 flood events with instantaneous flow exceeding 200 m³/s. They were divided according to the ratio of 77:16:7, and the last two floods were selected as the test set. The data division is shown in Figure 4.

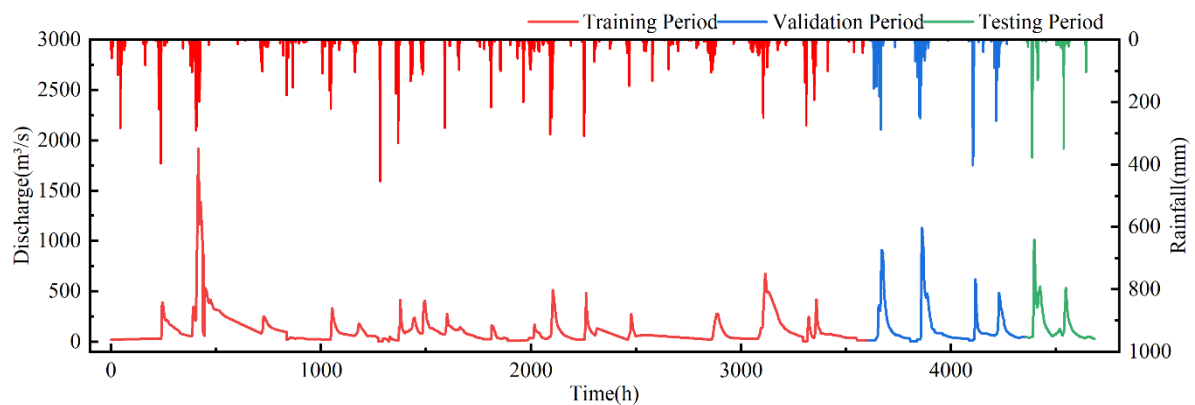


Figure 4. Division of the LouDe Station Dataset.

2.2. Input and Output Sequence Settings

Setting the input and output sequences reasonably can enable the model to obtain more comprehensive data and make accurate predictions. In order to evaluate the model’s learning situation for flood sequence data, the paper ensures that the model obtains all the relevant information that affects the outflow of the station. Considering that the confluence time of the HuaYuankou station is long, the time step of the model input sequence is set to 15 h, which is enough to meet the confluence time requirements of most basins. In order to accurately compare the prediction results of the model, the output sequence is set to one,

that is, to test the single-point prediction ability of the model and avoid the influence of the error weight of each time point in multi-step prediction.

2.3. Research Process

We processed the initial data and obtained two hydrological sequences, with matrix formats of $(31,043 \times 4)$ and (4684×19) , respectively, corresponding to the sequence time length and data factor types. According to Time step 15, we transformed the matrix format and obtained two groups of input and output data for model verification. We input the relevant data into RNN and LSTM models, MHAM-RNN and MHAM-LSTM models, and BOA-RNN and BOA-LSTM models built in Section 3, respectively, and obtained the underlying prediction results of RNN and LSTM models from three perspectives: basic model, hyperparameter optimization model, and attention mechanism coupling model. We analyzed the influence of structure on the model from the change in results and verified it with the model structure analysis. The specific research process is shown in Figure 5.

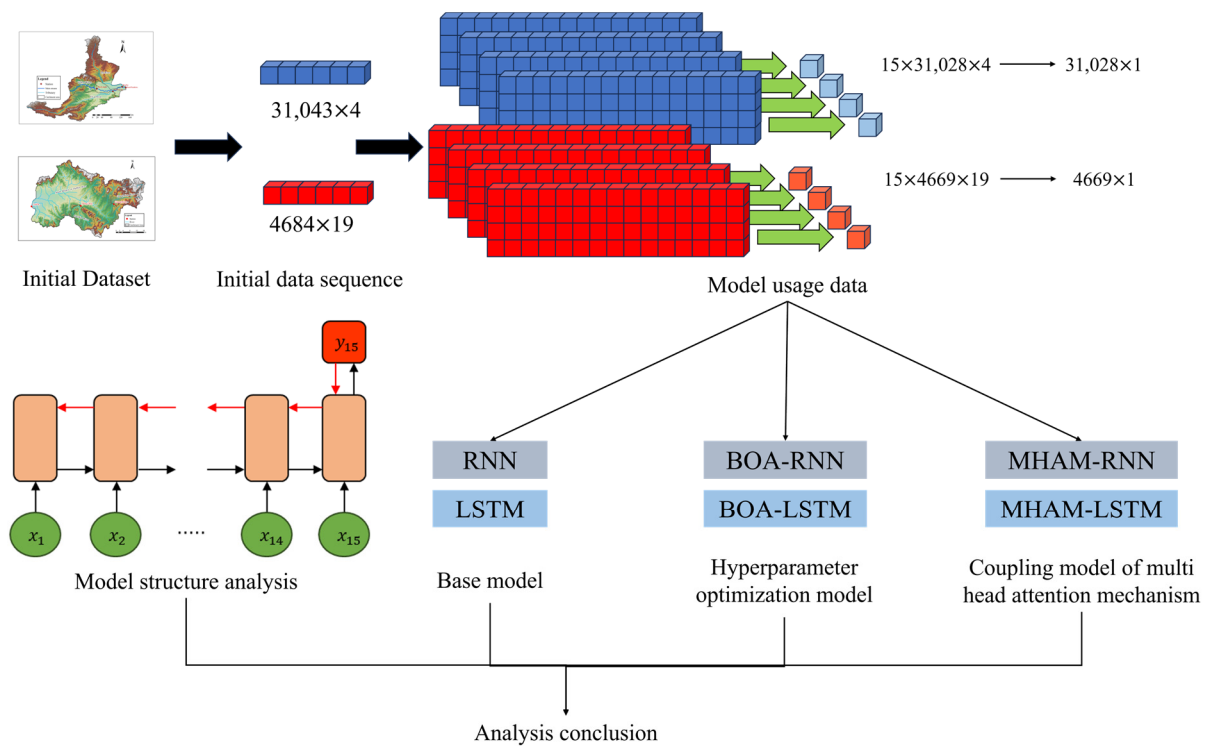


Figure 5. Research Process Diagram (in the figure, 31,043 and 4684 represent the length of the flood sequence time obtained from the HuaYuankou station and the LouDe station; 4 and 19 represent the types of model input factors; 31,028 and 4669 represent the length of the flood sequence used for model training, validation, and testing after removing Time step 15; 1 represents the type of the output target).

3. Methods

To compare the performance of RNN and LSTM models for flood forecasting, we consider three aspects: model structure, prediction results, and computational cost. The results are further divided into three perspectives: basic model results, algorithm optimization model results, and attention mechanism coupling model results, to obtain an objective and comprehensive comparison.

Therefore, in this section, we introduce the basic RNN unit, the LSTM unit, the way information flows in the model, the way attention mechanism is coupled, and the logical structure of algorithm optimization to clarify the characteristics of information flow in each model and analyze the relevant results based on this.

3.1. Basic Model

3.1.1. RNN Unit

As the earliest recurrent neural network model, the RNN unit has a clear structure, and its internal structure is shown in Figure 6:

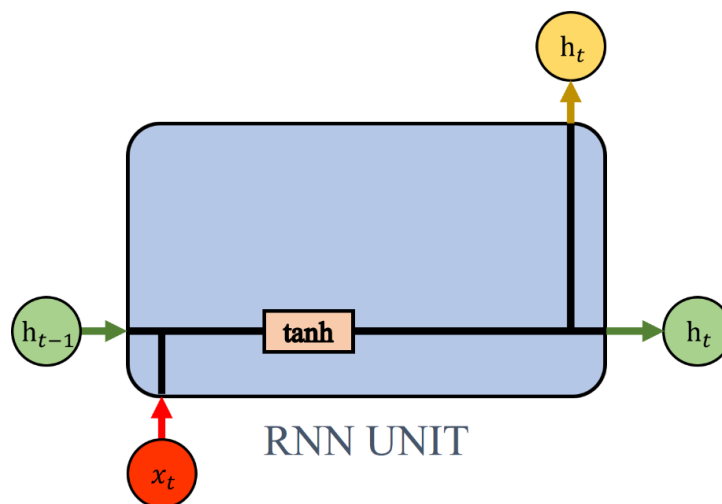


Figure 6. RNN unit structure. In this figure, h stands for hidden state; x stands for input information; \tanh is tangent activation; t and $t - 1$ stand for time.

In the unit shown in Figure 6, the information propagation mode is as follows:

$$h_t = \tanh(W_h[h_{t-1}, x_t] + b_h), \tag{1}$$

where W_h and b_h are the corresponding weights and biases. The hidden state h_t output at the time t is jointly determined by the input information x_t and the hidden state h_{t-1} at time $t - 1$.

3.1.2. LSTM Unit

The LSTM model introduces three gate units inside the RNN unit to control the information flow and uses a cell state to store the historical information, thereby solving the gradient vanishing or exploding problems. The internal structure of the unit is shown in Figure 7.

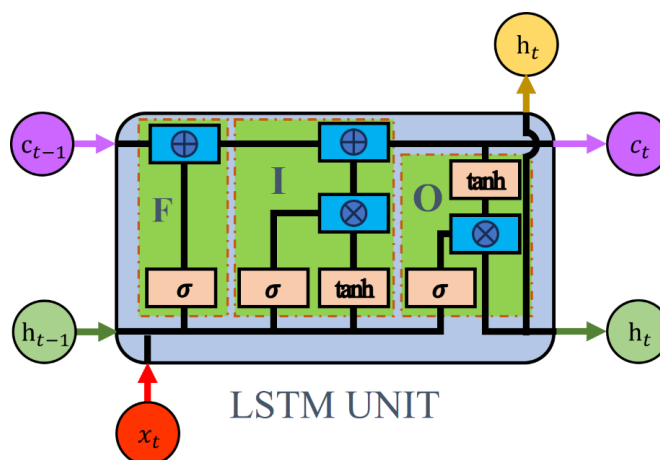


Figure 7. LSTM unit structure. In this figure, c represents cell state; “F, I and O”, respectively, represent the forget gate, the input gate, and the output gate; \oplus represents pointwise addition; \otimes represents pointwise multiplication; σ represents sigmoid activation.

The internal structure of LSTM is very complex, and the specific calculation process can be found in article [32]. The mathematical expression of its information transmission can be found in Equations (2)–(7):

$$\Gamma_f^{<t>} = \sigma(W_f[h_{t-1}, x_t] + b_f), \tag{2}$$

$$\Gamma_i^{<t>} = \sigma(W_i[h_{t-1}, x_t] + b_i), \tag{3}$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \tag{4}$$

$$c_t = \Gamma_f^{<t>} * c_{t-1} + \Gamma_i^{<t>} * \tilde{c}_t, \tag{5}$$

$$\Gamma_o^{<t>} = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{6}$$

$$h_t = \Gamma_o^{<t>} * \tanh(c_t), \tag{7}$$

where W and b are the variable weights and biases; Γ_f , Γ_i and Γ_o are the forget, input, and output gates that vary over time; \tilde{c}_t is the candidate cell state computed from the new input x_t and the previous hidden state h_{t-1} ; c_t is the updated cell state by combining \tilde{c}_t and the previous cell state c_{t-1} ; h_t is the hidden state at time t generated by the output gate.

3.1.3. Model Transmission Structure

Based on the RNN unit or LSTM unit, the overall model information flow architecture is shown in Figure 8.

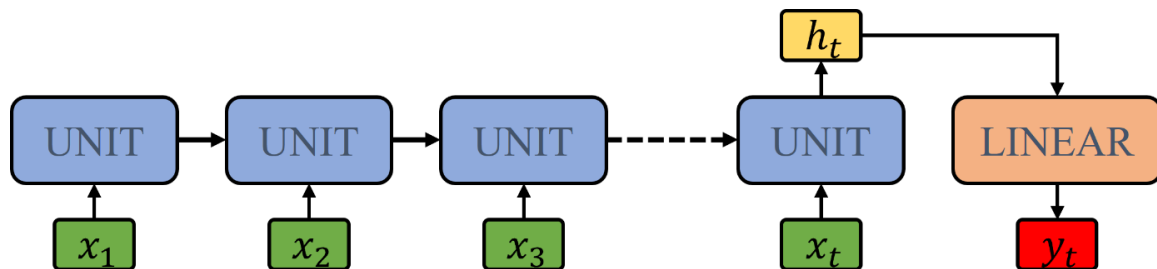


Figure 8. Model Information Flow. In this figure, the unit can be an RNN unit or an LSTM unit. Linear is a linear layer, y_t is output at time t .

3.2. Attention Mechanism Coupling Model

Neural network models have a black-box structure that limits their interpretability, which has been a major concern for many researchers, especially regarding LSTM models [33]. To shed some light on the way the model works, researchers applied the attention mechanism (AM) to it. The AM was first proposed by Bahdanau, et al. [34] and others to address the long sequence gradient propagation issue of recurrent neural network models in NLP tasks. AM allows researchers examination of the amount of attention that the model pays to different parts of input information, and thus obtention of some insight into its working mechanism. Therefore, the AM is also adopted by hydrologists to enhance the interpretability of their models [35]. Building on the AM, Google team [10] introduced a multi-head attention mechanism (MHAM) and developed the transformer architecture, which had a huge impact on various fields. To compare the performance of interpretable models, this paper integrates RNN and LSTM models with MHAM and develops MHAM-RNN and MHAM-LSTM for performance evaluation. The logic structure of the model is illustrated in Figure 9.

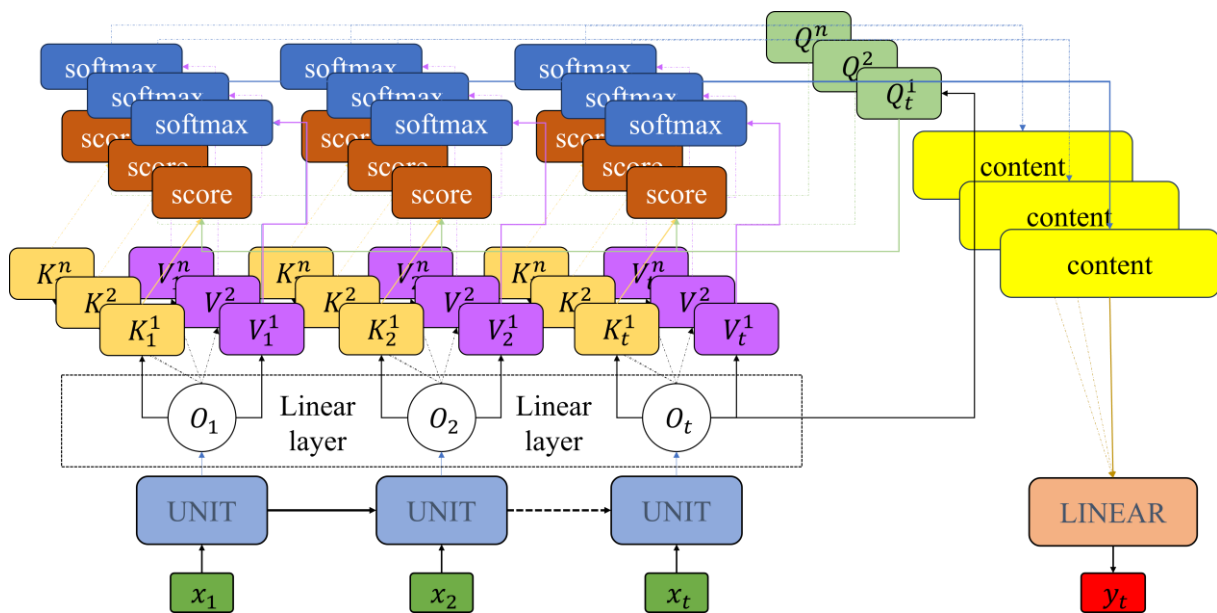


Figure 9. MHAM coupled logic structure. In this figure, the unit can be an RNN unit or an LSTM unit.

For the input information, the model processes the information flow as follows:

(1) Taking x_1 as an example, after passing through the RNN or the LSTM unit, it enters the linear unit O_1 and outputs n copies of answer sequences $\{K_1^1 K_1^2 \dots K_1^n\}$ and value sequences $\{V_1^1 V_1^2 \dots V_1^n\}$. At the same time, the question sequences $\{Q_1^1 Q_1^2 \dots Q_1^n\}$ are obtained from the last moment, representing the answers, values, and questions on different heads, respectively;

(2) Taking the n th head as an example, Q_t^n and K_1^n perform a vector-scaled dot product, and obtain the attention situation of x_1 on the n th head:

$$n(\alpha_t^1) = \frac{(K_1^n) * (Q_t^n)^T}{\sqrt{d^k}}, \tag{8}$$

where d^k is the dimension of the sequence K , and α_t^i is the degree of attention of Q_t to K_i at time t ($i = 1 \dots t$);

(3) The attention scores of n heads $n(\alpha_t^1)$ are normalized and activated by the softmax function, and the sequence $n(\hat{\alpha}_t^1)$ is obtained:

$$\hat{\alpha}_t^1 = \frac{\exp(\alpha_t^1)}{\sum_{j=1}^n \exp(\alpha_t^j)}; \tag{9}$$

(4) The activated attention sequence $h(\hat{\alpha}_t^1)$ and V_1^n perform a vector dot product, and obtain the calculation content of the n th head about x_1 :

$$content(h_1) = h(\hat{\alpha}_t^1) * V_1^n; \tag{10}$$

(5) After concatenating the calculation contents, the prediction result is obtained through the linear layer.

3.3. Model Hyperparameter Optimization

Appropriate hyperparameters can enhance the model performance and prediction accuracy, so they can be selected by using algorithm optimization methods [36]. To compare RNN and LSTM models more objectively, this paper adopts the BOA to choose important hyperparameters for each model and aims to achieve the best fit [37]. BOA is stable and

efficient, and widely applied in hyperparameter optimization tasks [38,39]. The details of BOA are not repeated here but can be referred to in [37]. The algorithm is integrated with each model to obtain BOA-RNN and BOA-LSTM models, whose logical structures are shown in Figure 10.

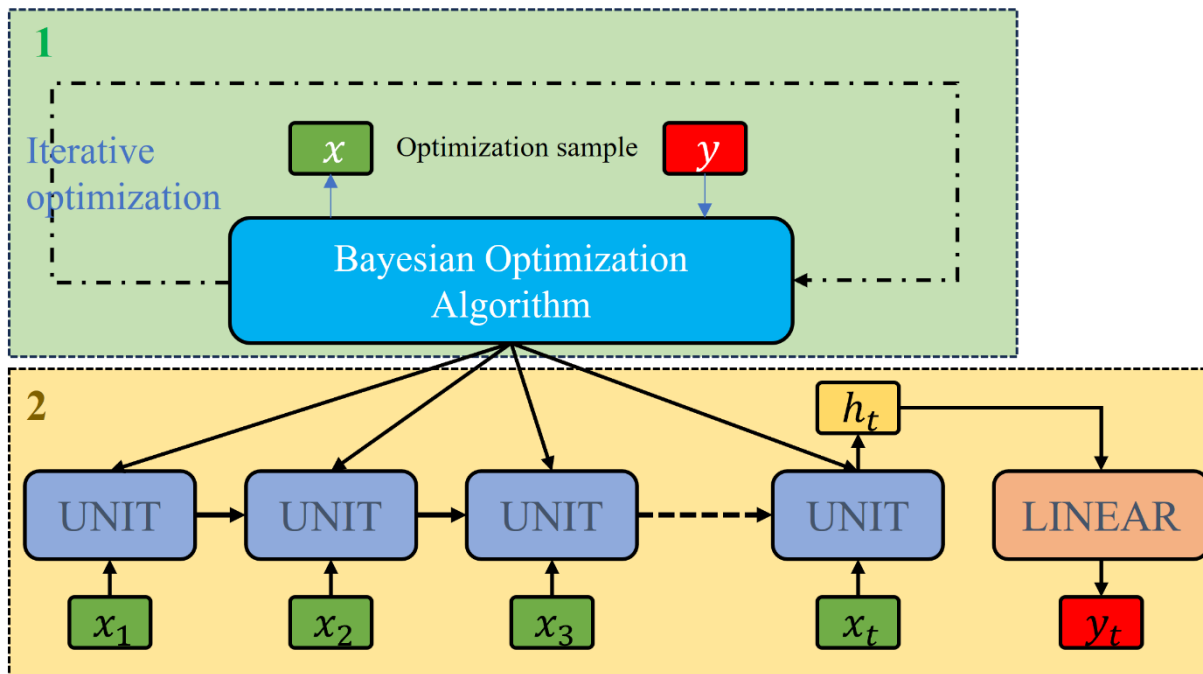


Figure 10. Logical structure of the BOA coupling model. In this figure, the unit can be an RNN unit or an LSTM unit (1 represents the algorithm optimization process, and 2 represents the process where the model makes predictions based on the hyperparameters found by the algorithm.).

Based on relevant research, this paper tries to optimize three hyperparameters: learning rate, neuron number, and regularization parameter. Table 1 shows their optimization ranges and selection reasons [40–42]. The learning rate affects the learning speed of the model. A suitable learning rate can help the model avoid saddle points and find optimal solutions; The neuron number determines the nonlinear representation ability of the hidden layer. A proper number can facilitate the model to extract a consistent representation from data. The regularization parameter prevents overfitting by adding a penalty term to the model and balances its performance in training and testing.

Table 1. Algorithm optimization target details.

Optimization Objectives	Optimization Scope	Reason for Selection
Learning rate	$(1 \times 10^{-4}, 1 \times 10^{-2})$	Control model gradient descent
Hidden units	(10, 200)	Control model’s nonlinear expression ability
L2 Regularization	$(1 \times 10^{-7}, 1 \times 10^{-3})$	Avoid overfitting

3.4. Analysis of Model Differences

The paper constructs three sets of models: RNN and LSTM, MHAM-RNN and MHAM-LSTM, BOA-RNN and BOA-LSTM. Comparing the three sets of models horizontally, the differences lie in the way information is transmitted between RNN and LSTM, which is used to analyze the impact of the way information is transmitted on the results when RNN and LSTM models are used as the basis. Comparing the three sets of models vertically, there are only hyperparameter differences between RNN and BOA-RNN models as well as

LSTM and BOA-LSTM models, which are used to analyze the impact of hyperparameters on RNN and LSTM models; there is a change in the way information is transmitted between RNN and MHAM-RNN models as well as LSTM and MHAM-LSTM models, which is used to further analyze the impact of the differences in information transmission brought about by the coupling of multi-head attention mechanisms on the model.

3.5. Related Parameter Settings

To better utilize model performance, this paper sets important hyperparameters such as learning rate, neuron number, and regularization parameter. For the sake of comparison and analysis, except for the target hyperparameters optimized by the algorithm optimization models, the other hyperparameters of each model are set to the same values, as shown in Table 2.

Table 2. Model-related settings.

Name	Setting	Reason
Learning rate	1×10^{-3}	Beneficial for stable gradient descent
Hidden units	128	Sufficient nonlinear expression ability
L2 Regularization	1×10^{-5}	Avoidable overfitting
Gradient descent algorithm	Adam	Stable effect
iterations	1500	Meet iteration requirements

3.6. Model Evaluation Indicators

The paper selects suitable indicators to evaluate the model performance based on the previous research results [43]. The Nash–Sutcliffe efficiency coefficient (*NSE*), the root mean square error (*RMSE*), the mean absolute error (*MAE*), and the Kling–Gupta efficiency coefficient (*KGE*) are adopted, and their calculation Equations are as follows:

$$NSE = 1 - \frac{\sum_{i=1}^N (Q_i - P_i)^2}{\sum_{i=1}^N (Q_i - Q_{avg})^2}, \quad (11)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Q_i - P_i)^2}{N}}, \quad (12)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |Q_i - P_i|, \quad (13)$$

$$KGE = 1 - \sqrt{(\alpha - 1)^2 + (\beta - 1)^2 + (R - 1)^2}, \quad (14)$$

where N is the number of data points, Q_i is the observed runoff at time i , P_i is the predicted runoff at time i , Q_{avg} is the mean of the observed runoff, $\alpha = \sigma_p / \sigma_o$ is the variability bias, $\beta = \mu_p / \mu_o$ is the mean bias, σ and μ represent the standard deviation and mean, respectively, and R is the linear correlation coefficient.

NSE is sensitive to the fluctuations of the data series and can characterize the tracking ability of the predicted values to the actual values. It is used to evaluate the stability of the model prediction. *RMSE* and *MAE* are used to calculate the error of the predicted values, indicating the overall prediction accuracy of the model. *KGE* combines the model correlation, bias, and flow variability into one objective for unified evaluation.

4. Results and Discussion

4.1. Model Structure Comparison Analysis

The difference in model performance ultimately stems from their different underlying architectures, and the architecture that is more in line with the task requirements will inevitably be able to obtain better forecasting results. Therefore, analyzing the adaptability

of the two models to the flood forecasting task requirements from the structural perspective can better explain the difference in performance of the two models.

The principle of deep learning models for flood forecasting tasks is simply to fit the true mapping relationship between effective data and prediction targets. The structural difference between the two models is that the LSTM model uses gated units to construct information transfer of cell states at longer time steps, avoiding gradient vanishing problems, as shown in Equations (4) and (5). Is mitigating gradient vanishing problems effective for approximating the true flood mapping relationship? We show the specific gradient propagation processes of the two models, as shown in Figure 11.

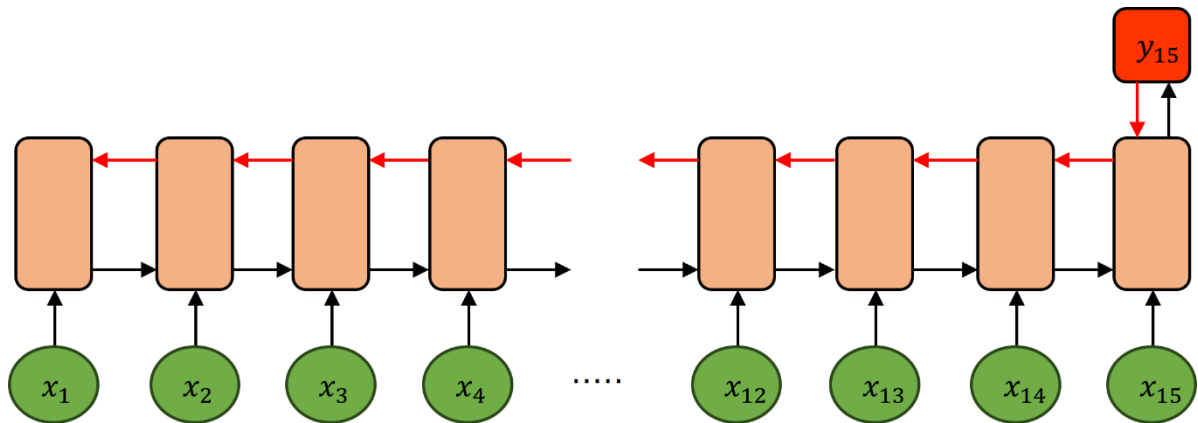


Figure 11. Gradient propagation process of each batch of data in flood forecasting tasks. The black arrow represents the forward-propagation process; the red arrow represents the backpropagation process.

There are two types of gradient problems in recurrent neural networks: spatial and temporal. Spatial gradient problems refer to the gradient vanishing or exploding when flowing between different hidden layers in the neural network. Since the data relationship of the flood forecasting task is not complex (compared to the TB-level tasks in NLP), the model can achieve good generalization without increasing the number of hidden layers; thus, there is no spatial gradient problem. Temporal gradient problems refer to the gradient vanishing or exploding when flowing through different time steps within each batch, as shown in Figure 11.

For the temporal dimension, the flood forecasting task exhibits the gradient propagation phenomenon shown in Figure 11 in each batch, but compared to the NLP task, the gradient accumulation in the flood forecasting task has its uniqueness.

(1) The batch time span of the flood forecasting task is short, and there is no phenomenon of long-term gradient accumulation.

Different from the NLP task, limited by the basin scale, the model input sequence time step in the flood forecasting task is relatively short. As shown in Figure 11, the longest gradient propagation process spans only from x_1 to y_{15} , and the gradient multiplication occurs no more than 15 times. Compared with the gradient propagation of thousands of time steps in the NLP task, the flood forecasting time scale is difficult to lead serious gradient propagation problems.

(2) The gradient propagation is more consistent with the physical level of flood phenomena.

As the NLP task depends on the linguistic context, it often requires establishing connection relationships across any time span, which can be mathematically formulated as Equation (15).

$$x_i = \text{linear}(y_j) \quad i, j = 1, 2, 3 \dots \text{size}(\text{batch}). \quad (15)$$

The cell state transmitted by the gated unit of the LSTM model is also used for this purpose.

The outflow state at Time step 15 in the flood forecasting task is solely determined by Time step 14, and the hydrological-related factors at Time steps 1–13 cannot affect Time step 15 across time scales. Flood propagation is strictly stepwise and monotonic, and gradient propagation better reflects this physical phenomenon, while the cross-time-scale connection established by the gated unit of the LSTM model is inconsistent with the physical level of the flood process.

(3) Long-distance information is not important for the flood forecasting task. Since the flood information propagation process is stepwise unidirectional, the flood information weight should decrease with time. To illustrate this situation, we apply the attention mechanism model to measure the attention level to the input information in the flood process using the DaWenkou station as an example, as illustrated in Figure 12.

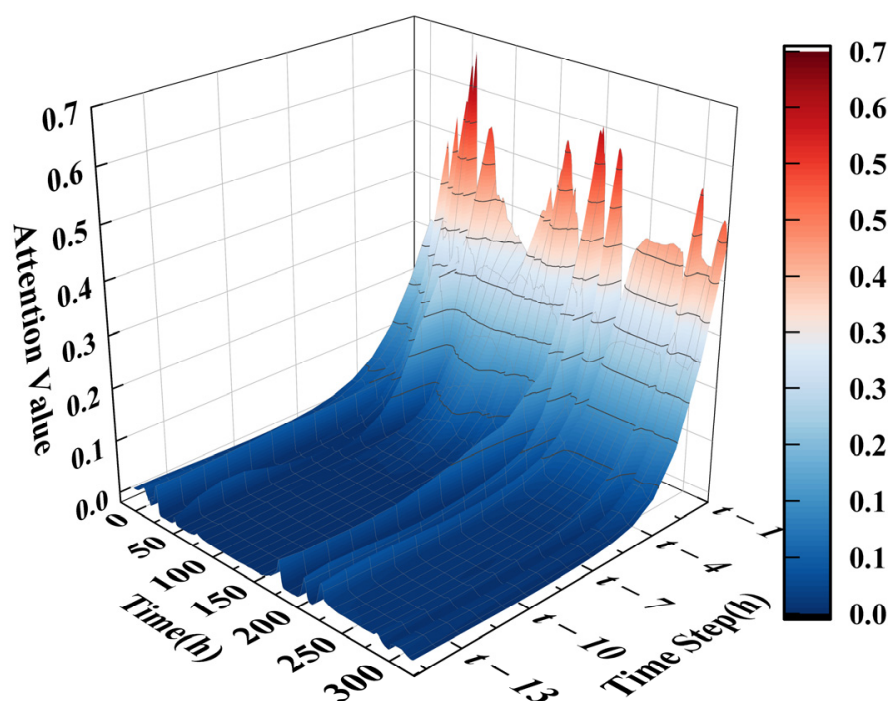


Figure 12. Attention distribution of the model for test period data at the LouDe station.

As can be seen from Figure 12, the model pays very little attention to the information with a longer time span ($t - 15$), which indicates that it has a small impact on the prediction result, while the attention level to the recent information ($t - 1$) remains high. Therefore, it can be seen that since the values at longer time steps have a small impact on the prediction result, even if the gradient vanishing phenomenon occurs, it does not affect the prediction result significantly, because the information itself does not require high weights.

In summary, the characteristics of the flood forecasting task itself determine that it does not need a cross-time-scale connection, and the LSTM model's gated unit's mitigation of gradient problems is also unlikely to improve flood forecasting, but instead reduces the interpretability due to the increased model complexity.

4.2. Basic Model Comparison Analysis

Comparing the prediction results of the two models is the most direct way to validate the structural analysis. In total, 10 experiments were performed in this paper for each of the two models at two stations, and the prediction result metrics were plotted into charts for comparison.

The average performance of the RNN model and the LSTM model experiment results are shown in Table 3 and Figure 13.

Table 3. Average performance of model experiment results.

Station	Model	NSE	KGE	MAE	RMSE
LouDe	RNN	0.9789	0.9591	13.1555	24.9860
	LSTM	0.9621	0.9184	20.4024	33.4670
HuaYuankou	RNN	0.9994	0.9988	14.2209	29.6959
	LSTM	0.9992	0.9984	17.3051	35.2501

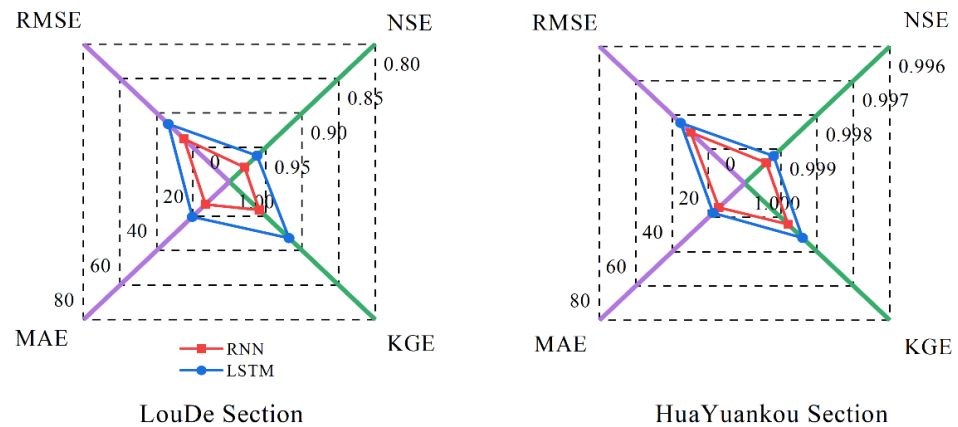


Figure 13. Average performance of model experiment results.

Combining Table 3 and Figure 13, we can see that at the LouDe station, compared with the LSTM model, the RNN model improved by 1.72%, 4.43%, 35.52%, and 25.34% in the four metrics of NSE, KGE, MAE and RMSE, respectively; at the HuaYuankou station, due to the longer time span of the flood process, the NSE and KGE metrics were not very sensitive to the performance difference between the two models, but the RNN model still outperformed the LSTM model, and the performance of the RNN model in the MAE and RMSE metrics improved by 18.09% and 17.22%, respectively. In general, the average performance of each metric of the RNN model at different stations during the test period was better than that of the LSTM model.

The average metrics can only show the average situation of the model prediction effect, and they cannot capture the variation of the prediction result metrics. If the model prediction effect fluctuates greatly, it is also difficult to apply it to practical work. Therefore, we plotted the results of 10 random experiments of the model into a graph to observe the variation of its prediction effect, as shown in Figure 14.

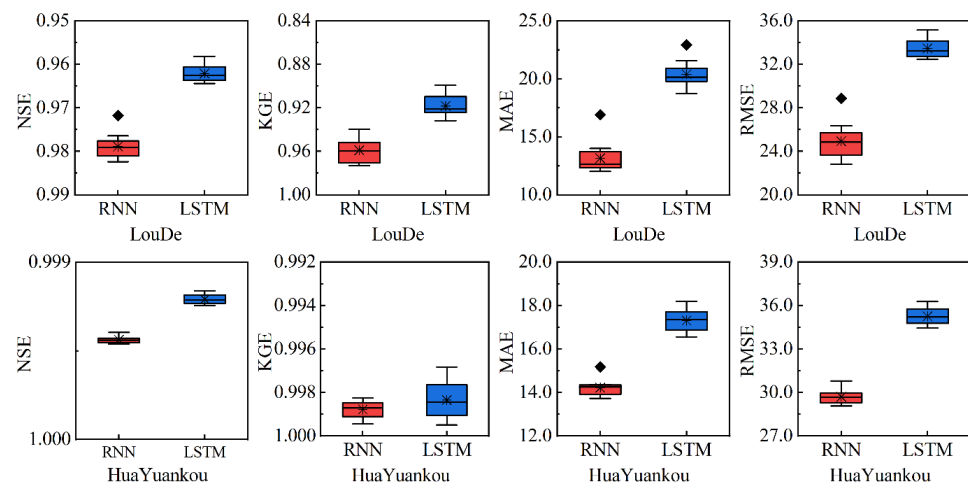


Figure 14. Distribution of indicators during the testing period of RNN and LSTM models.

Based on Figure 14, we can see that for the flood forecasting task, the two models have similar fluctuations in the prediction results, but the RNN model performs better than the LSTM model in all cases.

In addition to the distribution of the prediction performance, the extensibility of the lead time when performing the flood forecasting task also needs to be considered. To compare the RNN and LSTM models more comprehensively, we gradually increased the lead time to 3 h, and the average performance metrics of the models are shown in Table 4. The performance improvement of the RNN model is shown in Table 5. The changes in the prediction performance of the two models are shown in Figure 15.

Table 4. Average performance indicators of RNN and LSTM models under different lead times.

Lead Time	Station	Model	NSE	KGE	MAE	RMSE
2 h	LouDe	RNN	0.9305	0.9305	23.8539	45.3195
		LSTM	0.8988	0.8887	28.6722	54.6817
	HuaYuankou	RNN	0.9985	0.9988	24.0427	47.8815
		LSTM	0.9981	0.9976	27.3378	54.9296
3 h	LouDe	RNN	0.9009	0.9225	29.7196	54.1151
		LSTM	0.8783	0.8907	31.6699	59.9619
	HuaYuankou	RNN	0.9971	0.9979	35.0459	67.4454
		LSTM	0.9966	0.9957	36.9041	73.3273

Table 5. Performance improvement effect of the RNN model compared to the LSTM model.

Lead Time	Station	NSE	KGE	MAE	RMSE
2 h	LouDe	3.53%	4.70%	16.80%	17.12%
	HuaYuankou	0.04%	0.12%	12.05%	12.83%
3 h	LouDe	2.57%	3.57%	6.16%	9.75%
	HuaYuankou	0.05%	0.22%	5.04%	8.02%

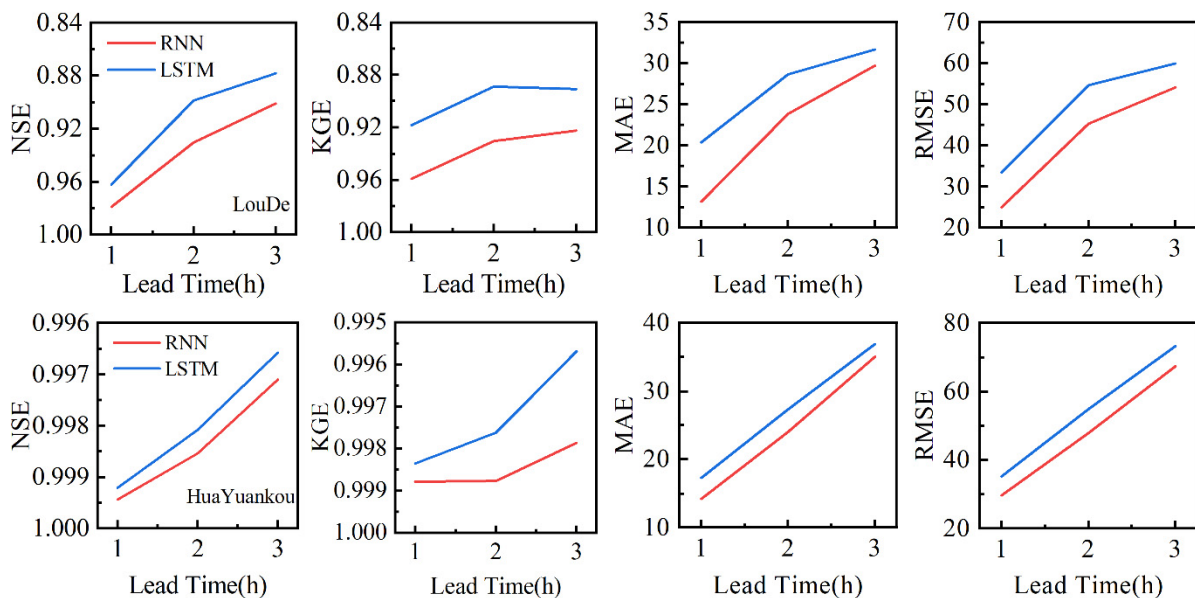


Figure 15. The variation of average performance of RNN and LSTM models with lead time.

From Tables 4 and 5 and Figure 15, we can observe that as the lead time increases, the average performance of the two models declines in all metrics, but the RNN model still outperforms the LSTM model significantly.

To intuitively show the model prediction effect at different lead times, we visualized the model prediction of the flood process at two stations in graphs, as illustrated in Figures 16–19.

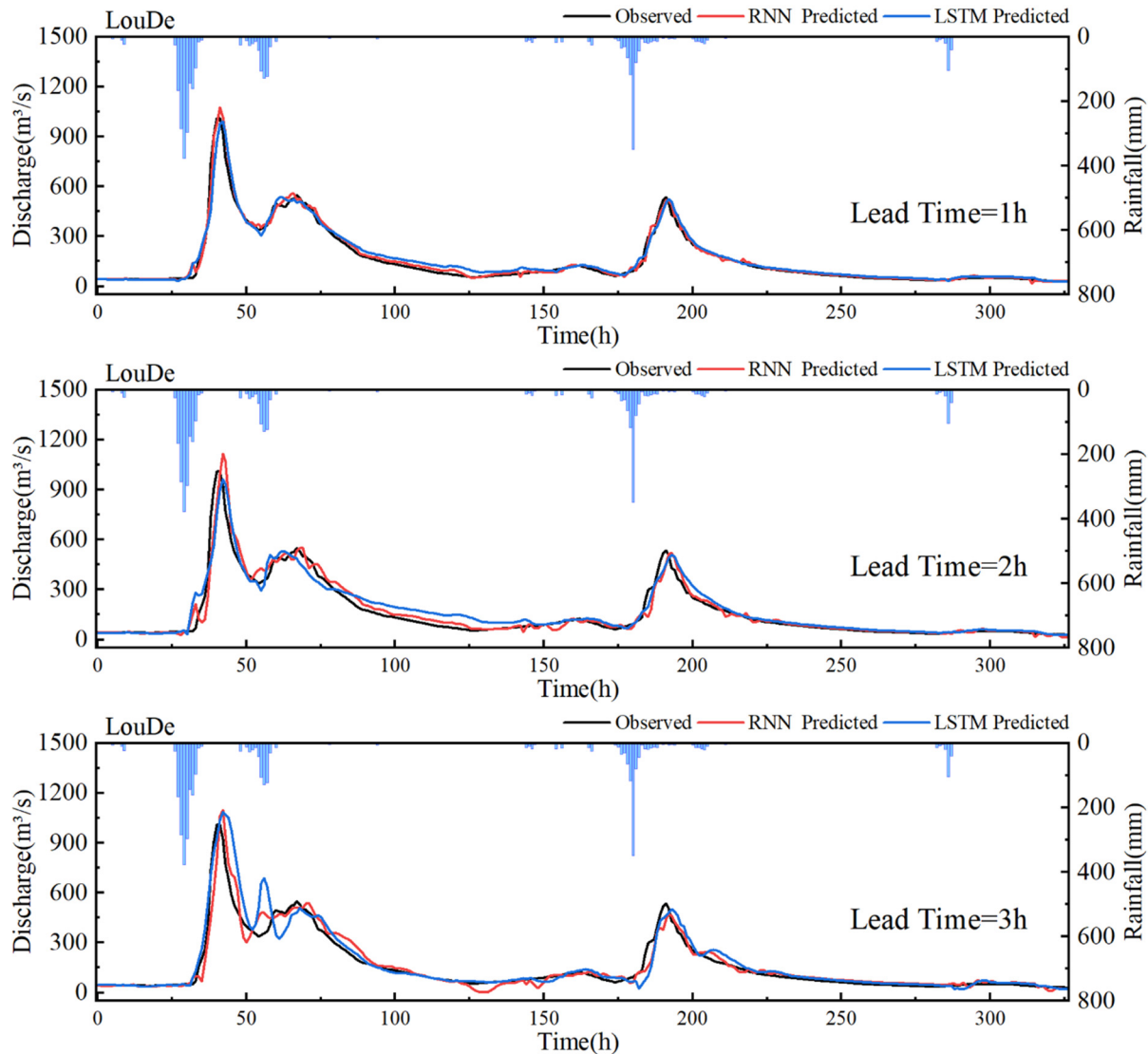


Figure 16. Model predictive effects under different lead times of the LouDe station.

As can be seen from Figure 16, there is a significant difference between the two models in the prediction effect of the flooding process at the LouDe station during the test period. Compared with the LSTM model, the RNN model is closer to the measured value except for being slightly larger at the peak, especially in the flood recession process. The LSTM model has a relatively poor prediction effect, and the prediction process at 3 h lead time shows obvious deviation.

Figure 17 shows the correlation of the predicted results for the two models at the LouDe station under different forecast periods. The RNN model demonstrates better performance in various scenarios.

Figures 18 and 19 show that the flood prediction performance of the two models at the HuaYuankou station is comparable. However, the RNN model has a higher correlation coefficient of the predicted results and a better prediction accuracy than the LSTM model.

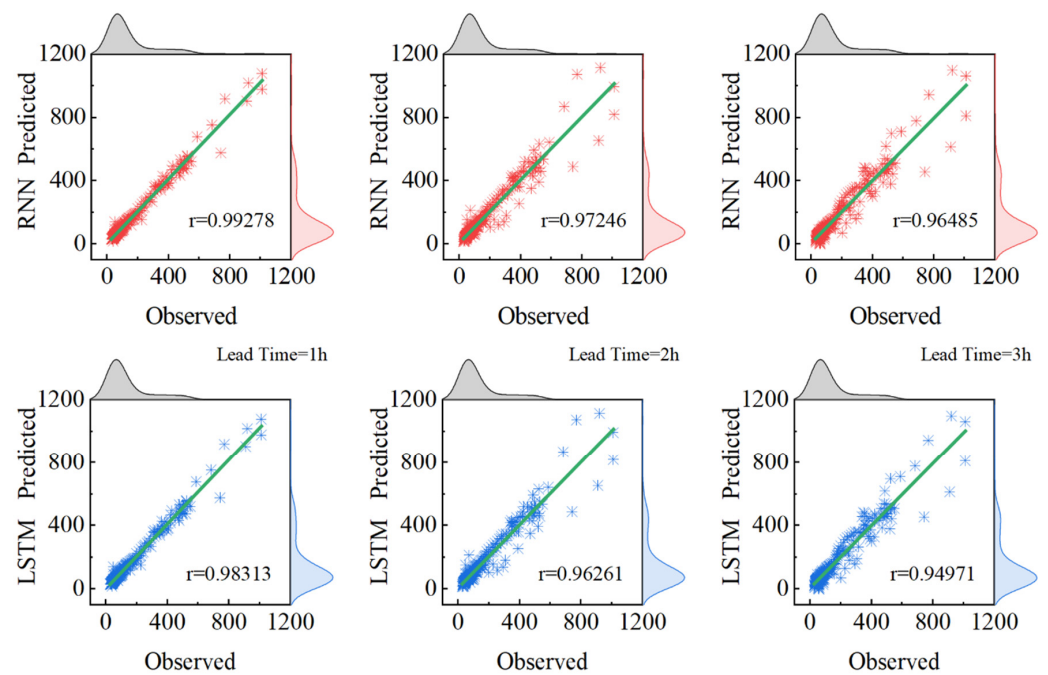


Figure 17. Scatter plot of the model prediction performance at the LouDe station under different lead times (“*” represents the distribution of data in observed and predicted values).

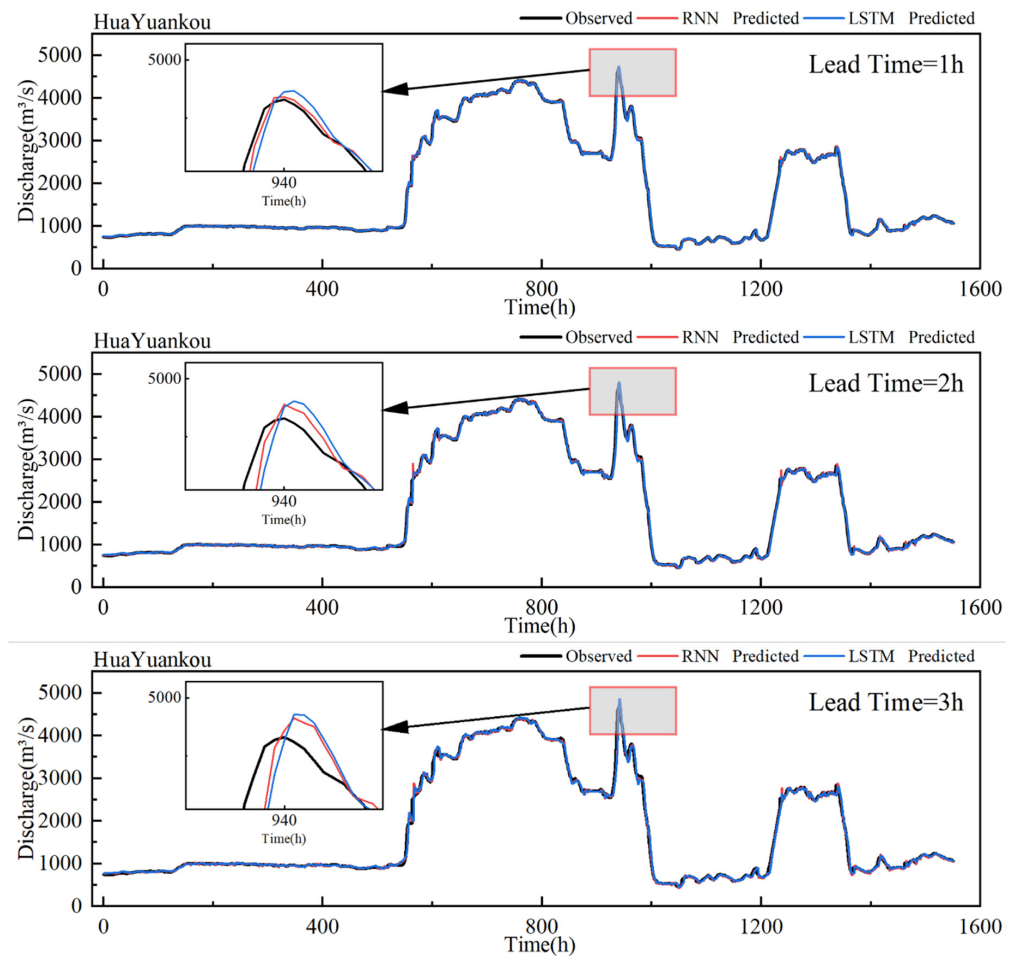


Figure 18. Model predictive effects under different lead times of the HuaYuankou station.

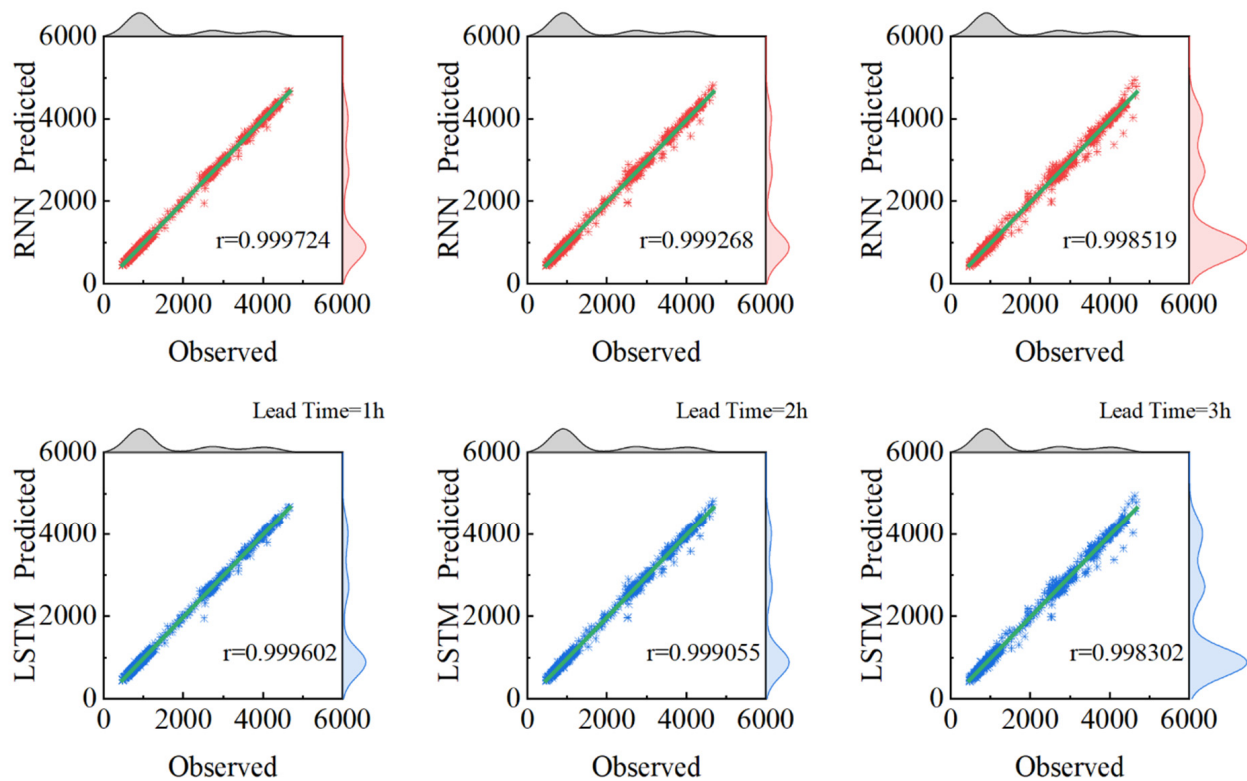


Figure 19. Scatter plot of the model prediction performance at the HuaYuankou station under different lead times (“*” represents the distribution of data in observed and predicted values).

The comparison of Tables 3–5 and Figures 13–19 reveals that the RNN model outperforms the LSTM model in flood forecasting tasks, which is different from their performance in NLP tasks. This finding is consistent with the analysis of the two models’ architectures in Section 4.1, which indicates that the RNN unit structure is more adaptive to the information propagation in flood forecasting tasks.

4.3. Basic Model Hyperparameter Optimization Comparison Analysis

Selecting model hyperparameters with optimization algorithms is a common optimization technique. Algorithmically optimized hyperparameters can usually exploit model performance and improve model abstraction ability better than manually selected ones; on the other hand, optimizing hyperparameters brings the model performance closer to its limit and enables a better evaluation of model performance. Therefore, we use the Bayesian optimization algorithm to couple the RNN and LSTM models, determine their relevant hyperparameters (see Table 1 for details), and construct the BOA-RNN and BOA-LSTM models to evaluate their performance in flood forecasting. Table 6 shows the selected hyperparameters of both models, and Table 7 and Figure 20 show their average performance indicators.

Table 6. Algorithm-selected model hyperparameters.

Model	Station	Learning Rate	Hidden Units	L2 Regularization
RNN	LouDe	$9.94721325044244 \times 10^{-3}$	102	$1.24128822320419 \times 10^{-6}$
	HuaYuankou	$6.31430706313691 \times 10^{-4}$	169	$2.29751978936061 \times 10^{-5}$
LSTM	LouDe	$8.60162246721079 \times 10^{-3}$	141	$1.00331686970300 \times 10^{-6}$
	HuaYuankou	$1.16040150867700 \times 10^{-3}$	130	$5.49314205875754 \times 10^{-6}$

Table 7. Average Performance of Algorithm Optimization Model Experimental Results.

Station	Model	NSE	KGE	MAE	RMSE
LouDe	BOA-RNN	0.9819	0.9679	11.4860	23.1318
	BOA-LSTM	0.9752	0.9607	13.4444	27.0929
HuaYuankou	BOA-RNN	0.9994	0.9988	14.5352	30.1463
	BOA-LSTM	0.9993	0.9985	15.6075	32.1723

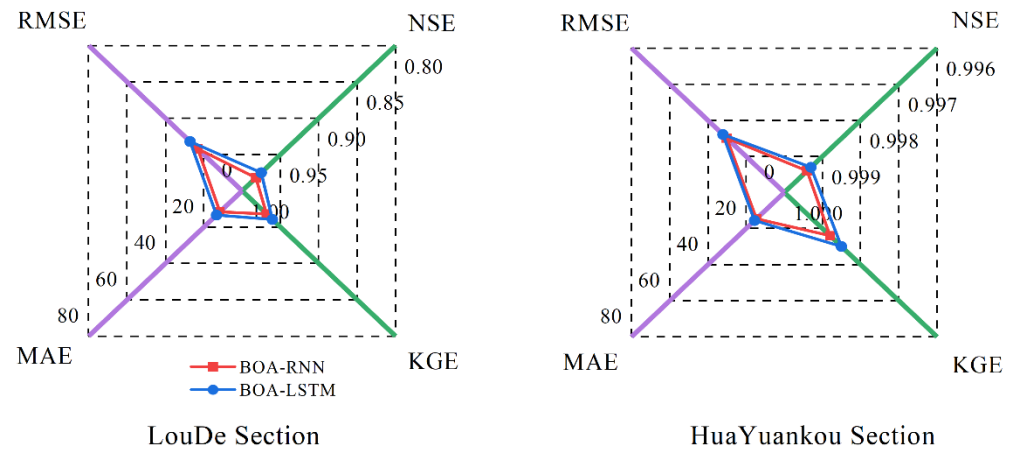


Figure 20. Average performance of algorithm optimization model experimental results.

Table 7 and Figure 20 show that at the LouDe station, the BOA-RNN model improved by 0.69%, 0.75%, 14.57% and 14.62% in NSE, KGE, MAE and RMSE, respectively, compared to the BOA-LSTM model; at the HuaYuankou station, both models had similar NSE and KGE indicators, but the BOA-RNN model performed better in MAE and RMSE indicators by 6.87% and 6.30%, respectively. The difference between both models resembles that of the basic models, and the BOA-RNN model continues to outperform the BOA-LSTM model.

However, Tables 3 and 7 reveal that after using the Bayesian optimization algorithm, the LSTM model improved its performance significantly; but the RNN model had a small improvement, and even performed slightly worse than the BOA-RNN model at the HuaYuankou station. Table 8 shows the change in the average values of both groups of models.

Table 8. Changes in model performance after algorithm optimization.

Station	Model	NSE	KGE	MAE	RMSE
LouDe	BOA-RNN/RNN	0.31%	0.92%	12.69%	7.42%
	BOA-LSTM/LSTM	1.36%	4.61%	34.10%	19.05%
HuaYuankou	BOA-RNN/RNN	0.00%	0.00%	−2.21%	−1.52%
	BOA-LSTM/LSTM	0.01%	0.01%	9.81%	8.73%

We attribute this phenomenon to the RNN model structure’s better adaptation to the flood forecasting task, which enables it to learn the data mapping relationship accurately. Therefore, the manually selected hyperparameters work well, and the hyperparameter optimization has little impact on the model learning effect. At the HuaYuankou station, however, the rich data make the data mapping relationship fitted by the RNN model closer to the physical level relationship, which makes it hard to improve the model performance indicators with hyperparameter optimization.

We also show the indicator distribution of the 10 experiments of both models in Figure 21.

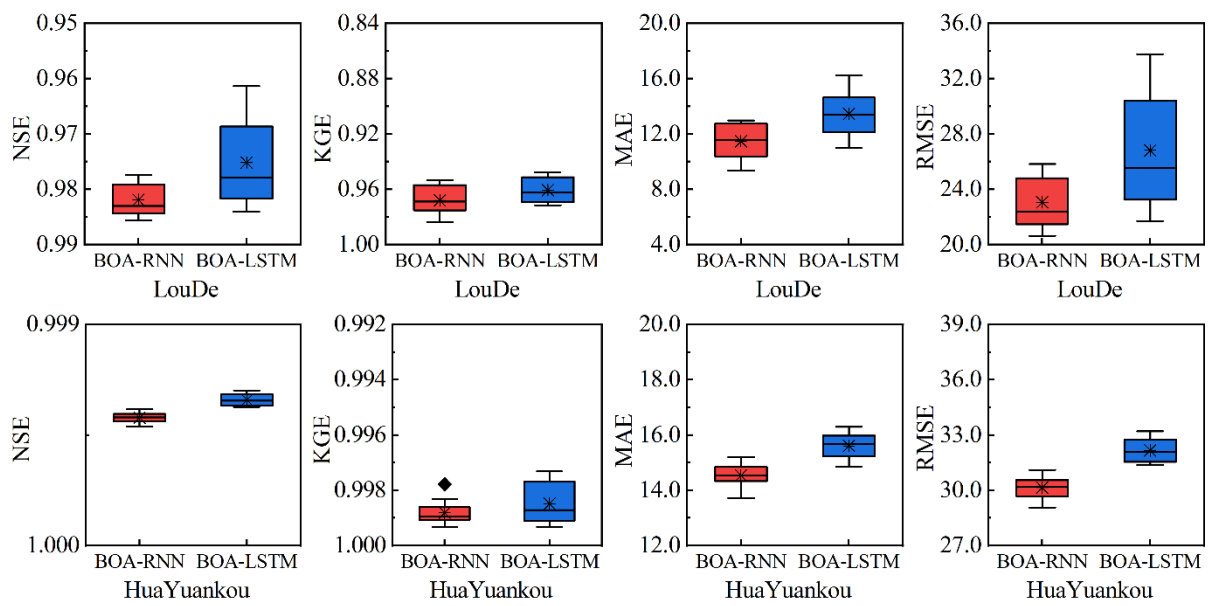


Figure 21. Distribution of indicators during the testing period of BOA-RNN and BOA-LSTM models.

The BOA-RNN model’s prediction performance distribution is still better than that of the BOA-LSTM model; in addition, the BOA-LSTM model’s prediction results fluctuate significantly.

The prediction effects of the two models at different stations during the test period are shown in Figures 22 and 23.

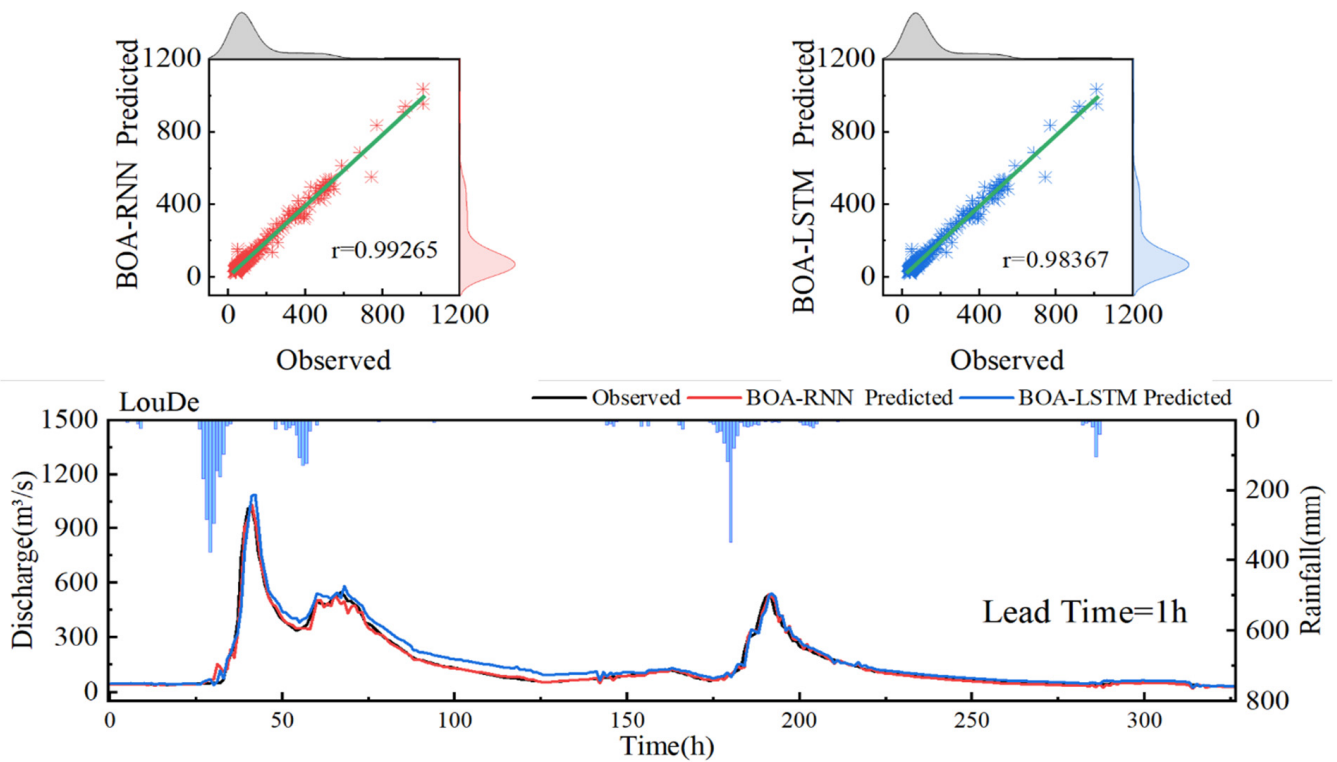


Figure 22. Algorithm optimization model prediction effect of the Loude station during the testing period.

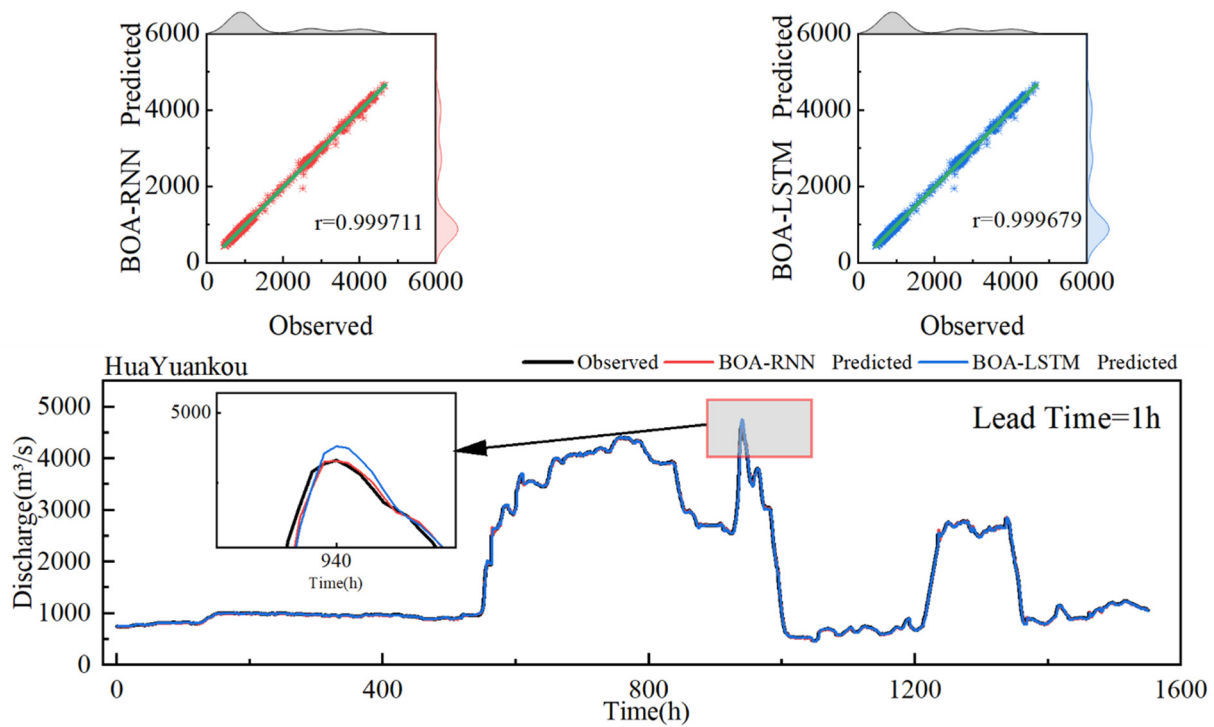


Figure 23. Algorithm optimization model prediction effect of the HuaYuankou station during the testing period.

The BOA-RNN model maintains a better prediction effect than the BOA-LSTM model.

Tables 6–8 and Figures 20–23 show that the BOA-LSTM model, which optimizes the hyperparameters with the algorithm, significantly improves the LSTM model’s performance, but the BOA-RNN model performs similarly to the RNN model. This suggests that the RNN model fits flood forecasting better, and makes relatively accurate predictions without hyperparameter optimization, saving computational and human resources in research. Moreover, the RNN model as the optimization object predicts better than the LSTM model, even with hyperparameter optimization.

4.4. Attention Mechanism Coupling Model Comparison Analysis

The LSTM model has a complex gate unit that makes its mechanism hard to explain. Therefore, researchers use the attention mechanism to measure the amount of attention the model pays to the data and then infer its mechanism. In this study, we couple the RNN and LSTM models with the advanced multi-head attention mechanism and construct the MHAM-RNN and MHAM-LSTM models. We analyze both the coupling effect of the attention mechanism on the model and the difference in attention to the input data between the two models.

Table 9 shows the average values of both model prediction results, and Figure 24 shows their performance distribution.

Table 9. Predicting average performance with a coupled attention mechanism model.

Station	Model	NSE	KGE	MAE	RMSE
LouDe	MHAM-RNN	0.9758	0.9569	14.2923	26.7232
	MHAM-LSTM	0.9556	0.9433	20.4595	36.2084
HuaYuankou	MHAM-RNN	0.9991	0.9979	18.6464	36.9343
	MHAM-LSTM	0.9986	0.9960	24.0945	46.4824

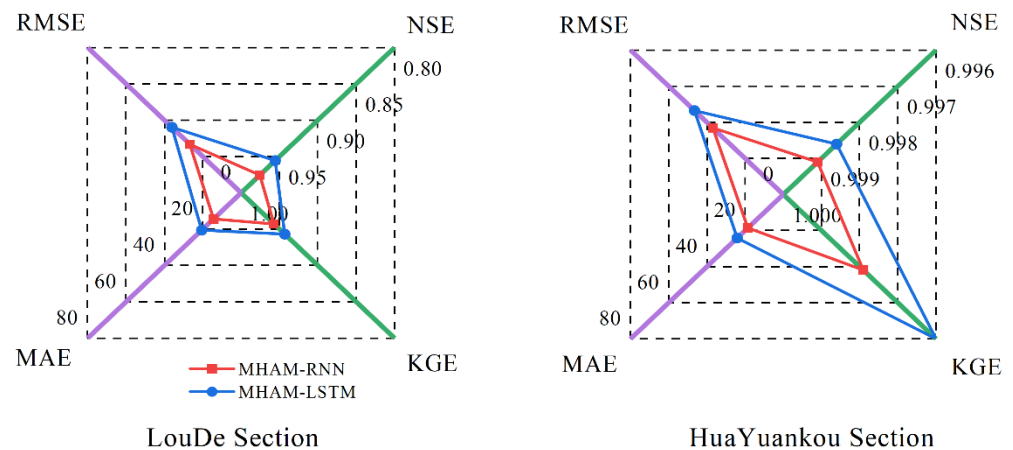


Figure 24. Predicting average performance with a coupled attention mechanism model.

Table 9 and Figure 24 show that the RNN model as the base model outperforms the LSTM model in all aspects after coupling with the multi-head attention mechanism. At the LouDe station, the MHAM-RNN model improves NSE, KGE, MAE, and RMSE by 2.11%, 1.44%, 30.13%, and 26.20%, respectively, compared to the MHAM-LSTM model; at the HuaYuankou station, it also improves these indicators by 0.05%, 0.19%, 22.61%, and 20.54%, respectively. This suggests that the RNN model’s output hidden state is more compatible with the multi-head attention mechanism, and its unit structure preserves more information.

However, Tables 3 and 9 reveal that the models coupled with the multi-head attention mechanism perform worse than the basic models, as shown in Table 10.

Table 10. Changes in average performance indicators relative to the basic model.

Station	Model	NSE	KGE	MAE	RMSE
LouDe	MHAM-RNN/RNN	−0.32%	−0.23%	−8.64%	−6.95%
	MHAM-LSTM/LSTM	−0.68%	2.71%	−0.28%	−8.19%
HuaYuankou	MHAM-RNN/RNN	−0.03%	−0.09%	−31.12%	−24.38%
	MHAM-LSTM/LSTM	−0.06%	−0.24%	−39.23%	−31.86%

At the LouDe station, all indicators except the MHAM-LSTM model’s KGE indicator decline slightly; at the HuaYuankou station, both models show a significant decline in all indicators, with the worst performance degradation in the MHAM-LSTM model.

We attribute this phenomenon to the logical structure of the multi-head attention mechanism. Figure 5 shows that the multi-head attention mechanism establishes the final mapping by relating each time to time t after obtaining the model unit information; that is, it bridges the input data and the final result across time. Its information transmission logic structure can be generalized in Equation (16),

$$y_t^n = linear(x_i^n) \quad i = 1, 2, \dots, t, \tag{16}$$

where t is the final time of each group of data; n is the number of heads of the multi-head attention mechanism; i is a different time; *linear* is the complex relationship between the input data and the result that directly connects them.

As we analyzed in Section 4.1, the flood information flows in a one-way sequence, and the cross-time bridge contradicts the objective law of flood propagation. This feature avoids the gradient problem but hardly impacts the flood prediction result positively. The LSTM basic and derived models predict worse than the RNN model, which also confirms our analysis results. Therefore, adding the attention mechanism to the basic model reinforces the cross-time data bridge in another way, which also lowers the prediction result.

The relatively small performance decline at the LouDe station and the KGE indicator increase in the MHAM-LSTM model as a special case have more reasons for the relatively small data amount at this station, which hinders full reflection of the model performance, and for the multi-head attention mechanism's positive impact on learning complex high-dimensional data (the input data feature value at the LouDe station is 19, or 19-dimensional input data). However, the overall prediction result performance still declines.

To further confirm our analysis on the time span bridge, we graph both models' input data attention degree and analyze their difference in data sensitivity. The multi-head attention mechanism has 16 heads, or 16 attention dimensions, which are hard to understand, so we simplify the attention result to one dimension for easy analysis, shown in Figure 25 (for the LouDe station).

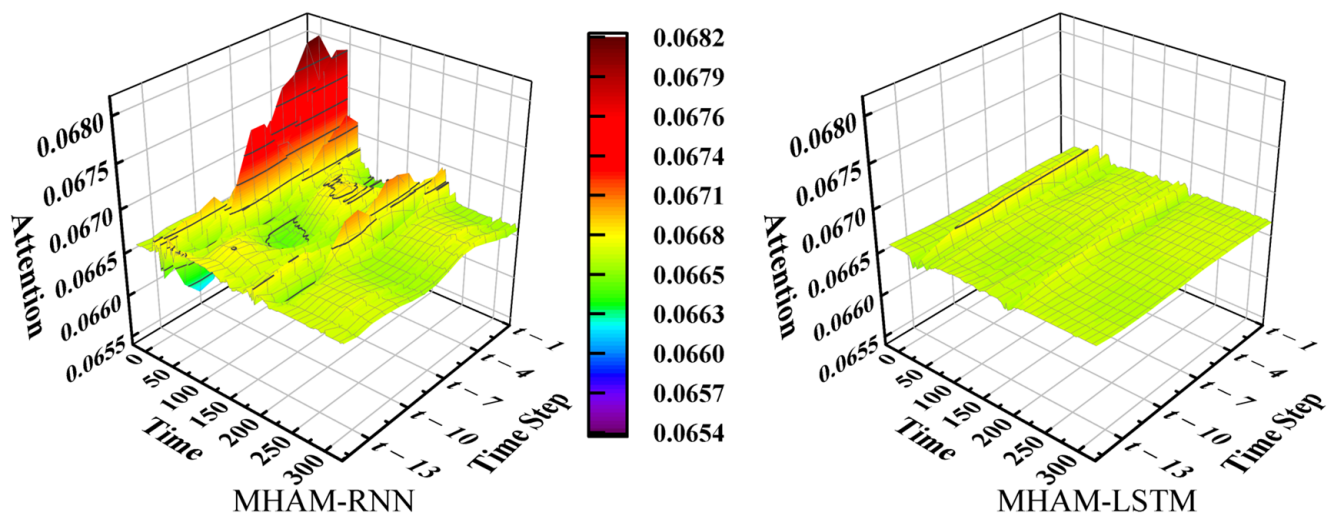


Figure 25. Attention difference of the model for test period data.

Figure 25 intuitively shows the attention differences between the two models. Under the same z-axis, MHAM-RNN with RNN as the underlying model can still distinguish between recent and distant data with different attention weights; however, due to the time bridging ability of the LSTM model, the MHAM-LSTM model is unable to differentiate between different data with the attention mechanism, which leads to the decrease in prediction performance quality. It can be seen that the cross-time bridging of the LSTM model has a negative impact on flood forecasting tasks.

Combining Tables 9 and 10 and Figures 24 and 25, it can be seen that although the introduction of an attention mechanism can improve the interpretability of the model, it has a negative impact on model performance. In addition, neither the attention mechanism nor the cross-time bridging method in the LSTM model have a positive impact on the flood forecasting task.

4.5. Model Parameter and Computational Cost Comparison Analysis

The model development is based on the torch framework of Python 3.8 language, and the GPU and CPU models used for computation are NVIDIA GeForce RTX3080 and Intel Core i7-11800H, respectively. The model computation time cost and parameter number used in this paper are shown in Table 11 (taking the Lode station as an example, the Bayesian algorithm optimization model is excluded due to the high time cost of calculating the optimization process).

As shown in Table 11, the RNN model has the lowest parameter number and computation time cost.

Table 11. Model cost.

Model	RNN	LSTM	BOA-RNN	BOA-LSTM	MHAM-RNN	MHAM-LSTM
Parameters	19,201	76,417	19,201	76,417	68,737	125,953
Time Cost(s)	57.02	63.00	--	---	80.13	88.12

5. Conclusions

To address the research gap in comparing the RNN model and the LSTM model for flood forecasting tasks, this paper conducts a comparative analysis of the models from the perspectives of model structure, algorithm improvement, and attention mechanism coupling based on the measured flood process data of the Lode station and the HuaYuankou station. The main conclusions are as follows:

(1) In flood forecasting tasks, compared with LSTM and its derived models, the RNN model has a simpler structure, a lower computation cost, and a better prediction performance, which makes it more suitable for flood forecasting work;

(2) The RNN model has a stronger interpretability and a better physical mapping of the flood process. Due to the uniqueness of flood data, the LSTM model's construction of the cell state and the attention mechanism and other cross-time bridging methods do not apply to flood sequence forecasting work;

(3) There is no definite relationship between the complexity of the model structure and the quality of prediction results. The model structure should be analyzed according to the characteristics of the target data.

However, the field of hydrological forecasting is not limited to flood forecasting direction. For medium and long-term runoff forecasting work with a larger time span, this paper still lacks corresponding research. Whether there will be any difference in the advantages and disadvantages of the RNN model and the LSTM model in a larger time span unit structure and what are the reasons for these differences needs further exploration. This is of great significance for understanding the application of various deep learning models in the hydrology direction and using them to obtain a clear runoff mechanism.

Author Contributions: Y.W.: Methodology, Program implementation, Writing—original draft. W.W.: Conceptualization, data curation, Writing—original draft preparation. H.Z.: Writing and editing—original draft. D.X.: Writing—original draft, Formal analysis. All authors have read and agreed to the published version of the manuscript.

Funding: The project of Key Science and Technology of the Henan province (202102310259; 202102310588), and the Henan province University Scientific and Technological Innovation team (No: 18IRTSTHN009).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data cannot be made publicly available; readers should contact the corresponding author for details.

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

References

- Collet, L.; Beevers, L.; Stewart, M.D. Decision-Making and Flood Risk Uncertainty: Statistical Data Set Analysis for Flood Risk Assessment. *Water Resour. Res.* **2018**, *54*, 7291–7308. [\[CrossRef\]](#)
- Herath, H.; Chadalawada, J.; Babovic, V. Hydrologically informed machine learning for rainfall-runoff modelling: Towards distributed modelling. *Hydrol. Earth Syst. Sci.* **2021**, *25*, 4373–4401. [\[CrossRef\]](#)
- Hao, S.; Wang, W.; Ma, Q.; Li, C.; Wen, L.; Tian, J.; Liu, C. Model-Based Mechanism Analysis of “7.20” Flash Flood Disaster in Wangzongdian River Basin. *Water* **2023**, *15*, 304. [\[CrossRef\]](#)
- Wang, W.-C.; Zhao, Y.-W.; Chau, K.-W.; Xu, D.-M.; Liu, C.-J. Improved flood forecasting using geomorphic unit hydrograph based on spatially distributed velocity field. *J. Hydroinformatics* **2021**, *23*, 724–739. [\[CrossRef\]](#)
- Lian, X.; Hu, X.L.; Bian, J.; Shi, L.S.; Lin, L.; Cui, Y.L. Enhancing streamflow estimation by integrating a data-driven evapotranspiration submodel into process-based hydrological models. *J. Hydrol.* **2023**, *621*, 129603. [\[CrossRef\]](#)

6. Yang, S.Y.; Yang, D.W.; Chen, J.S.; Santisirisomboon, J.; Lu, W.W.; Zhao, B.X. A physical process and machine learning combined hydrological model for daily streamflow simulations of large watersheds with limited observation data. *J. Hydrol.* **2020**, *590*, 125206. [[CrossRef](#)]
7. Li, B.-J.; Sun, G.-L.; Liu, Y.; Wang, W.-C.; Huang, X.-D. Monthly Runoff Forecasting Using Variational Mode Decomposition Coupled with Gray Wolf Optimizer-Based Long Short-term Memory Neural Networks. *Water Resour. Manag.* **2022**, *36*, 2095–2115. [[CrossRef](#)]
8. Yuan, X.; Wang, J.H.; He, D.M.; Lu, Y.; Sun, J.R.; Li, Y.; Guo, Z.P.; Zhang, K.Y.; Li, F. Influence of cascade reservoir operation in the Upper Mekong River on the general hydrological regime: A combined data-driven modeling approach. *J. Environ. Manag.* **2022**, *324*, 116339. [[CrossRef](#)]
9. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [[CrossRef](#)]
10. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010. [[CrossRef](#)]
11. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [[CrossRef](#)]
12. Chen, C.; Luan, D.B.; Zhao, S.; Liao, Z.; Zhou, Y.; Jiang, J.G.; Pei, Q.Q. Flood Discharge Prediction Based on Remote-Sensed Spatiotemporal Features Fusion and Graph Attention. *Remote Sens.* **2021**, *13*, 5023. [[CrossRef](#)]
13. Li, W.; Kiaghadi, A.; Dawson, C. Exploring the best sequence LSTM modeling architecture for flood prediction. *Neural Comput. Appl.* **2021**, *33*, 5571–5580. [[CrossRef](#)]
14. Chen, P.-A.; Chang, L.-C.; Chang, F.-J. Reinforced recurrent neural networks for multi-step-ahead flood forecasts. *J. Hydrol.* **2013**, *497*, 71–79. [[CrossRef](#)]
15. Kao, I.F.; Liou, J.-Y.; Lee, M.-H.; Chang, F.-J. Fusing stacked autoencoder and long short-term memory for regional multistep-ahead flood inundation forecasts. *J. Hydrol.* **2021**, *598*, 126371. [[CrossRef](#)]
16. Zou, Y.; Wang, J.; Lei, P.; Li, Y. A novel multi-step ahead forecasting model for flood based on time residual LSTM. *J. Hydrol.* **2023**, *620*, 129521. [[CrossRef](#)]
17. Andréassian, V.; Perrin, C.; Berthet, L.; Le Moine, N.; Lerat, J.; Loumagne, C.; Oudin, L.; Mathevet, T.; Ramos, M.H.; Valéry, A. HESS Opinions “Crash tests for a standardized evaluation of hydrological models”. *Hydrol. Earth Syst. Sci.* **2009**, *13*, 1757–1764. [[CrossRef](#)]
18. Beven, K. Changing ideas in hydrology—The case of physically-based models. *J. Hydrol.* **1989**, *105*, 157–172. [[CrossRef](#)]
19. Holländer, H.M.; Blume, T.; Bormann, H.; Buytaert, W.; Chirico, G.B.; Exbrayat, J.F.; Gustafsson, D.; Hölzel, H.; Kraft, P.; Stamm, C.; et al. Comparative predictions of discharge from an artificial catchment (Chicken Creek) using sparse data. *Hydrol. Earth Syst. Sci.* **2009**, *13*, 2069–2094. [[CrossRef](#)]
20. Perrin, C.; Michel, C.; Andréassian, V. Does a large number of parameters enhance model performance? Comparative assessment of common catchment model structures on 429 catchments. *J. Hydrol.* **2001**, *242*, 275–301. [[CrossRef](#)]
21. 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](#)]
22. Kang, J.L.; Wang, H.M.; Yuan, F.F.; Wang, Z.Q.; Huang, J.; Qiu, T. Prediction of Precipitation Based on Recurrent Neural Networks in Jingdezhen, Jiangxi Province, China. *Atmosphere* **2020**, *11*, 246. [[CrossRef](#)]
23. Le, X.-H.; Hung Viet, H.; Lee, G.; Jung, S. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water* **2019**, *11*, 1387. [[CrossRef](#)]
24. Gholami, H.; Mohammadifar, A.; Golzari, S.; Song, Y.; Pradhan, B. Interpretability of simple RNN and GRU deep learning models used to map land susceptibility to gully erosion. *Sci. Total Environ.* **2023**, *904*, 166960. [[CrossRef](#)] [[PubMed](#)]
25. Byeon, W.; Breuel, T.M.; Raue, F.; Liwicki, M. Scene labeling with LSTM recurrent neural networks. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3547–3555. [[CrossRef](#)]
26. Eck, D.; Schmidhuber, J. A First Look at Music Composition Using LSTM Recurrent Neural Networks. 2002. Available online: <https://people.idsia.ch/~juergen/blues/IDSIA-07-02.pdf> (accessed on 15 March 2002).
27. Graves, A. Generating Sequences With Recurrent Neural Networks. *arXiv* **2013**, arXiv:1308.0850.
28. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
29. Chen, C.; Jiang, J.G.; Zhou, Y.; Lv, N.; Liang, X.X.; Wan, S.H. An edge intelligence empowered flooding process prediction using Internet of things in smart city. *J. Parallel Distrib. Comput.* **2022**, *165*, 66–78. [[CrossRef](#)]
30. Peng, T.; Zhang, C.; Zhou, J.Z.; Xia, X.; Xue, X.M. Multi-Objective Optimization for Flood Interval Prediction Based on Orthogonal Chaotic NSGA-II and Kernel Extreme Learning Machine. *Water Resour. Manag.* **2019**, *33*, 4731–4748. [[CrossRef](#)]
31. Li, T.; Li, J.B.; Zhang, D.D. Yellow River flooding during the past two millennia from historical documents. *Prog. Phys. Geogr. Earth Environ.* **2020**, *44*, 661–678. [[CrossRef](#)]
32. Sherstinsky, A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [[CrossRef](#)]
33. Jiang, S.J.; Zheng, Y.; Wang, C.; Babovic, V. Uncovering Flooding Mechanisms Across the Contiguous United States Through Interpretive Deep Learning on Representative Catchments. *Water Resour. Res.* **2022**, *58*, e2021WR030185. [[CrossRef](#)]

34. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
35. Ding, Y.K.; Zhu, Y.L.; Feng, J.; Zhang, P.C.; Cheng, Z.R. Interpretable spatio-temporal attention LSTM model for flood forecasting. *Neurocomputing* **2020**, *403*, 348–359. [[CrossRef](#)]
36. Ahmadlou, M.; Ghajari, Y.E.; Karimi, M. Enhanced classification and regression tree (CART) by genetic algorithm (GA) and grid search (GS) for flood susceptibility mapping and assessment. *Geocarto Int.* **2022**, *37*, 13638–13657. [[CrossRef](#)]
37. Pelikan, M.; Goldberg, D.E.; Cantú-Paz, E. BOA: The Bayesian optimization algorithm. In Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation, San Francisco, CA, USA, 13–17 July 1999; Volume 1, pp. 525–532. [[CrossRef](#)]
38. Alizadeh, B.; Bafti, A.G.; Kamangir, H.; Zhang, Y.; Wright, D.B.; Franz, K.J. A novel attention-based LSTM cell post-processor coupled with bayesian optimization for streamflow prediction. *J. Hydrol.* **2021**, *601*, 126526. [[CrossRef](#)]
39. Japel, R.C.; Buyel, J.F. Bayesian optimization using multiple directional objective functions allows the rapid inverse fitting of parameters for chromatography simulations. *J. Chromatogr. A* **2022**, *1679*, 463408. [[CrossRef](#)]
40. Abidi, M.A.; Gribok, A.V.; Paik, J. Selection of the Regularization Parameter. In *Optimization Techniques in Computer Vision: Ill-Posed Problems and Regularization*; Springer: Cham, Switzerland, 2016; pp. 29–50. [[CrossRef](#)]
41. Adil, M.; Ullah, R.; Noor, S.; Gohar, N. Effect of number of neurons and layers in an artificial neural network for generalized concrete mix design. *Neural Comput. Appl.* **2022**, *34*, 8355–8363. [[CrossRef](#)]
42. Iiduka, H. Appropriate Learning Rates of Adaptive Learning Rate Optimization Algorithms for Training Deep Neural Networks. *IEEE Trans. Cybern.* **2022**, *52*, 13250–13261. [[CrossRef](#)]
43. Chadalawada, J.; Babovic, V. Review and comparison of performance indices for automatic model induction. *J. Hydroinform.* **2019**, *21*, 13–31. [[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.