Journal of
Hydroinformatics

Journal of Hydroinformatics Vol 25 No 6, 2561 doi: 10.2166/hydro.2023.160

A new stable and interpretable flood forecasting model combining multi-head attention mechanism and multiple linear regression

Yi-yang Wang[a], Wenchuan Wang[a,*], Kwok-wing Chau[b], Dong-mei Xu[a], Hong-fei Zang[a], Chang-jun Liu[c] and Qiang Ma[c]

[a] College of Water Resources, North China University of Water Resources and Electric Power, Zhengzhou 450046, China
[b] Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China
[c] Research Center on Flood and Drought Disaster Reduction, China Institute of Water Resources and Hydropower Research, Beijing 100081, China
*Corresponding author. E-mail: wangwen1621@163.com

WW, 0000-0003-1367-5886

ABSTRACT

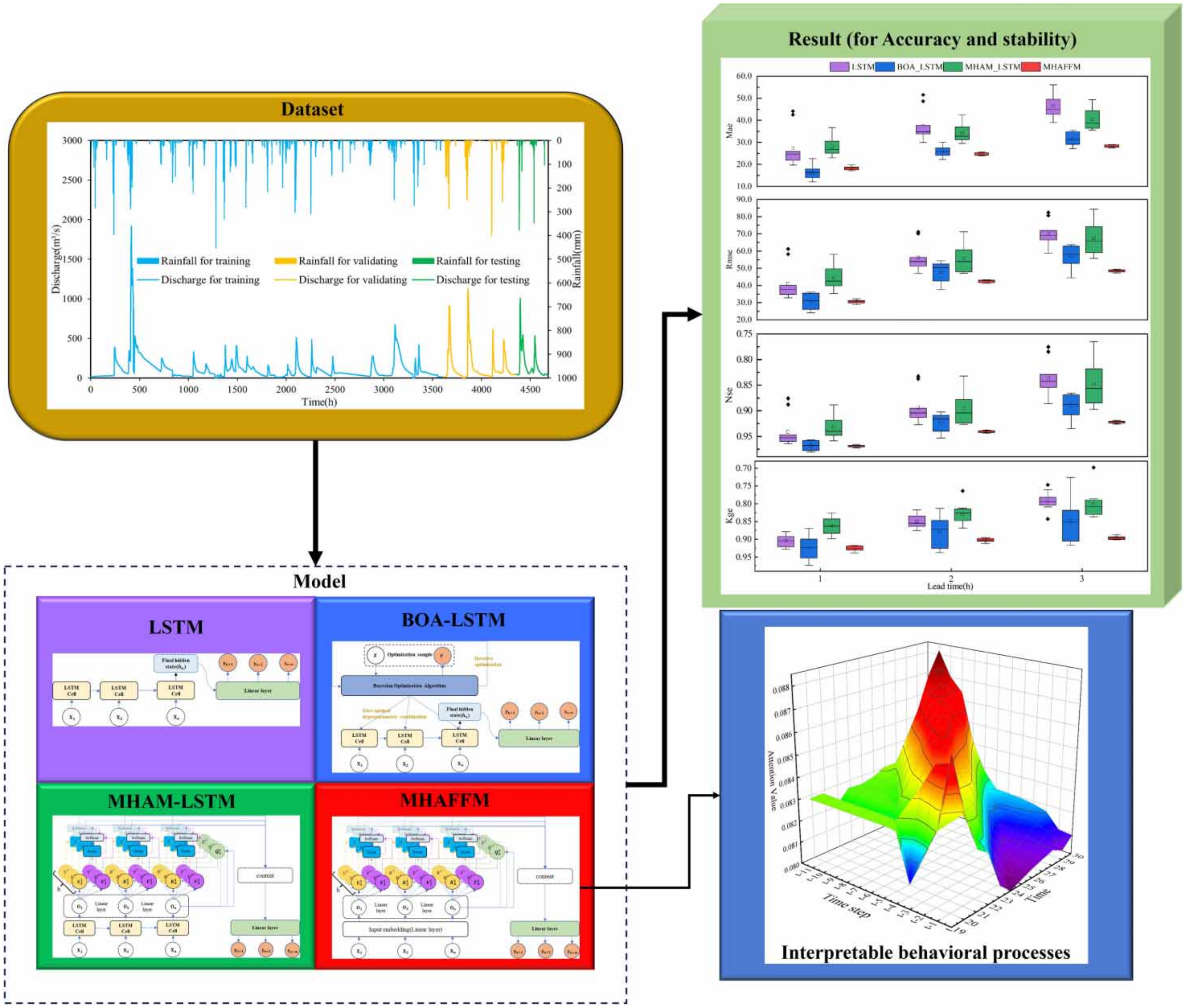
This article proposes a multi-head attention flood forecasting model (MHAFFM) that combines a multi-head attention mechanism (MHAM) with multiple linear regression for flood forecasting. Compared to models based on Long Short-Term Memory (LSTM) neural networks, MHAFFM enables precise and stable multi-hour flood forecasting. First, the model utilizes characteristics of full-batch stable input data in multiple linear regression to solve the problem of oscillation in the prediction results of existing models. Second, full-batch information is connected to MHAM to improve the model's ability to process and interpret high-dimensional information. Finally, the model accurately and stably predicts future flood processes through linear layers. The model is applied to Dawen River Basin, and experimental results show that the MHAFFM, compared to three benchmarking models, namely, LSTM, BOA-LSTM (LSTM with Bayesian Optimization Algorithm for Hyperparameter Tuning), and MHAM-LSTM (LSTM model with MHAM in hidden layer), significantly improves the prediction performance under different lead time scenarios while maintaining good stability and interpretability. Taking Nash–Sutcliffe efficiency index as an example, under a lead time of 3 h, the MHAFFM model exhibits improvements of 8.85, 3.71, and 10.29% compared to the three benchmarking models, respectively. This research provides a new approach for flood forecasting.

Key words: flood forecasting, LSTM, multi-head attention mechanism, multiple linear regression

HIGHLIGHTS

- Proposes a novel multi-head attention flood forecasting model (MHAFFM).
- Multi-head attention mechanism strengthens the model's ability to handle high-dimensional data.
- Linear layers effectively harness the performance of the multi-head attention mechanism.
- MHAFFM significantly enhances the stability of forecasted results.
- Even in longer lead time scenarios, the model maintains high accuracy and stability.

GRAPHICAL ABSTRACT



1. INTRODUCTION

Existing flood forecasting models can be roughly divided into two types according to the driving process. One is a process-driven hydrological model based on the specific physical process of flow generation and confluence (Wang *et al.* 2021), and the other is a data-driven model based on measured hydrological-related data (Zhang *et al.* 2020; Cao *et al.* 2022; Gao *et al.* 2022). The former is limited by insufficient knowledge of some flood processes, and it is difficult to describe the mechanism of specific hydrological processes in detail (Chomba *et al.* 2022). The latter is widely used in flood forecasting because it only needs to capture the relationship between input and output and does not need to describe complex physical processes (Liang *et al.* 2018; Liu *et al.* 2019; Krisnayanti *et al.* 2022; Yuan *et al.* 2022). However, classical data-driven models have a premise that the residual sequence is independent and obeys the normal distribution. It is difficult for a general hydrological sequence to meet this condition, which poses certain limitations to this type of method (Wang *et al.* 2012).

With the improvement of computing power, data-driven machine learning models, especially deep learning models, have shown powerful learning capabilities (Ekwueme 2022). They are not constrained by hypothetical principles and are widely used in flood forecasting (Ditthakit *et al.* 2023; Min *et al.* 2023; Xie *et al.* 2023). Among various deep learning models, the recurrent neural network model (RNN) can express spatiotemporal anisotropy and has structural advantages in time series data prediction (Elman 1990; Frame *et al.* 2022). However, due to the forward and backward transmission of weights of the RNN model, longer sequences are prone to gradient explosion or disappearance, which limits the application of the model (Noh 2021). In response to this problem, Hochreiter & Schmidhuber (1997) constructed an LSTM model based on the RNN model and used gating units to filter data, which effectively reduced the occurrence of these problems. Hu *et al.* (2018) used this model for rainfall–runoff modeling for the first time and achieved good results. Afterward, the LSTM model has been widely used in runoff prediction and achieved good prediction results (Abbas *et al.* 2020; Gao *et al.* 2020; Cao *et al.* 2022).

However, due to the LSTM model being a black box model, the randomness of the gating unit for data screening leads to poor interpretability of the model, making it difficult for practitioners to fully trust this type of model (Herath *et al.* 2021; Li *et al.* 2022a; Paudel *et al.* 2023). In 2019, the International Association of Hydrological Sciences emphasized in the 23 unresolved issues that it was of great significance to solve the confusion and reduce the uncertainty of model structure, parameters, and inputs in hydrological prediction (Blöschl *et al.* 2019). Jiang *et al.* (2022) also demonstrated the significance of interpreting deep learning models in understanding the mechanisms of flood formation through their research. It can be seen that the interpretability of the model cannot be ignored in hydrological forecasting.

In addressing the issue of model interpretability, Cai *et al.* (2022) improved model performance by adding physical mechanism constraints to the model, ensuring that the recursive process of the model aligns with physical mechanisms. Chadalawada *et al.* (2020) are dedicated to flexibly combining machine learning algorithms to construct rainfall–runoff model components with physical significance. In the field of deep learning, Bahdanau *et al.* (2014) proposed an attention mechanism that was embedded into the hidden layer of RNN models to improve model performance and observe the degree of model attention to data, thus enhancing model interpretability. RNN combined with the attention mechanism has achieved good performance in many fields (Xu *et al.* 2019; Zhao *et al.* 2020; Hwang *et al.* 2021).

In the direction of hydrological forecasting, Chen *et al.* (2020) combined the LSTM model with a self-attention mechanism for daily runoff forecasting, and the results were more accurate than those of the LSTM model alone. Gao *et al.* (2022) combined the attention mechanism with the gated recurrent unit (GRU) model, introduced a linear layer for input information processing, and used a seq2seq architecture for multistep flood forecasting, which improved the accuracy of flood forecasting. However, in the current stage of hydrological forecasting, research on deep learning models incorporating attention mechanisms is primarily focused on improving prediction accuracy, with a limited investigation into model prediction result stability. This lack of exploration hinders the practical application and promotion of interpretable deep learning models in real-world scenarios.

Therefore, besides model prediction accuracy, this study also pays particular attention to the distribution of model performance indicators. Through comparing the changes in the model's predicted outcome metrics, we find that the existing coupling methods that combine the LSTM model with the attention mechanism although can balance interpretability and the average performance of flood forecasting results, the stability of the performance metrics is poor. In extreme cases, it can even be worse than the basic model, which reduces the reliability of the model and makes it unsuitable for practical flood forecasting applications. To address this issue, this article analyzes factors that affect the stability of the model and replaces the traditional use of LSTM as the hidden layer by using multiple linear regression to stabilize the input data in a full batch. By doing so, it overcomes the problem of model prediction result oscillations and constructs a stable deep learning model.

In addition, in current flood forecasting research, the attention mechanism or self-attention mechanism combined with the LSTM model is generally adopted. However, due to their structural characteristics, these two mechanisms have limited ability to extract features from a high-dimensional array, and it is difficult to establish accurate affine transformation for a nonlinear flood routing process that is influenced by multiple factors, which also affects the model accuracy and result stability (Vaswani *et al.* 2017). In response to this problem, existing research directions in hydrological forecasting typically address it through data processing, with commonly used methods including principal component analysis and data decomposition (Sarraf 2015; Adnan *et al.* 2021; Carreau & Guinot 2021). Although these processing methods can improve the model performance, they require preprocessing of all runoff data, and future information is introduced into the data processing process, which the model lacks realistic conditions in actual flood forecasting (Zhang *et al.* 2015; Tan *et al.* 2018; Zuo *et al.* 2020).

To enhance the model's practicality, this article introduces a multi-head attention architecture from the perspective of model design. The multi-head attention mechanism was introduced in an article by the Google team on building the transformer model, aimed at handling high-dimensional data information (Vaswani *et al.* 2017). Currently, the ChatGPT model based on this architecture has achieved tremendous success in various applications. However, in the field of hydrology, there is limited research that utilizes this architecture for prediction tasks. Therefore, this article adopts the multi-head attention mechanism to solve the problem of high-dimensional array processing for runoff data by utilizing its characteristic of partitioning information subspaces for high-dimensional data and improving the model's applicability.

Overall, this article draws on ideas of multiple linear regression and multi-head attention architecture to propose the multi-head attention flood forecasting model (MHAFFM) model, breaking away from the LSTM model framework and using a new architectural form to improve the accuracy and stability of deep learning model predictions for flood forecasting. In addition, it also analyzes the interpretability of the MHAFFM model to help understand the behavioral logic of flood forecasting models and improve the model's trustworthiness. The main contributions of this article are as follows:

(1) The introduction of the multi-head attention mechanism, utilizing its ability to jointly focus on different subspaces of data, enhances the model's feature extraction capability for flood processes. It provides a novel way for model predictions in high-dimensional data scenarios.
(2) By modifying the hidden layer of the model based on the concept of multiple linear regression, we enable the full-batch input information to enter the attention architecture, establishing the MHAFFM model, addressing the stability issue in flood forecasting model predictions caused by the LSTM model framework.
(3) By analyzing the behavioral logic of the MHAFFM model, a highly interpretable flood forecasting model with high accuracy is provided.

The rest of this article is arranged as follows: Section 2 introduces three models for comparison: LSTM, BOA-LSTM, and MHAM-LSTM. The MHAFFM model is used as the benchmark model. Model evaluation metrics are also provided. Sections 3–5 present the results of the study conducted at Dawen River Basin in Shandong Province, China. Finally, Section 6 provides a summary of the article.

2. METHODOLOGY

2.1. LSTM

The LSTM model belongs to the RNN type. Its structure is based on the continuity of time series information, which is input into the hidden layer according to the time step for weight verification. When compared with other types of neural network models, the LSTM model has the following advantages: (1) Due to the structure of its processing information, this type of model can learn spatiotemporal correlation of data without encoding the time position of the data when processing time series information. (2) The LSTM model, as an improved version of the RNN model, introduces a gated recurrent unit to alleviate the problem of gradient dispersion and explosion, which improves its applicability.

The information flow process of LSTM is shown in Figure 1.

For the input information X_{n-1} at time $n-1$ and the inheritance information H_{n-2} and C_{n-2} at time $n-2$, the information flow process is as follows:

$$\Gamma_f^{\langle n-1\rangle} = \sigma(W_f[H_{n-2}, X_{n-1}] + b_f) \tag{1}$$

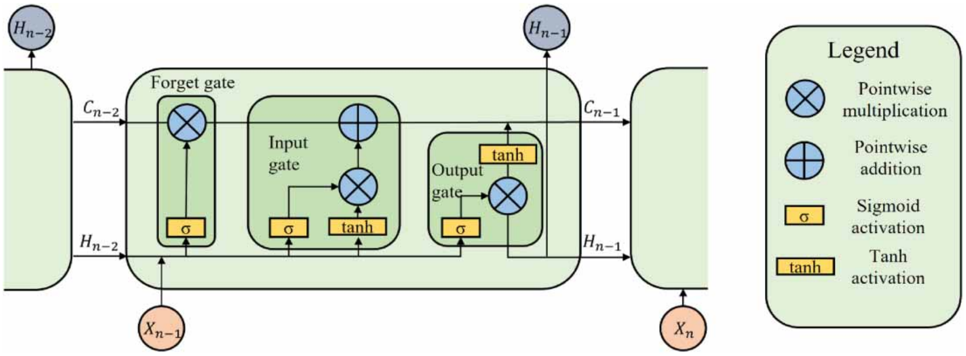


Figure 1 | Schematic diagram of LSTM cell structure.

In the forget gate, the previous memory state is discarded through $\Gamma_f^{\langle n-1\rangle}$. W_f is the weight that controls the behavior of the forget gate. $[H_{n-2}, X_{n-1}]$ are concatenated, multiplied by W_f, and added to the bias b_f. The result is then activated by the sigmoid function to obtain a value between 0 and 1. $\Gamma_f^{\langle n-1\rangle}$ is multiplied by the previous memory state C_{n-2} to represent the selective inheritance of information:

$$\Gamma_i^{\langle n-1\rangle} = \sigma(W_i[H_{n-2}, X_{n-1}] + b_i) \tag{2}$$

$$\tilde{C}^{\langle n-1\rangle} = \tanh(W_c[H_{n-2}, X_{n-1}] + b_c) \tag{3}$$

$$C^{\langle n-1\rangle} = \Gamma_f^{\langle n-1\rangle} * C_{n-2} + \Gamma_i^{\langle n-1\rangle} * \tilde{C}^{\langle n-1\rangle} \tag{4}$$

In the input gate, a new vector $\tilde{C}^{\langle n-1\rangle}$ is created through Equation (3) to update the cell state. $\Gamma_i^{\langle n-1\rangle}$ is multiplied with $\tilde{C}^{\langle n-1\rangle}$ to reflect the current state, representing the selective input of information. Equation (4) sums up the selectively input information to output the new cell state $C^{\langle n-1\rangle}$:

$$\Gamma_o^{\langle n-1\rangle} = \sigma(W_o[H_{n-2}, X_{n-1}] + b_o) \tag{5}$$

$$H_{n-1} = \Gamma_o^{\langle n-1\rangle} * \tan h\left(C^{\langle n-1\rangle}\right) \tag{6}$$

In the output gate, the new cell state $C^{\langle n-1\rangle}$ is activated by the tan$h$ function and a new hidden state H_{n-1} is selected for output by $\Gamma_o^{\langle n-1\rangle}$.

The LSTM model is based on the unit shown in Figure 1. The overall information flow architecture is shown in Figure 2.

Information of early-stage precipitation $\{x_1, x_2 \ldots x_n\}$ is passed through the LSTM unit, and at time n, all temporal information that has passed through gating units is accumulated in h_n. h_n is then fed into a linear layer to establish a linear relationship between it and the predicted runoff $\{y_{n+1}, y_{n+2}, y_{n+m}\}$.

2.2. BOA-LSTM

To better demonstrate the performance advantages of the MHAFFM model over the LSTM model and avoid subjective biases caused by manually selecting hyperparameters, this article employs the Bayesian optimization algorithm to determine the globally optimal hyperparameters for the LSTM model and construct the BOA-LSTM model for comparative experiments (Pelikan 2005). The specific process of the Bayesian optimization algorithm is detailed in the original article and is not repeated here. The Bayesian optimization algorithm is known for its high efficiency and global optimization capability, and it is widely used in model parameter optimization tasks (Jäpel & Buyel 2022; Mao *et al.* 2022). The architecture of BOA-LSTM model is shown in Figure 3. The optimized learning rate, number of neurons, and regularization parameter for LSTM model are listed in Table 1.

2.3. MHAM-LSTM

To address the issue of gradient propagation in RNN-based models, researchers generally place attention layers after the hidden layer and directly establish a mapping between the input and output time step information (Bahdanau *et al.* 2014;

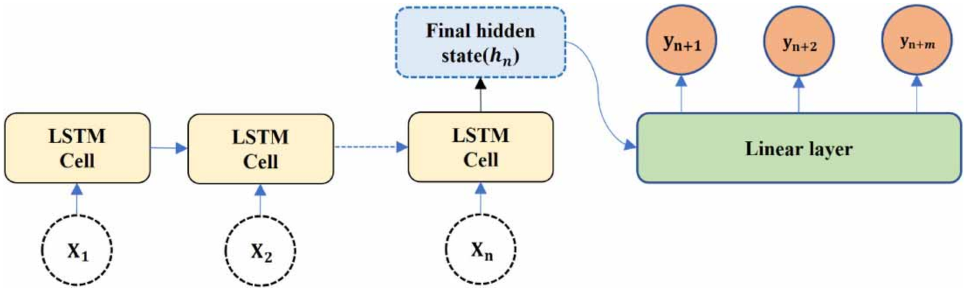


Figure 2 | LSTM model architecture.

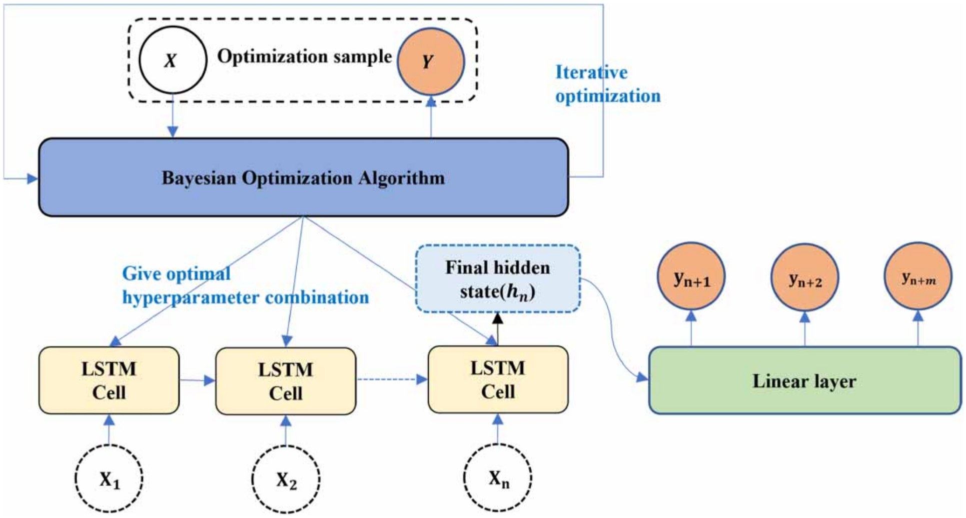


Figure 3 | BOA-LSTM model architecture.

Table 1 | Optimal hyperparameters combination of LouDe section

Learning rate	Hidden units	L2 regularization
6.291784×10^{-3}	180	1.49×10^{-6}

Moreno-Pino *et al.* 2023; Zhao *et al.* 2023). Currently, attention layers generally use attention mechanisms or self-attention mechanisms to establish mappings, but this approach has limited processing capacity for high-dimensional information. To address this problem, this article introduces a multi-head attention mechanism and makes adaptive modifications to establish the attention layer to enhance information acquisition capabilities. The model architecture is shown in Figure 4.

For the hidden state sequence $\{H_1,H_2\ldots H_n\}$ output by the LSTM unit, the multi-head attention mechanism processes are as follows:

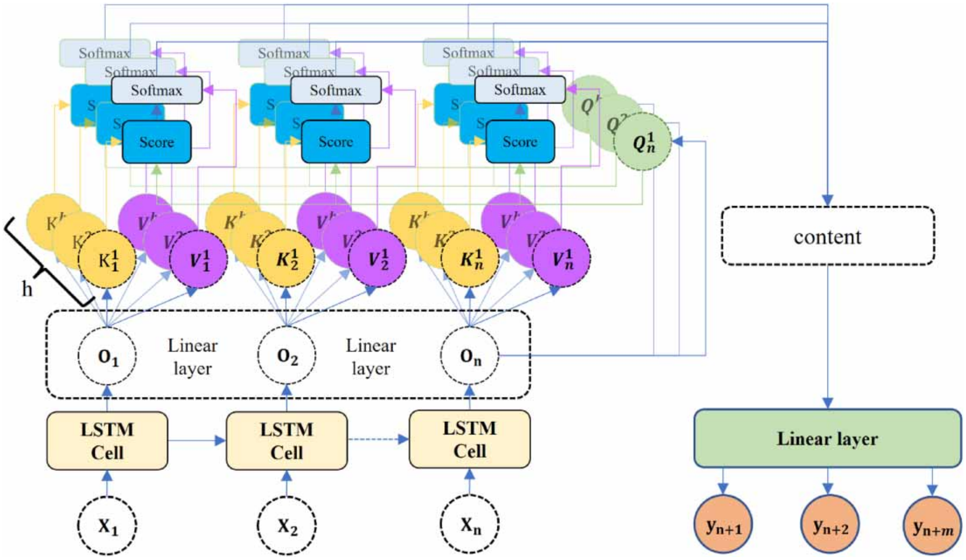


Figure 4 | MHAM-LSTM model architecture.

(1) The hidden state sequence $\{H_1,H_2...H_n\}$ output by the LSTM unit is fed into a linear layer, which produces the query sequence $\{Q_1, Q_2 \ldots Q_n\}$, the key sequence $\{K_1, K_2 \ldots K_n\}$, and the value sequence $\{V_1, V_2 \ldots V_n\}$, respectively.

(2) The three sequences are divided into h equal parts (number of heads for the multi-head attention mechanism) to obtain the subsequences $\{Q_1^1, Q_1^2 \ldots Q_1^h\}$, $\{Q_2^1, Q_2^2 \ldots Q_2^h\}$, $\{Q_n^1, Q_n^2 \ldots Q_n^h\}$; $\{K_1^1, K_1^2 \ldots K_1^h\}$, $\{K_2^1, K_2^2 \ldots K_2^h\}$, $\{K_n^1, K_n^2 \ldots K_n^h\}$; and $\{V_1^1, V_1^2 \ldots V_1^h\}$, $\{V_2^1, V_2^2 \ldots V_2^h\}$, $\{V_n^1, V_n^2 \ldots V_n^h\}$.

(3) Using the Q sequence at time step n as the final query and performing vector scaling dot product with the K sequence, compute the attention scores for each time step. Taking the computation process of the hth head at time step n as an example:

$$h(\alpha_n^1, \alpha_n^1 \ldots \alpha_n^n) = \frac{(K_1^h, K_2^h \ldots K_n^h) * (Q_n^h)^T}{\sqrt{d^k}} \tag{7}$$

where d^k represents the dimension of sequence K and α_n^i denotes the attention weight that Q_n imposes to K_i at time step n $(i = 1\ldots\text{n})$.

(4) The attention scores $h(\alpha_n^1, \alpha_n^1 \ldots \alpha_n^n)$ for h heads are then normalized and activated by the softmax function, resulting in the sequence $h(\hat{\alpha}_n^1, \hat{\alpha}_n^2 \ldots \hat{\alpha}_n^n)$:

$$\hat{\alpha}_n^i = \frac{\exp(\alpha_n^i)}{\sum_{j=1}^{n} \exp(\alpha_n^j)} \tag{8}$$

(5) The activated attention sequence $h(\hat{\alpha}_n^1, \hat{\alpha}_n^2 \ldots \hat{\alpha}_n^n)$ is then multiplied element-wise with the value sequence $(V_1^h, V_2^h \ldots V_n^h)$ and summed up to obtain the computation content for the hth head:

$$content(h) = h(\hat{\alpha}_n^1, \hat{\alpha}_n^2 \ldots \hat{\alpha}_n^n) * (V_1^h, V_2^h \ldots V_n^h) \tag{9}$$

(6) After concatenating the computed contents from all heads, the sequence is fed into a linear layer to obtain the output sequence $\{y_{n+1}, y_{n+2} \ldots y_{n+m}\}$:

$$\{y_{n+1}, y_{n+2} \ldots y_{n+m}\} = \text{linear(content)} \tag{10}$$

The linear layer 'linear' is a linear function of the form $y = kx + b$.

When compared to the traditional multi-head attention mechanism, the adaptive modification is to compute attention for the final question Q_n instead of the question sequence $\{Q_1, Q_2 \ldots Q_n\}$. The reason is as follows:

The traditional multi-head attention mechanism was originally proposed to solve natural language processing (NLP) problems with a seq2seq architecture, where there is temporal parallelism between the input and output sequences and a close context relationship within the sequences. Specifically, there is a possibility of correlation between Q_1 and $\{K_1, K_2 \ldots K_n\}$, and $\{Q_1, Q_2 \ldots Q_n\}$ contains similar levels of information. However, for flood forecasting problems, the output sequence usually lags behind the input sequence and there is temporal unidirectionality. There is no physical correlation between Q_1 and $\{K_2 \ldots K_n\}$, and $\{Q_1, Q_2 \ldots Q_{n-1}\}$ contains lower amounts of information compared to Q_n. Therefore, Q_n is used to replace $\{Q_1, Q_2 \ldots Q_n\}$ for attention computation.

2.4. MHAFFM

Through experiments, it has been found that although MHAM-LSTM model can improve the accuracy of flood forecasting results, its stability is poor. The reasons for this phenomenon are as follows:

(1) The limited amount of observed data for flood forecasting makes it difficult to support the stable identification of the global optimal solution by the model.

Due to the limited development time of observation techniques, the dataset used in flood forecasting tasks is relatively small. Compared to GB-level datasets in NLP tasks, it is difficult to support the model in stably finding the global optimum. This can cause the model to fall into other local optimal solutions, thereby reducing the stability of the model.

(2) Due to the gating mechanism of LSTM hidden layer, it is difficult to feed the entire batch of flood data into the attention layer.

As shown in Equations (1)–(4), the hidden layer selectively inherits previous information and selectively inputs current information. While this information processing method alleviates the gradient problem, it undoubtedly damages the input information required by the model, also leading to reduced model stability.

The first point is limited by actual conditions and has little room for improvement. However, regarding the method of information transmission, this article attempts to use full-batch input to reconstruct the hidden layer to improve the information acquisition of the attention layer. To achieve full-batch data passing through the hidden layer, this article introduces the idea of multivariate linear regression:

The idea of multiple linear regression is to establish a linear relationship between the independent variables and the dependent variable, as shown in Equation (11):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \ + \beta_n x_n + \varepsilon \tag{11}$$

where y is the dependent variable, $x_1, x_2, \ \ldots, x_n$ are independent variables, $\beta_0, \beta_1, \beta_2, \ \ldots, \beta_n$ are model parameters, and ε is the error term.

This article uses a linear layer to imitate the linear relationship between input factors and the target value in multiple linear regression, and constructs a hidden layer to pass the data information in full batch. The model structure is shown in Figure 5.

This article uses a linear layer as the input encoding layer to replace the LSTM layer to process input information, in order to ensure that all input information is passed to the multi-head attention structure. When compared with MHAM-LSTM, the advantages of this model are as follows: (1) replacing the LSTM layer with a simple linear layer for input data encoding, which improves the speed of the model; and (2) breaking away from the influence of LSTM model and improving the utilization of input data.

2.5. Model evaluation indices

Based on previous literature research, appropriate indicators were selected to evaluate the model performance (Chadalawada & Babovic 2017). Nash–Sutcliffe efficiency (NSE), root mean square error (RMSE), mean absolute error (MAE), Kling–Gupta efficiency coefficient (KGE), peak error (TPE), and absolute percentage bias (APB) are used as performance evaluation

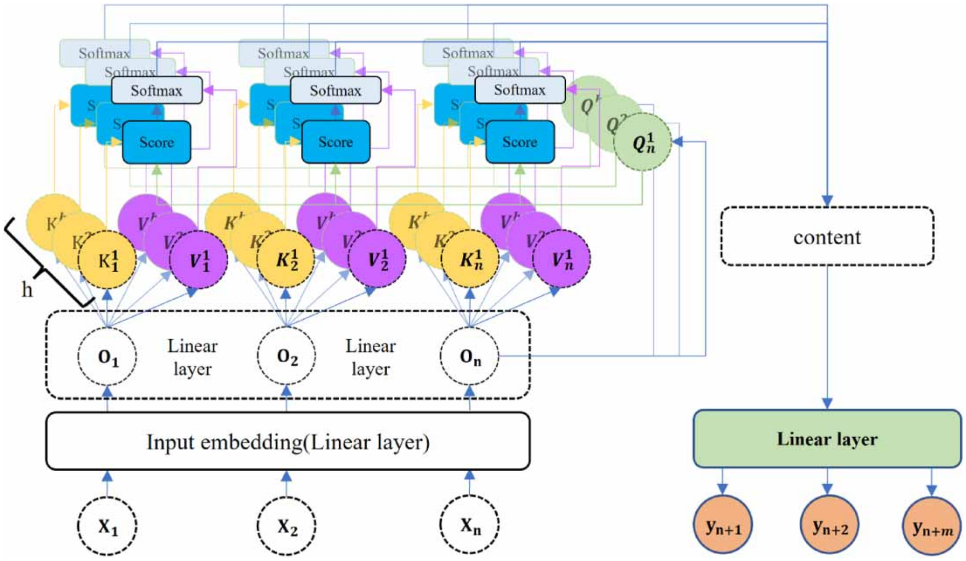


Figure 5 | MHAFFM model architecture.

metrics for the model. Their formulas are as follows:

$$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{N} (Q_i - P_i)^2}{\sum_{i=1}^{N} (Q_i - Q_{\mathrm{avg}})^2} \tag{12}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (Q_i - P_i)^2}{N}} \tag{13}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |Q_i - P_i| \tag{14}$$

$$\mathrm{KGE} = 1 - \sqrt{(\alpha - 1)^2 + (\beta - 1)^2 + (R - 1)^2} \tag{15}$$

$$\mathrm{TPE} = \frac{\sum_{j=1}^{h} |Q_j - P_j|}{\sum_{j=1}^{h} Q_j} \tag{16}$$

$$\mathrm{APB} = \left| \frac{\sum_{i=1}^{N} (Q_i - P_i)}{\sum_{i=1}^{N} Q_i} \right| \tag{17}$$

where N represents 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 value of observed runoff, $\alpha = \sigma_p/\sigma_o$ is the coefficient of variability bias, $\beta = \mu_p/\mu_o$ is the coefficient of mean bias (σ and μ represent standard deviation and mean, respectively), R is the linear correlation coefficient, h represents the peak flow of the top 2% of the flood events, Q_j is the observed peak flow, and P_j is the predicted peak flow.

NSE is sensitive to the fluctuations of the data series and can characterize the tracking ability of the predicted values to the actual values, which is used to evaluate the prediction stability of the model (Kumar *et al.* 2016). RMSE and MAE are used to compute errors of predicted values, indicating the overall prediction accuracy of the model (Ćalasan *et al.* 2020). KGE combines model correlation, bias, and flow variability into a single metric to evaluate the model's performance. TPE is used to evaluate the prediction accuracy of the top 2% of the runoff process of the model (Gao *et al.* 2022). APB is used to evaluate the prediction error of the flood volume of the model (Miao *et al.* 2022).

NSE, RMSE, MAE, and KGE are used to analyze the performance of the model on the entire flood process, while TPE and APB are used to analyze the prediction effect of the model on single flood events.

2.6. Methodological framework

To facilitate an understanding of the logical relationships between the models of different methods, the model difference diagram and comparative analysis diagram are shown in Figures 6 and 7.

3. RESEARCH AREA AND MODEL PARAMETERS

3.1. Research area and data

The research object of this article is Dawen River Basin in Shandong Province, China, which belongs to Yellow River Basin in the middle and lower reaches. Dawen River originates from north of Xuangu Mountain in Shandong, with a total length of 209 km and a basin area of 9,098 km^2. It flows from east to west and joins the Dongping Lake before flowing into the Yellow River, serving as the last tributary before the Yellow River flows into the sea (Li *et al.* 2022b). Due to the influence of the monsoon climate, more than 70% of the annual precipitation in this basin occurs during the flood season, making it prone to seasonal flood disasters. After the floods converge into the Yellow River, they may even pose a threat to the safety of the downstream main river channel. Therefore, it is necessary to establish an accurate and stable hydrological model framework for this region.

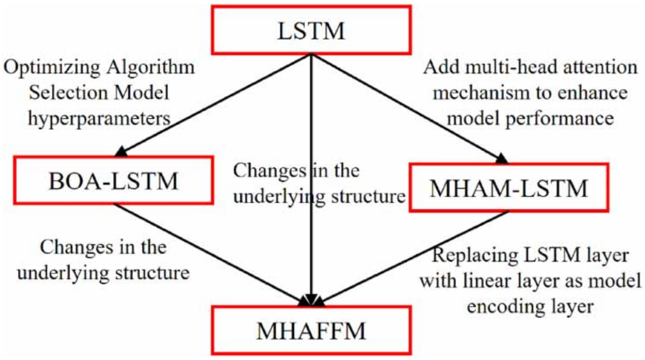

LSTM
Optimizing Algorithm Selection Model hyperparameters
Add multi-head attention mechanism to enhance model performance
BOA-LSTM
Changes in the underlying structure
MHAM-LSTM
Changes in the underlying structure
Replacing LSTM layer with linear layer as model encoding layer
MHAFFM

Figure 6 | Differences between different models.

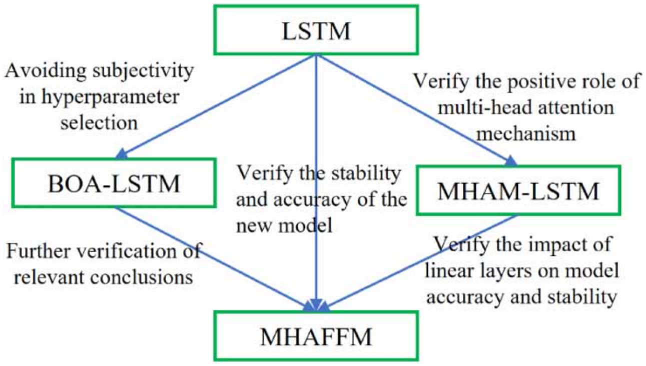

LSTM
Avoiding subjectivity in hyperparameter selection
Verify the positive role of multi-head attention mechanism
BOA-LSTM
Verify the stability and accuracy of the new model
MHAM-LSTM
Further verification of relevant conclusions
Verify the impact of linear layers on model accuracy and stability
MHAFFM

Figure 7 | Model analysis process.

Based on the measured data from hydrological and rainfall stations within the Dawen River basin, hourly runoff processes from the year 2000 to 2020 were compiled. The data source is the Shandong Hydrological and Water Resources Bureau of the Yellow River Conservancy Commission. Considering changes in underlying surface conditions, this article uses hourly runoff data from the Loude control section in the upstream basin and hourly flood event data from relevant rainfall stations from 2000 to 2020 for the model's flood process prediction performance analysis.

Loude station is located on the southern branch of Dawen River, which is a control station for the southern branch of Dawen River. There are two inflow sections, Guangming Reservoir, and Dongzhou Reservoir. Sixteen rainfall stations such as Xiafeng, Mengyinzhai, and Baoanzhuang are installed between Dongzhou and Guangming sections and Loude section, controlling the topography and river conditions above the control section as shown in Figure 8.

A total of 4,864 h of flood process data, including 22 flood events with instantaneous flow rates exceeding 200 m^3/s, are selected for Loude station. Four flood events are used for validation, and the last two flood events are used for testing. The data partitioning is shown in Figure 9.

3.2. Input and output sequence selection

When using rainfall–runoff data to generate the model dataset, time step and forecast lead time for input and output sequences of the dataset need to be selected. Increasing the time step can allow the model to acquire more relevant information and improve the accuracy of flood process prediction, but it will also increase the model's runtime. The time step is generally selected based on the basin's runoff time. Considering the size of the basin, the time step is set to 12 h, which enables the model to obtain relevant information on flood processes and better demonstrate the model's learning ability. The maximum forecast lead time is set to 3 h, which can demonstrate the model's ability to learn the rules of flood processes. In short, this article uses the rainfall–runoff information from time $t - 11$ to t to predict the flood process from time $t + 1$ to $t + 3$.

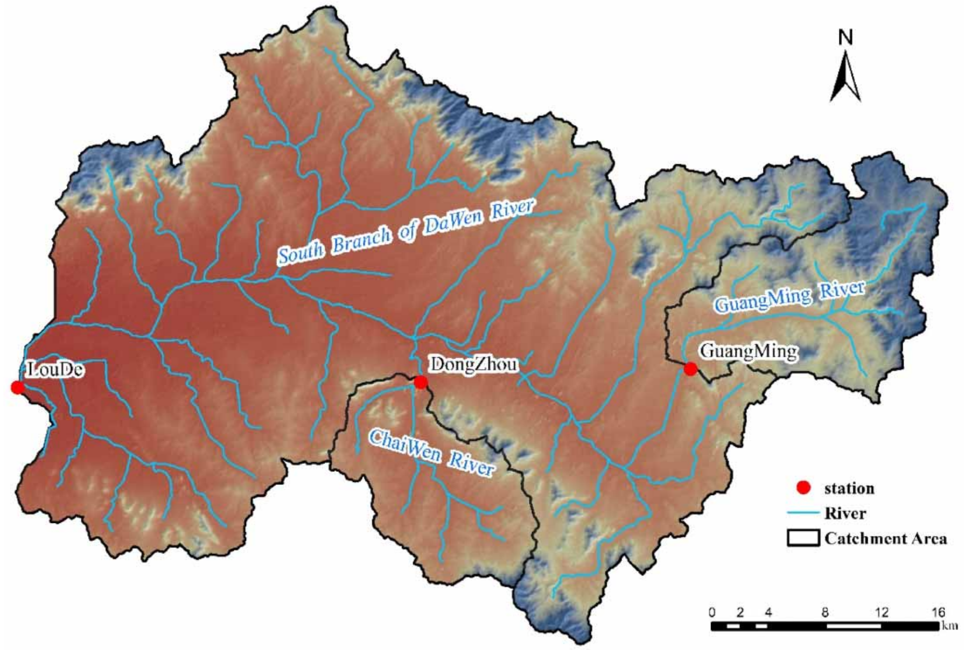


Figure 8 | Study basin information.

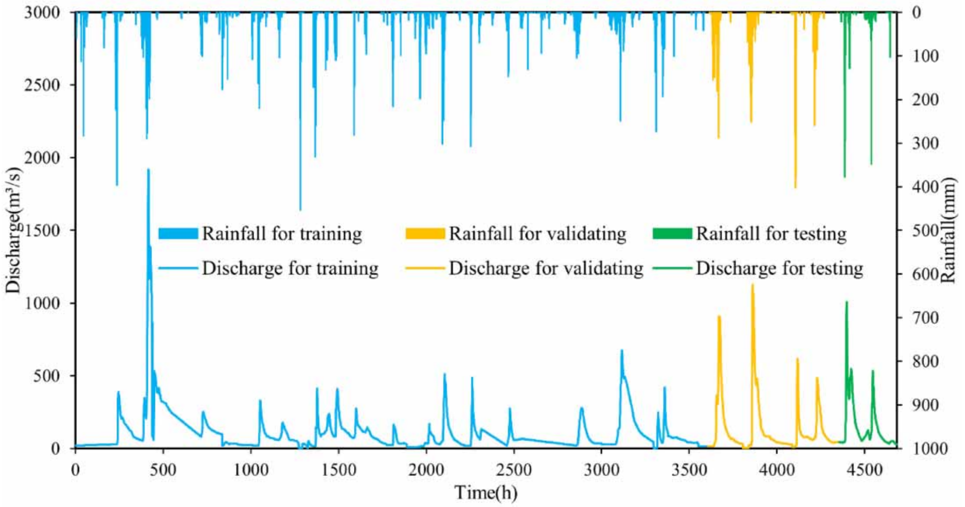


Figure 9 | Rainfall and discharge data division of LouDe reservoir section.

In addition, to improve the model's running speed, this article uses parallelization techniques to divide the dataset into subsets, each containing 500 input and output data (Dean *et al.* 2012).

3.3. Hyperparameter settings

To better utilize the performance of the models, the hyperparameters are adjusted in this article. Due to using LSTM as the baseline comparative model in the article, the performance of the LSTM model was taken as a reference to set the learning rate, the number of neurons, and regularization parameters.

The learning rate is set to 1×10^{-3}, which allows the LSTM model to converge stably to the optimal value and avoid the occurrence of divergent results. In addition, 128 neurons are used for all LSTM models with hidden layers, which are sufficient to fit the nonlinear characteristics of the watershed data and have reasonable computational costs. The LSTM model's regularization parameter is set to 1×10^{-5}, which helps prevent the LSTM model from overfitting.

To enhance the comparability between different models, the number of nonlinear expression neurons in the encoding layer of the MHAFFM model is also set to 128, giving the encoding layer the same nonlinear expression ability as the LSTM hidden layer. For the MHAM-LSTM model and the MHAFFM model that uses a multi-head attention structure, the number of model heads is set to 16, considering the data size to achieve multi-thread processing for the flood forecasting task. At the same time, this article uses the Adam optimization algorithm for gradient descent and the mean squared error to compute the iteration error loss. Each model is trained 1,500 times, from which the optimal solution is selected (Kingma & Ba 2014).

3.4. Data processing workflow

For ease of understanding, the data processing procedure in the article is summarized in a generalized form as shown in Figure 10.

4. RESULTS

4.1. Model performance

To validate the overall performance advantage of the MHAFFM model and the superiority of the multi-head attention mechanism, in addition to the four models constructed in this study, we also introduced the SAM-LSTM model (self-attention mechanism coupled with the LSTM model) proposed in related research in this section.

Average performances of MHAFFM, MHAM-LSTM, SAM-LSTM BOA-LSTM, and LSTM models in terms of NSE, RMSE, and MAE indicators during the testing period are shown in Figure 11. The variations of model performance with lead time are

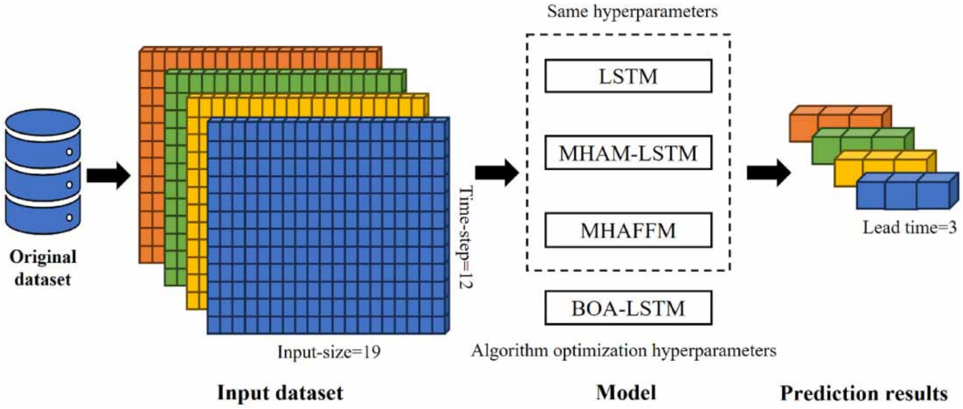


Figure 10 | Dataset processing flow.

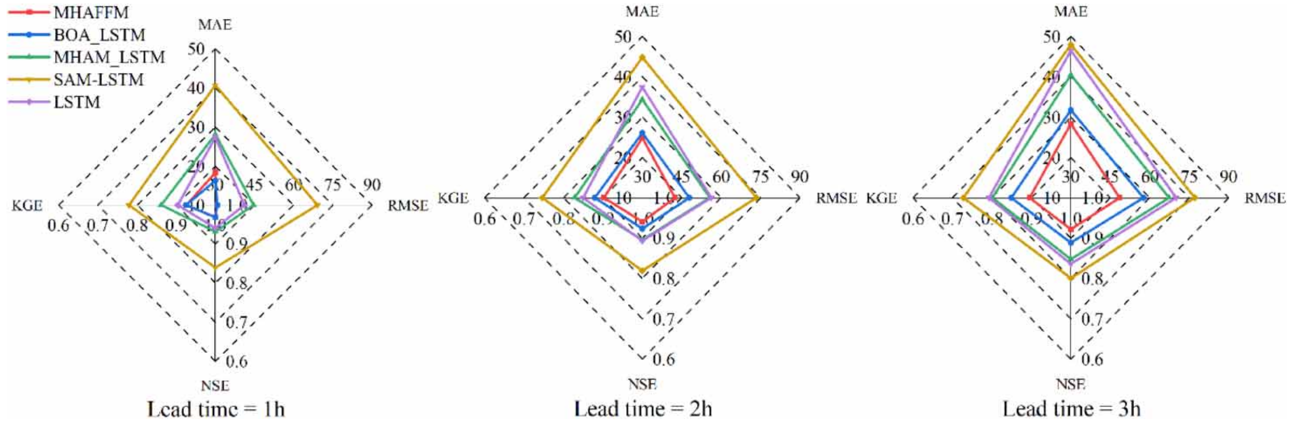


Figure 11 | Average performances of MHAFFM, MHAM-LSTM, SAM-LSTM, BOA-LSTM, and LSTM models on NSE, KGE, RMSE, and MAE indicators under different lead times.

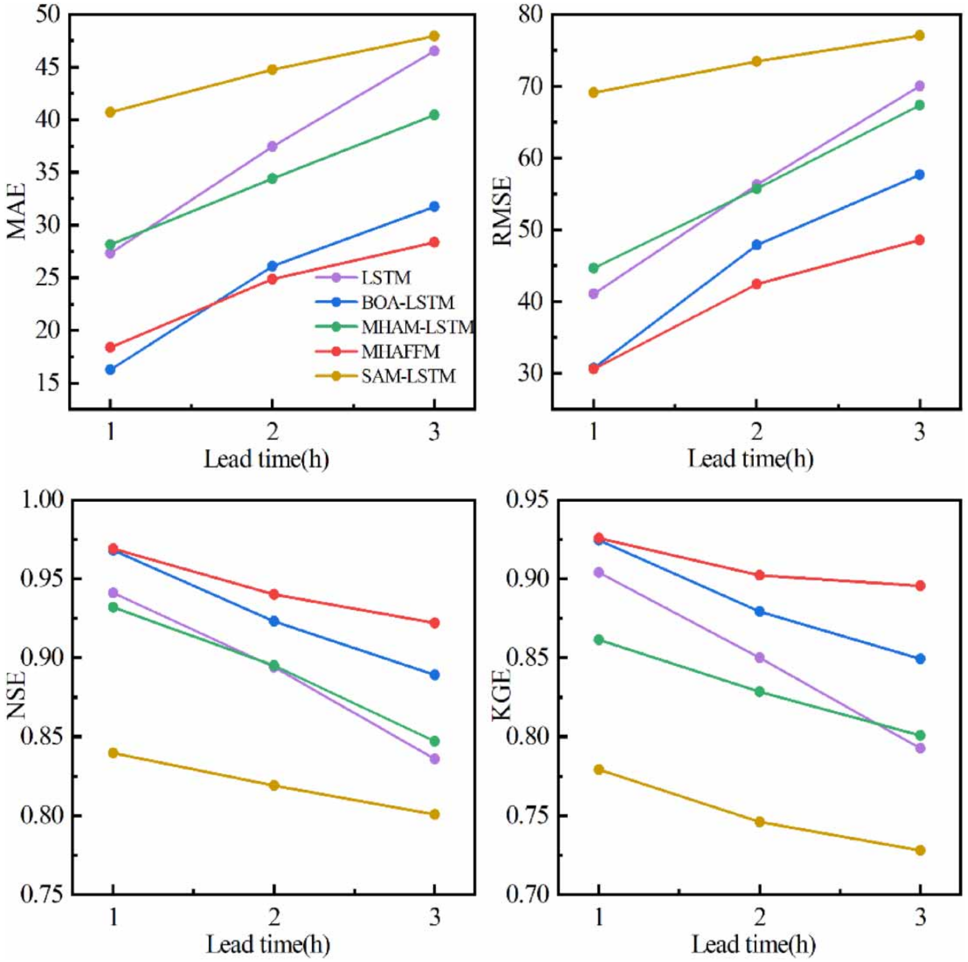


Figure 12 | Average performance of MHAFFM, MHAM-LSTM, SAM-LSTM, BOA-LSTM, and LSTM models in terms of MAE, RMSE, NSE, and KGE indicators under different lead times.

depicted in Figure 12. The specific values are presented in Table 2. The performance improvements of the MHAFFM model are listed in Table 3.

Based on the comprehensive analysis of Figures 11 and 12 and Tables 2 and 3, the MHAFFM model shows improved performance in all scenarios except for 1-h lead time where the MAE indicator is slightly lower than that of the BOA-LSTM model. Overall, the MHAFFM model outperforms other three benchmarking models in terms of performance indicators. Furthermore, the MHAFFM model exhibits a relatively small performance degradation phenomenon with an increase in lead time, and its performance advantage becomes more prominent compared to other models as the lead time increases. From this, it can be seen that although both approaches of algorithmic hyperparameter optimization and coupling attention mechanism can improve model performance, they are still inferior to the excellent performance of the MHAFFM model. In addition, the SAM-LSTM model constructed with the self-attention mechanism performed the worst in all indicators, while the MHAM-LSTM model constructed with the multi-head attention mechanism showed significant improvement compared to it. This indicates that the self-attention mechanism has poor adaptability to high-dimensional input data.

The stability performances of the five models on MAE, RMSE, NSE, and KGE indicators are shown in Figure 13.

When compared to MHAM-LSTM, SAM-LSTM, BOA-LSTM, and LSTM models, the MHAFFM model with a linear layer as the data input layer has significantly better stability than other models in all four metrics. On the other hand, the other four models with LSTM hidden layers have poor stability performance, with significant oscillations in their prediction results and lower reliability. As the lead time increases, the prediction results of the LSTM model, BOA-LSTM model, SAM-LSTM model, and MHAM-LSTM model all show increased oscillations. However, the MHAFFM model can still maintain excellent stability in its performance.

Table 2 | Average performances of MHAFFM, MHAM-LSTM, SAM-LSTM, BOA-LSTM, and LSTM models in terms of MAE, RMSE, NSE, and KGE indicators under different lead times

Model name						
Lead time (h)	**Performance metrics**	**LSTM**	**BOA-LSTM**	**SAM-LSTM**	**MHAM-LSTM**	**MHAFFM**
1	MAE	27.298	**16.235**	40.732	28.134	18.371
	RMSE	41.094	30.734	69.099	44.672	**30.618**
	NSE	0.941	0.968	0.840	0.932	**0.969**
	KGE	0.904	0.925	0.779	0.861	**0.926**
2	MAE	37.436	26.106	44.781	34.39	**24.875**
	RMSE	56.238	47.849	73.462	55.688	**42.423**
	NSE	0.894	0.923	0.819	0.895	**0.94**
	KGE	0.85	0.879	0.746	0.828	**0.902**
3	MAE	46.537	31.75	47.955	40.461	**28.356**
	RMSE	70.038	57.661	77.084	67.385	**48.577**
	NSE	0.836	0.889	0.801	0.847	**0.922**
	KGE	0.793	0.849	0.728	0.801	**0.896**

Note: The bold values represent the best performance for each model.

Table 3 | Average performance variations of MHAFFM model compared to MHAM-LSTM, SAM-LSTM, BOA-LSTM, and LSTM models in terms of MAE, RMSE, NSE, and KGE indicators under different lead times

Compared to (%)					
Lead time (h)	**Performance metrics**	**LSTM**	**BOA-LSTM**	**SAM-LSTM**	**MHAM-LSTM**
1	MAE	32.7	−13.16	54.9	34.7
	RMSE	25.49	0.38	55.69	31.46
	NSE	2.98	0.1	15.41	3.97
	KGE	2.41	0.13	18.84	7.48
2	MAE	33.55	4.72	44.45	27.67
	RMSE	24.57	11.34	42.25	23.82
	NSE	5.15	1.84	14.78	5.03
	KGE	6.16	2.64	20.89	8.93
3	MAE	39.07	10.69	40.87	29.92
	RMSE	30.64	15.75	36.98	27.91
	NSE	10.29	3.71	15.14	8.85
	KGE	13.01	5.46	23.09	11.84

As it can be observed, replacing the LSTM layer with a linear layer as the information processing layer in the multi-head attention mechanism can provide the model with more stable information, significantly enhance the overall control of the flooding process, and improve the performance in terms of MAE, RMSE, NSE, and KGE.

Furthermore, the SAM-LSTM model showed the poorest stability in its prediction results among the five models, indicating that its architecture is not suitable for the current flood forecasting task.

4.2. Model performance on single flood event

In terms of single flood forecasting performance, average evaluation indicators of the four models (due to the poor performance of the SAM-LSTM model, plotting its indicator distribution would affect the observation of the other models; and therefore, it is no longer evaluated) during the test period are shown in Figures 14–17, respectively, for TPE and APB indicators. Specific results are listed in Tables 4–7.

By combining Figures 14 and 15 with Tables 4 and 5, it can be observed that compared to the other three models, the performance advantage of the MHAFFM model in controlling flood peak and discharge for the first test event gradually becomes more prominent as the lead time increases. However, the performance of the MHAFFM model does not reach its optimal level at a 1-h lead time.

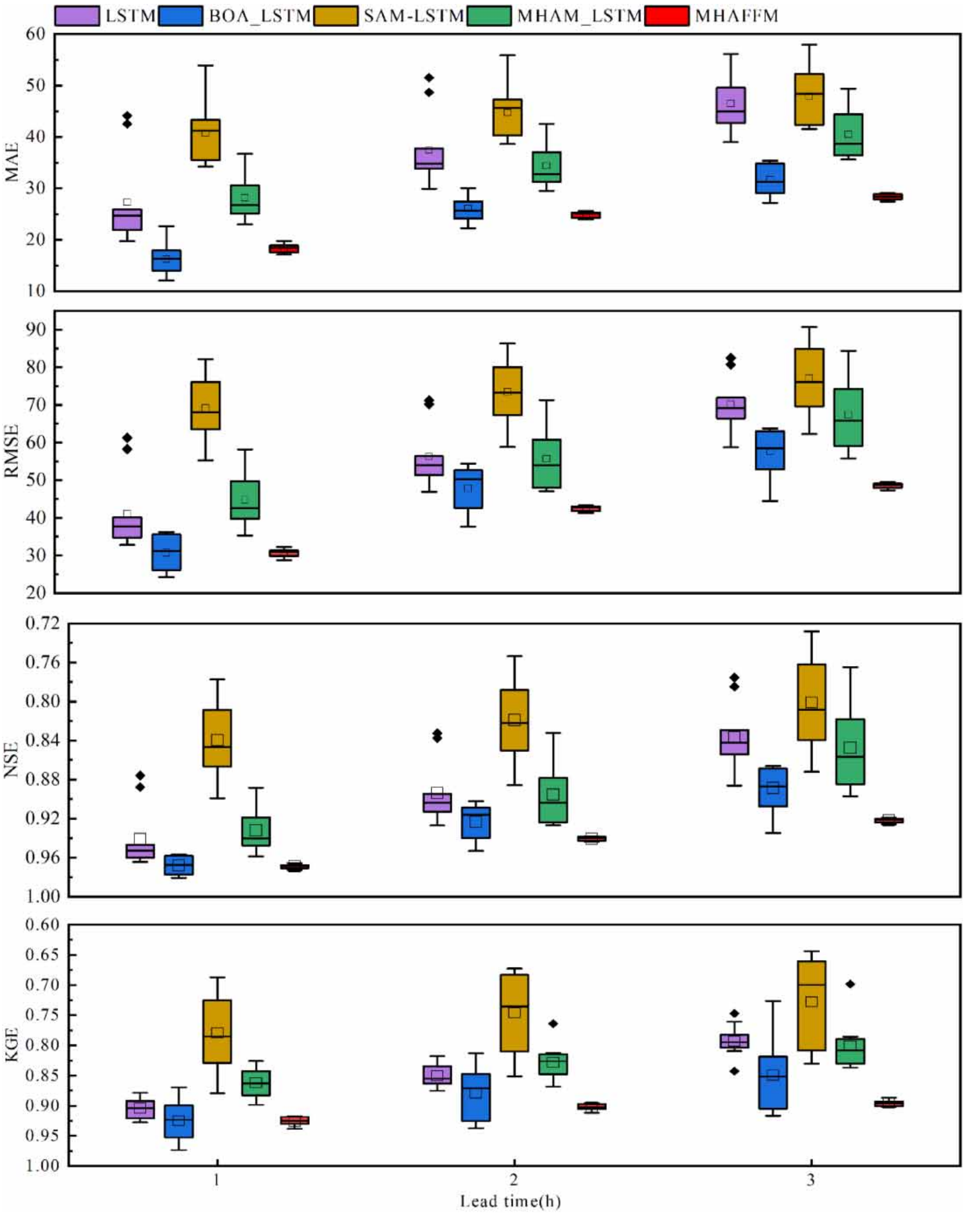


Figure 13 | Stability performances of MHAFFM, MHAM-LSTM, SAM-LSTM, BOA-LSTM, and LSTM models in terms of MAE, RMSE, NSE, and KGE indicators under different lead times.

By combining Figures 16 and 17 with Tables 6 and 7, it can be observed that compared to the other three models, the performance advantage of the MHAFFM model in controlling the flooding process for the second test event becomes more prominent as the lead time increases. However, there is a significant decline in performance on the 3-h TPE indicator.

Taking into account the overall performance of each model, in most cases, the MHAFFM model exhibits the best performance, but there are exceptions. Analyzing the exceptions, we believe that the following are the reasons: During the model training process, the search for the optimal solution is based on the entire flood dataset. As a result, the learned data relationships are guided by the overall optimization objective, making it challenging to achieve the best performance on specific indicators for individual flood events. However, the fact that the MHAFFM model generally outperforms the other three models also indirectly proves that it has learned data mapping relationships that are closer to reality compared to the other models.

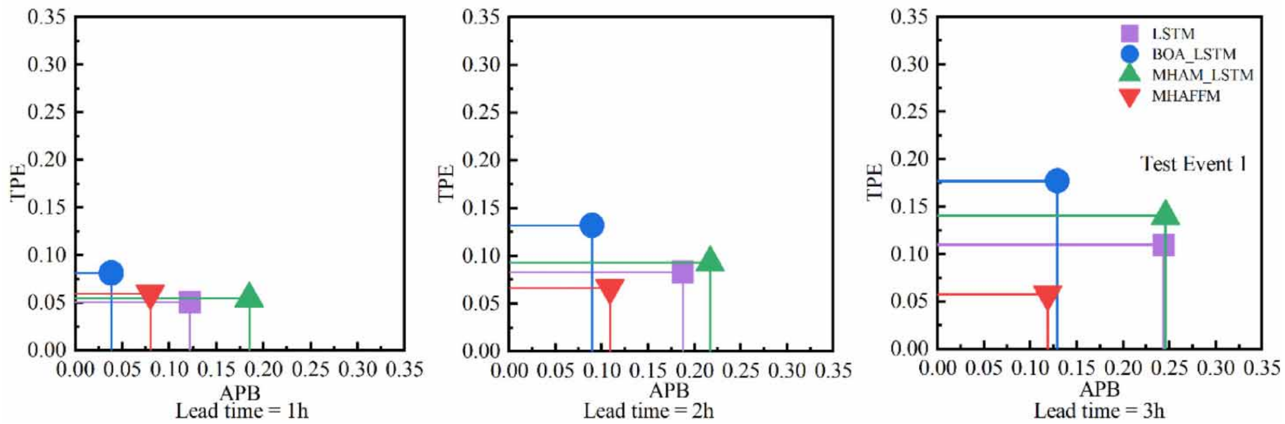


Figure 14 | Variations of TPE with APB indicators of MHAFFM, MHAM-LSTM, BOA-LSTM, and LSTM models under different lead times for test event 1.

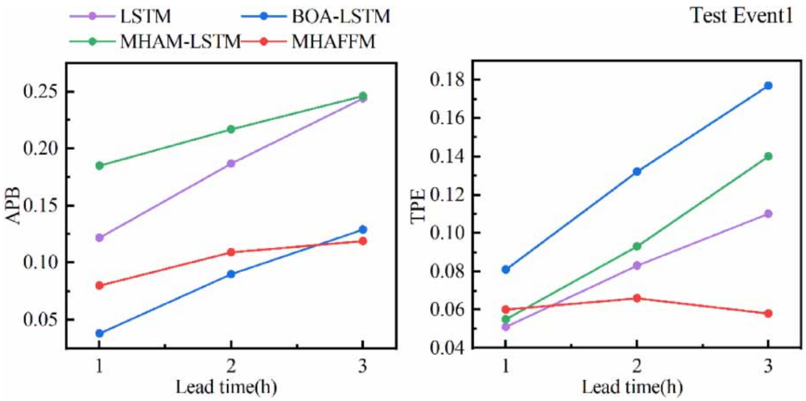


Figure 15 | Variations of TPE or APB indicators with lead times of MHAFFM, MHAM-LSTM, BOA-LSTM, and LSTM models for test event 1.

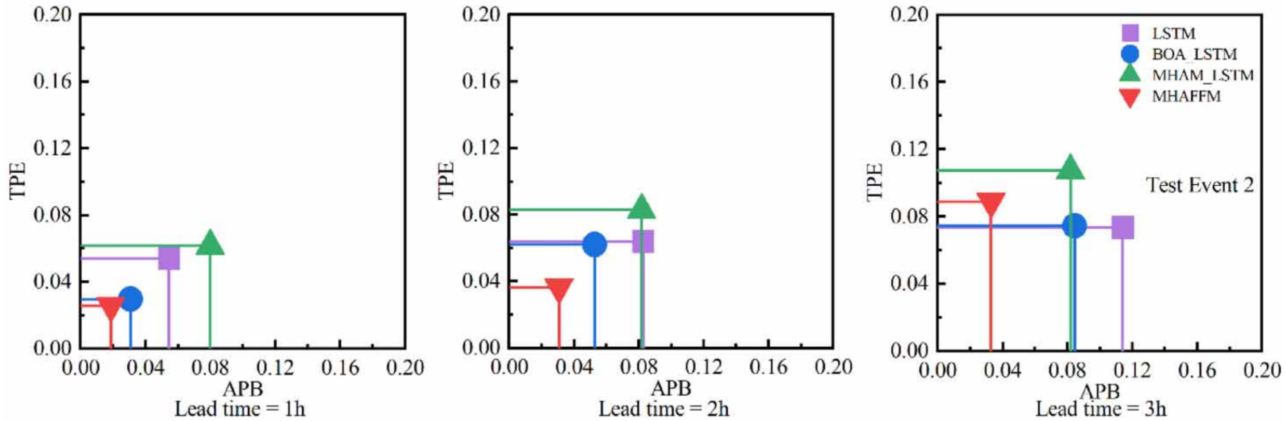


Figure 16 | Variations of TPE with APB indicators of MHAFFM, MHAM-LSTM, BOA-LSTM, and LSTM models under different lead times for test event 2.

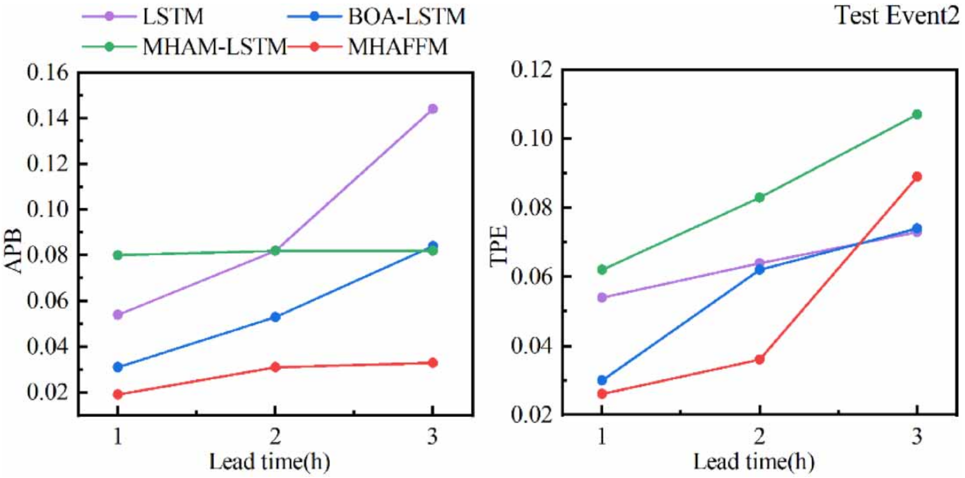


Figure 17 | Variations of TPE or APB indicators with lead times of MHAFFM, MHAM-LSTM, BOA-LSTM, and LSTM models for test event 2.

Table 4 | Average performances of MHAFFM, MHAM-LSTM, BOA-LSTM, and LSTM models in terms of TPE and APB indicators under different lead times for test event 1

Lead time (h)	1		2		3	
Performance metrics	APB	TPE	APB	TPE	APB	TPE
LSTM	0.122	**0.051**	0.187	0.083	0.244	0.110
BOA-LSTM	**0.038**	0.081	**0.090**	0.132	0.129	0.177
MHAM-LSTM	0.185	0.055	0.217	0.093	0.246	0.140
MHAFFM	0.080	0.060	0.109	**0.066**	**0.119**	**0.058**

Note: The bold values represent the best performance for each model.

Table 5 | Average performance variations of the MHAFFM model compared to MHAM-LSTM, BOA-LSTM, and LSTM models in terms of APB and TPE indicators under different lead times for test event 1

Lead time (h)	1		2		3	
Compared to	APB	TPE	APB	TPE	APB	TPE
LSTM	**34.43%**	−17.65%	**41.71%**	**20.48%**	**51.23%**	**47.27%**
BOA-LSTM	−110.53%	**25.93%**	−21.11%	**50.00%**	**7.75%**	**67.23%**
MHAM-LSTM	**56.76%**	−9.09%	**49.77%**	**29.03%**	**51.63%**	**58.57%**

Note: The bold values represent the best performance for each model.

Table 6 | Average performances of MHAFFM, MHAM-LSTM, BOA-LSTM, and LSTM models in terms of TPE and APB indicators under different lead times for test event 2

Lead time (h)	1		2		3	
Performance metrics	APB	TPE	APB	TPE	APB	TPE
LSTM	0.054	0.054	0.082	0.064	0.144	**0.073**
BOA-LSTM	0.031	0.030	0.053	0.062	0.084	0.074
MHAM-LSTM	0.080	0.062	0.082	0.083	0.082	0.107
MHAFFM	**0.019**	**0.026**	**0.031**	**0.036**	**0.033**	0.089

Note: The bold values represent the best performance for each model.

Table 7 | Average performance variations of the MHAFFM model compared to MHAM-LSTM, BOA-LSTM, and LSTM models in terms of APB and TPE indicators under different lead times for test event 2

Lead time (h)	1		2		3	
Compared to (%)	APB	TPE	APB	TPE	APB	TPE
LSTM	**64.82%**	**51.85%**	**62.20%**	**43.75%**	**77.08%**	−21.90%
BOA-LSTM	**38.71%**	**13.33%**	**41.51%**	**41.94%**	**60.71%**	−20.30%
MHAM-LSTM	**76.25%**	**58.07%**	**62.20%**	**56.63%**	**59.76%**	**16.82%**

Note: The bold values represent the best performance for each model.

The stability performance of the four models in terms of single flood evaluation indicators is shown in Figures 18 and 19. The stability of the MHAFFM model is still significantly better than that of other models and does not change with the increase in lead time. The stability of the indicators of the other three models is poor, and the degree of oscillation increases significantly with the increase in lead time.

Taking into consideration both Figures 18 and 19, for individual flood events, the performance of the MHAFFM model in predicting results remains stable. This indicates that the MHAFFM model consistently learns consistent flood process mapping relationships across multiple instances. Therefore, it demonstrates the model's outstanding data abstraction capability, which is both stable and reliable.

4.3. Fit of flood process

Flood process curves predicted by the four models during the testing period are shown in Figures 20 and 22. The scatter plots of the model predictions compared to the observed values are shown in Figure 21 and 23.

From the visual perspective, it can be observed from Figures 20–23 that the MHAFFM model fits the observed flood process more accurately than the other three models. The predicted flood process also exhibits higher correlation coefficients with the observed data.

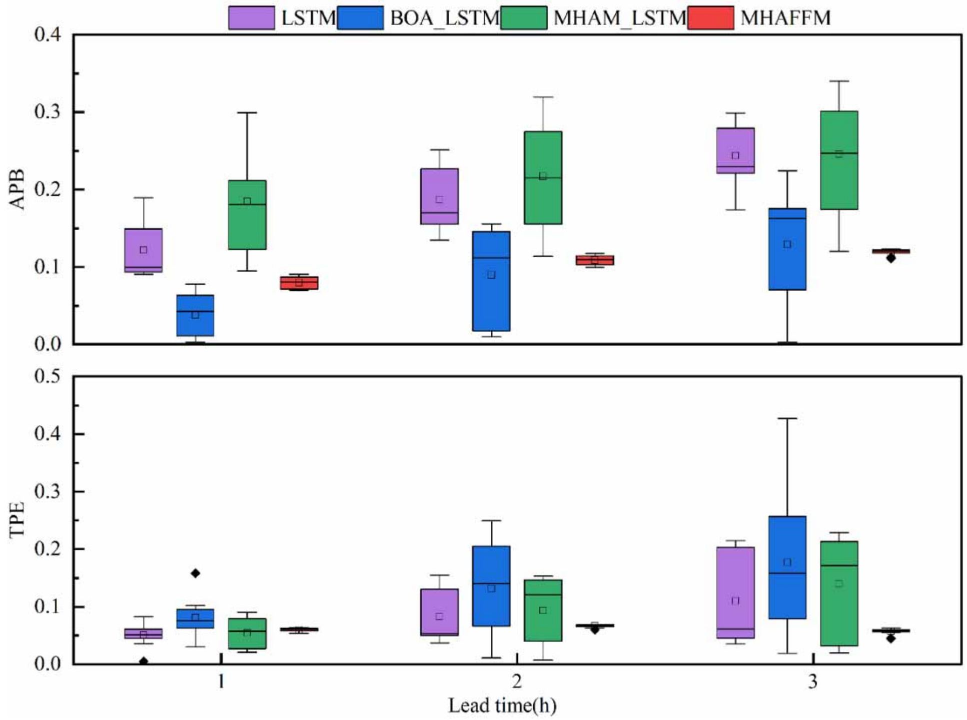


Figure 18 | Stability performance of MHAFFM, MHAM-LSTM, BOA-LSTM, and LSTM models in terms of TPE and APB indicators under different lead times for test event 1.

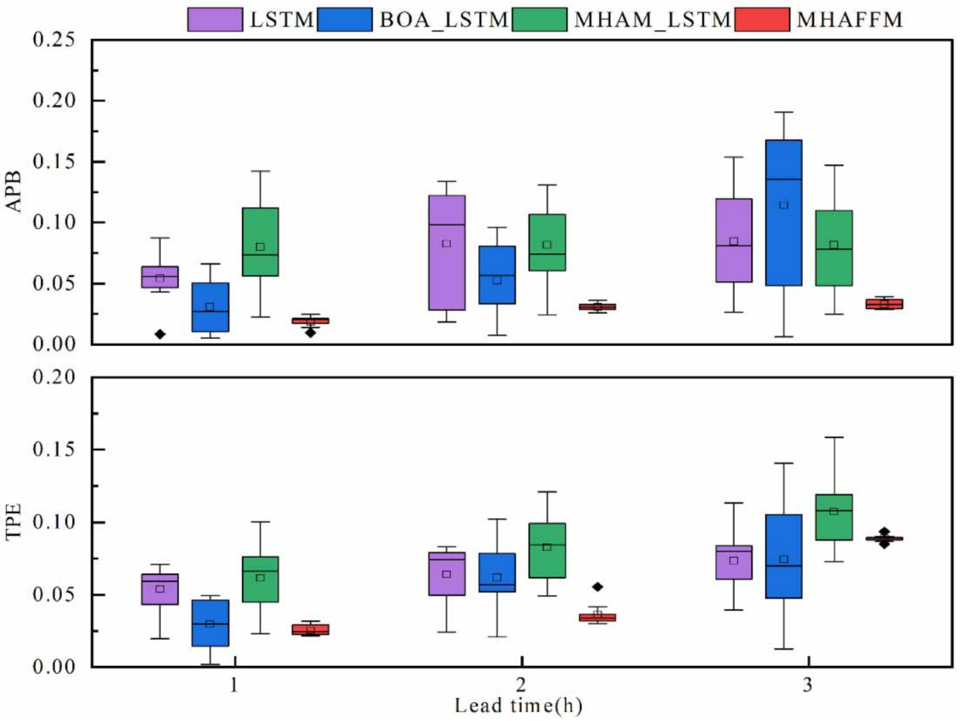


Figure 19 | Stability performance of MHAFFM, MHAM-LSTM, BOA-LSTM, and LSTM models in terms of TPE and APB indicators under different lead times for test event 2.

4.4. Visualization of attention mechanism in the MHAFFM model

To demonstrate the interpretability of the model, we select the period from 19 to 30 of flood process 1 to visualize the attention effects of 16 heads of the MHAFFM model, as shown in Figure 24. In addition, to facilitate the understanding of attention mechanisms of 16 heads, 16-dimensional data are reduced to one dimension, and the visualization effect is shown in Figure 25.

From Figure 24, it can be seen that the attention effects of 16 heads of the MHAFFM model are all different, indicating that the model generates differentiated attention effects for different subinformation spaces and can handle high-dimensional input information well. However, the model's division of data blocks into 16 parts for attention output makes it difficult to understand their mapping to the actual input. Therefore, this article reduces the 16-dimensional data to one dimension.

Based on Figure 25 and the flood timing information, it can be observed that for flood event 1 during the period of 19–30, which corresponds to the flood rising process, the MHAFFM model pays more attention to input information at time step 19, which corresponds to the first rainfall process in the physical space. Moreover, as the time step increases, the main focus is still on input information at that time step. By time step 30, the main focus of the flood peak is still on input information at time step t–11 (i.e., at 19:00). Furthermore, the model's main focus is in the vicinity of the attention space diagonal (i.e., around 19:00). This indicates that the flood generation process mainly originates from the rainfall process around time step 19, which is consistent with physical cognition, and the interpretability of the model is relatively good.

An analysis of the reasons behind this attention pattern is as follows: First, the generation of a reasonable attention space is based on the complete transmission of fundamental information. Since the linear layer does not filter information, it conveys all available information to the attention layer (which also contributes to the stability of the model's predictions), allowing the multi-head attention mechanism to process the complete information. Second, the multi-head attention mechanism assigns appropriate weights to the input information through the scoring function. The information vectors before 22:00 have higher scaled dot-product values, obtaining higher weights (as observed in the green portion above the attention space).

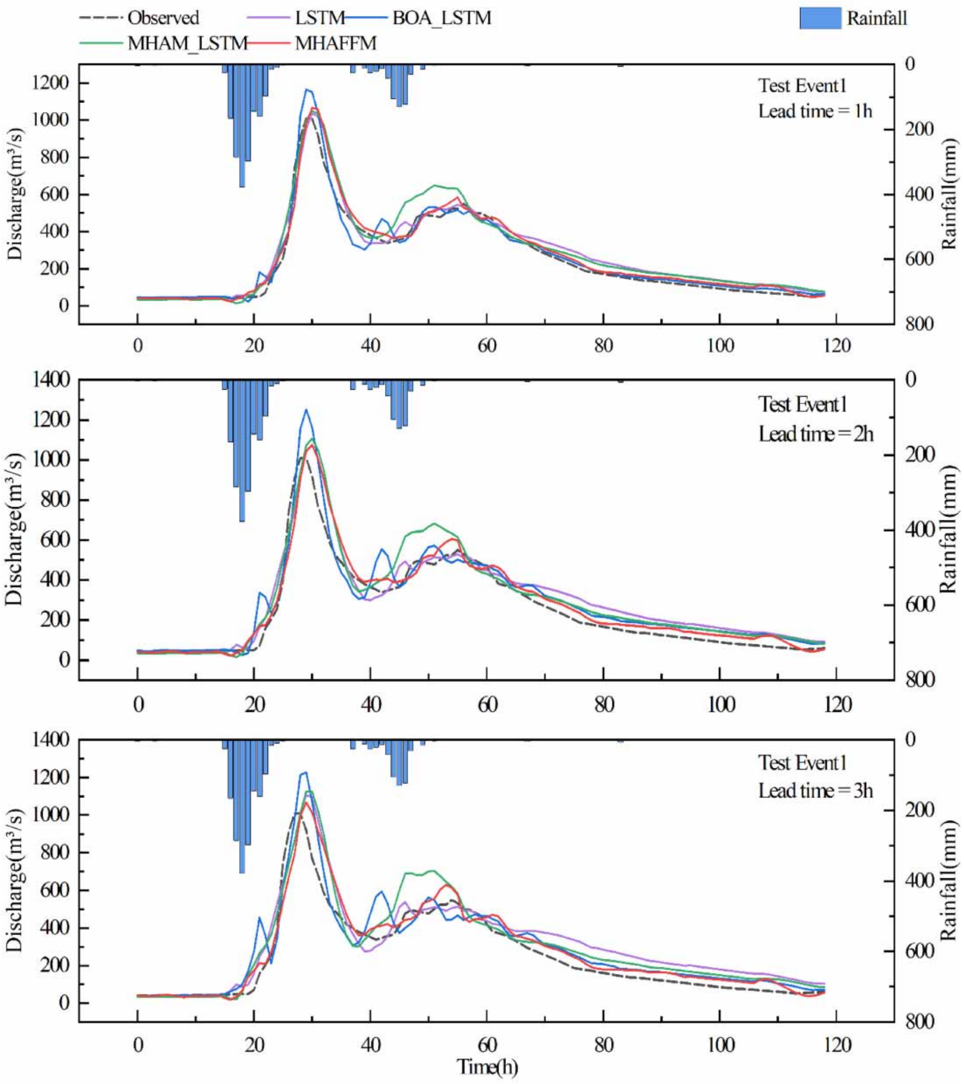


Figure 20 | Flood process curves predicted by MHAFFM, MHAM-LSTM, BOA-LSTM, and LSTM against observed flood process curves under different lead times for test event 1.

4.5. Model execution speed

Model development is based on the torch framework in Python 3.8. The computations are performed using NVIDIA GeForce RTX3080 and Intel Core i7-11800H CPUs. The average run time for each model is listed in Table 8.

In terms of model computation speed, the LSTM model is the fastest, followed by the MHAFFM model, while the MHAM-LSTM model takes slightly longer due to its coupled structure. On the other hand, the BOA-LSTM model requires hyperparameter optimization, resulting in a much longer computation time compared to the other three models.

5. ANALYSIS AND DISCUSSION

5.1. Comparative analysis with LSTM

Based on the results, it can be seen that compared to the LSTM model with the same parameters, the MHAFFM model achieves a significant improvement in both average performance and stability of model evaluation indicators with a small increase in time cost. At the same time, the MHAFFM model also has strong interpretability.

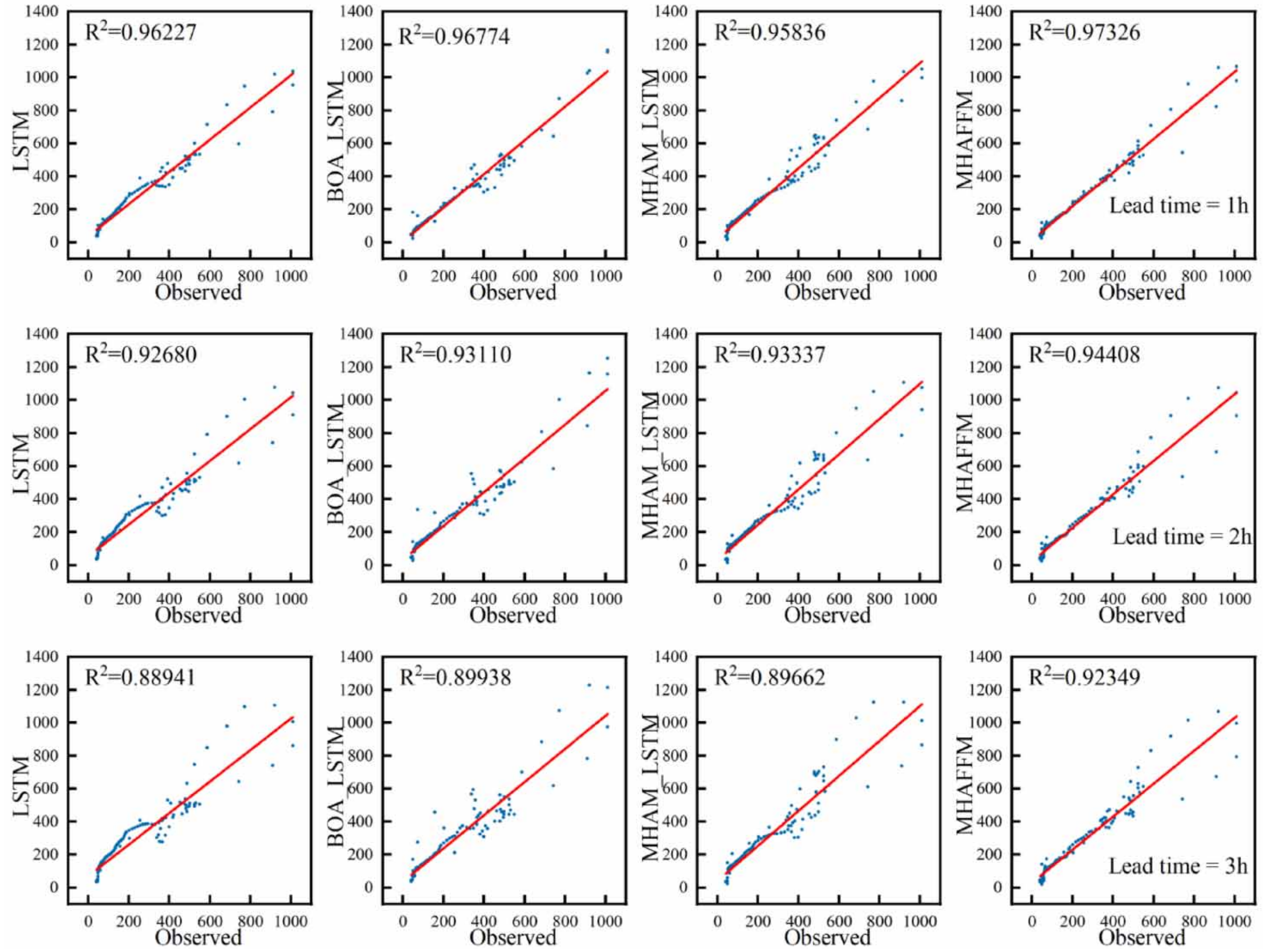


Figure 21 | Correlation analysis of prediction results and observations among LSTM, BOA-LSTM, MHAM-LSTM, and MHAFFM models under different lead times for test event 1.

The analysis is as follows:

(1) In terms of model evaluation metrics, the combination of linear layers and multi-head attention mechanism in the MHAFFM model can better map the flood process generation mechanism and fit the observed runoff sequence compared to the LSTM-gated recurrent unit.

(2) In terms of the stability of model prediction results, the screening method of discarding part of the data in an LSTM-gated recurrent unit leads to oscillating prediction results, and the oscillation becomes stronger as the forecast horizon increases. On the other hand, the MHAFFM model has better stability and is not significantly affected by the forecast horizon.

(3) When compared to the current situation where the working mechanism of the LSTM model is difficult to explain (Li *et al.* 2022a), the multi-head attention mechanism in the MHAFFM model enhances the interpretability of the working mechanism.

5.2. Comparative analysis with BOA-LSTM

Using the Bayesian optimization algorithm, important hyperparameters of the LSTM model are optimized, resulting in a significant improvement in the model's performance for flood forecasting at the expense of increased computation time. However, for the MHAFFM model, the BOA-LSTM model does not show a significant performance advantage, and its prediction performance gradually decreases compared to the MHAFFM model as the forecast horizon increases. In addition, the Bayesian optimization algorithm only improves the performance of the LSTM model in terms of evaluation metrics, while the

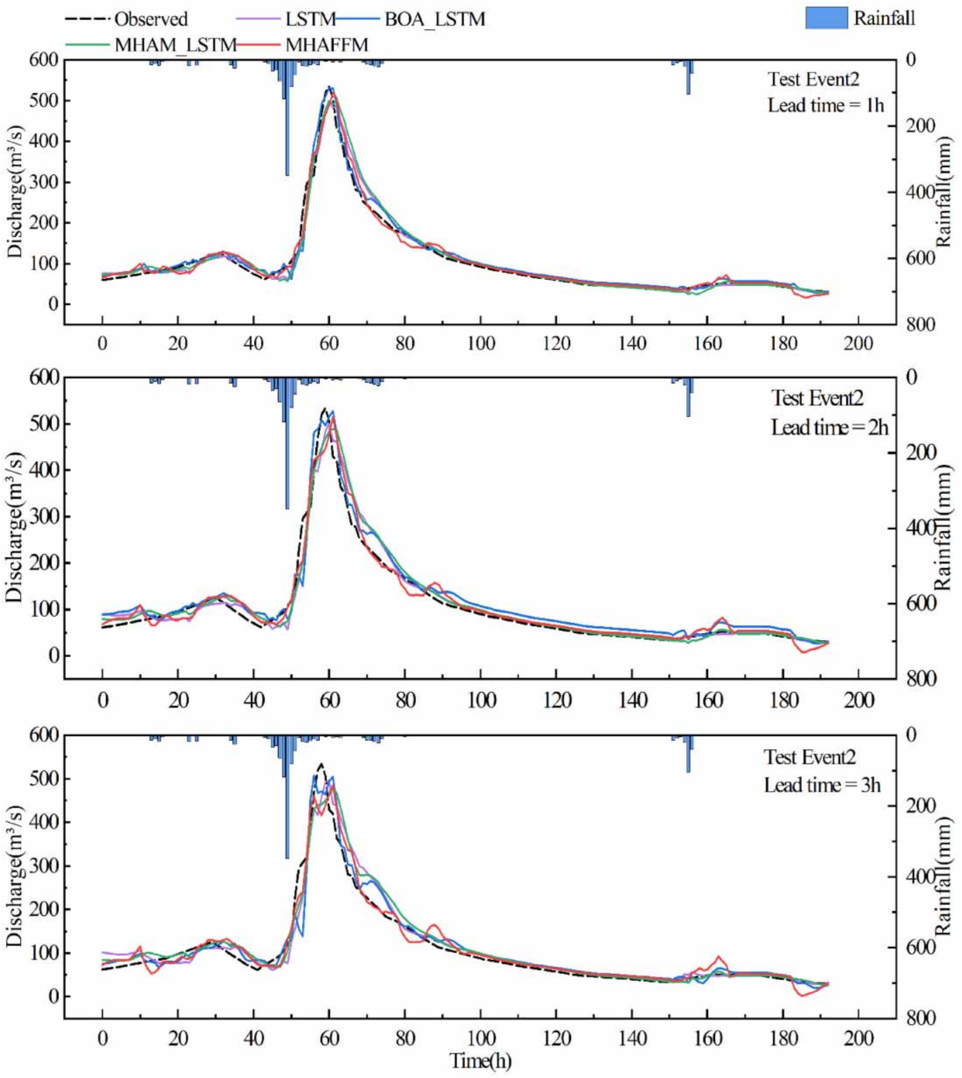


Figure 22 | Flood process curves predicted by MHAFFM, MHAM-LSTM, BOA-LSTM, and LSTM against observed flood process curves under different lead times for test event 2.

stability of the model does not change significantly. Prediction results still exhibit obvious oscillations, indicating that the hyperparameter settings have little impact on the model's stability.

When compared to the BOA-LSTM model that obtains the optimal hyperparameter combination for the current data, the MHAFFM model achieves superior prediction results with less time cost. This indicates that the combination of linear layers and multi-head attention mechanism in the MHAFFM model exhibits structural advantages over the LSTM model for the given data. The performance of the MHAFFM model is less affected by the performance degradation caused by increasing lead time, resulting in more accurate and stable predictions.

5.3. Comparison analysis with MHAM-LSTM

When compared to the MHAM-LSTM model, the MHAFFM model differs only in the data processing layer. However, in terms of model performance, stability, and computation time, the MHAFFM model outperforms the MHAM-LSTM model in all aspects. This suggests that the data processing layer of the two models has a significant impact on the performance of the subsequent multi-head attention mechanism.

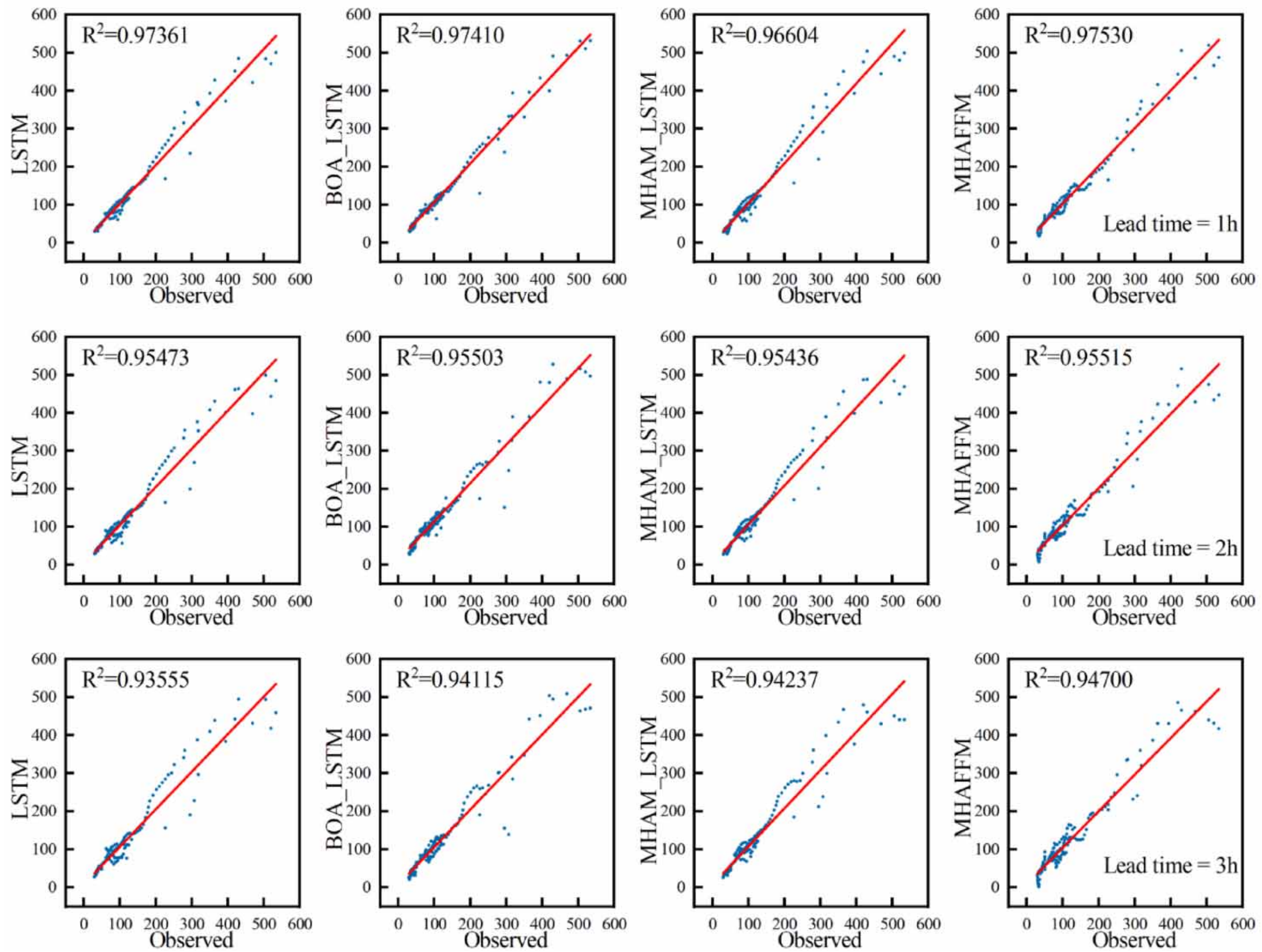


Figure 23 | Correlation analysis of prediction results and observations among LSTM, BOA-LSTM, MHAM-LSTM, and MHAFFM models under different lead times for test event 2.

The MHAFFM model adopts a linear layer to fully pass the data, which allows the multi-head attention mechanism to better explore the linear relationship between input information and flood processes, thereby obtaining stable and excellent prediction results. In contrast, the MHAM-LSTM model selectively passes data through gate units in the hidden layer, making it difficult for the multi-head attention mechanism to effectively work, resulting in poor performance and stability in flood forecasting. In addition, the MHAFFM model has a significantly faster operating speed than the MHAM-LSTM model. Therefore, when compared to the LSTM hidden layer, the linear layer activates the multi-head attention mechanism more concisely and efficiently, achieving more excellent prediction results.

6. CONCLUSION AND FUTURE PERSPECTIVES

This article introduces a multi-head attention mechanism to construct a flood forecasting model and explores the impact of the data processing method in the hidden layer on the performance of the multi-head attention mechanism. Ultimately, the LSTM architecture is abandoned and a linear layer is used as the hidden layer combined with the multi-head attention mechanism to propose the MHAFFM model. Through comparison with the LSTM model, BOA-LSTM model, and MHAM-LSTM model, the following conclusions are drawn:

(1) When compared to LSTM, BOA-LSTM, and MHAM-LSTM models, the proposed MHAFFM model exhibits less performance degradation with increasing lead time. It efficiently accomplishes flood forecasting tasks under different lead times, resulting in high prediction accuracy and good stability.

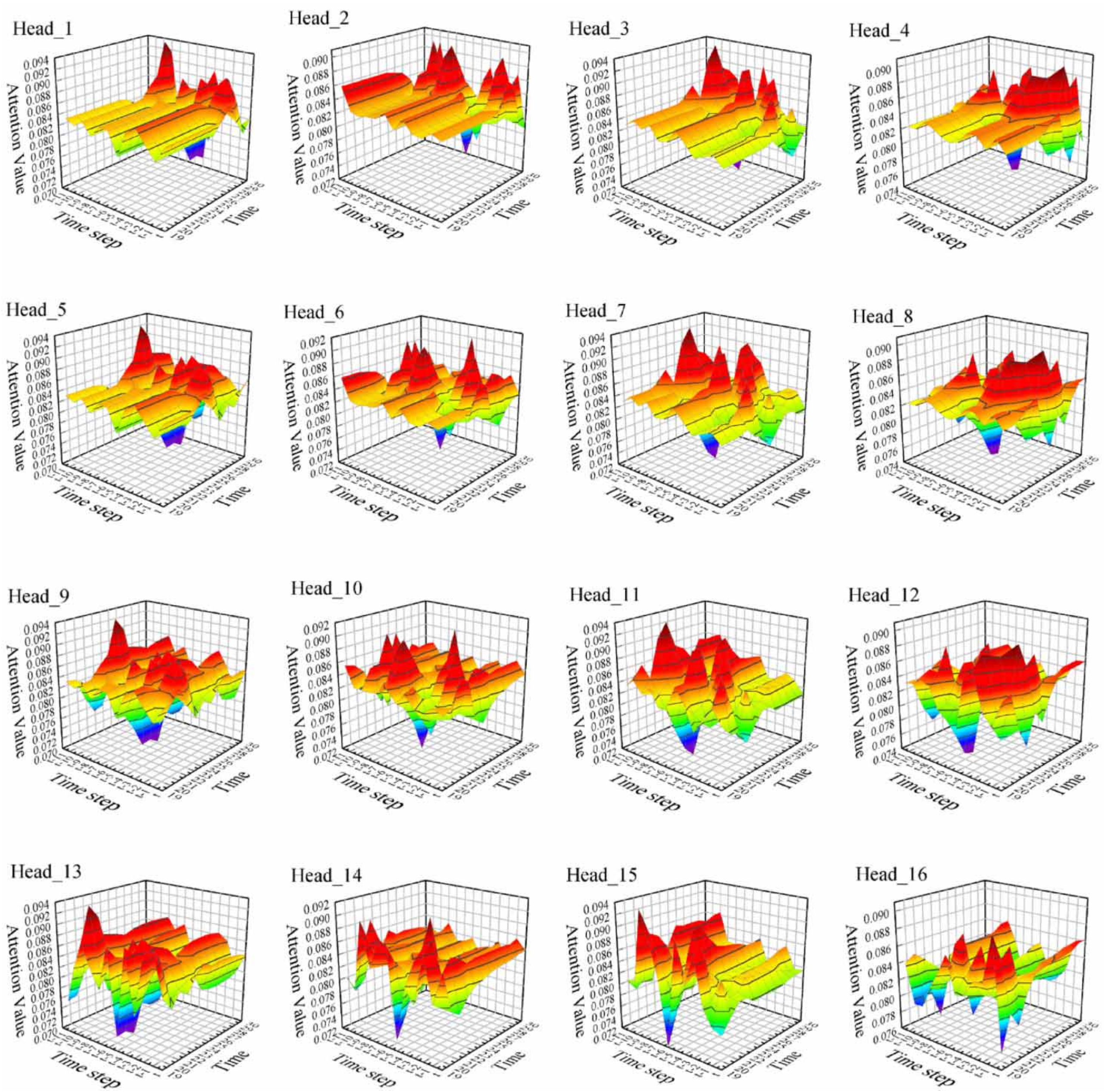


Figure 24 | Visualization of attention in MHAFFM model with 16 heads.

(2) The multi-head attention mechanism enables the model to obtain differentiated attention effects, which not only endows the model with interpretability but also enhances the model's ability to process high-dimensional data information.

(3) In the task of streamflow forecasting, the method of linear layer fully batched data input is more conducive to the performance of the multi-head attention mechanism compared to the LSTM hidden layer. This approach achieves higher prediction accuracy and stability and improves its reliability.

However, this study still has some limitations. First, the study area is characterized by a monsoon climate, and whether the conclusions are applicable under other climate conditions requires further analysis. Second, the forecast period is limited to

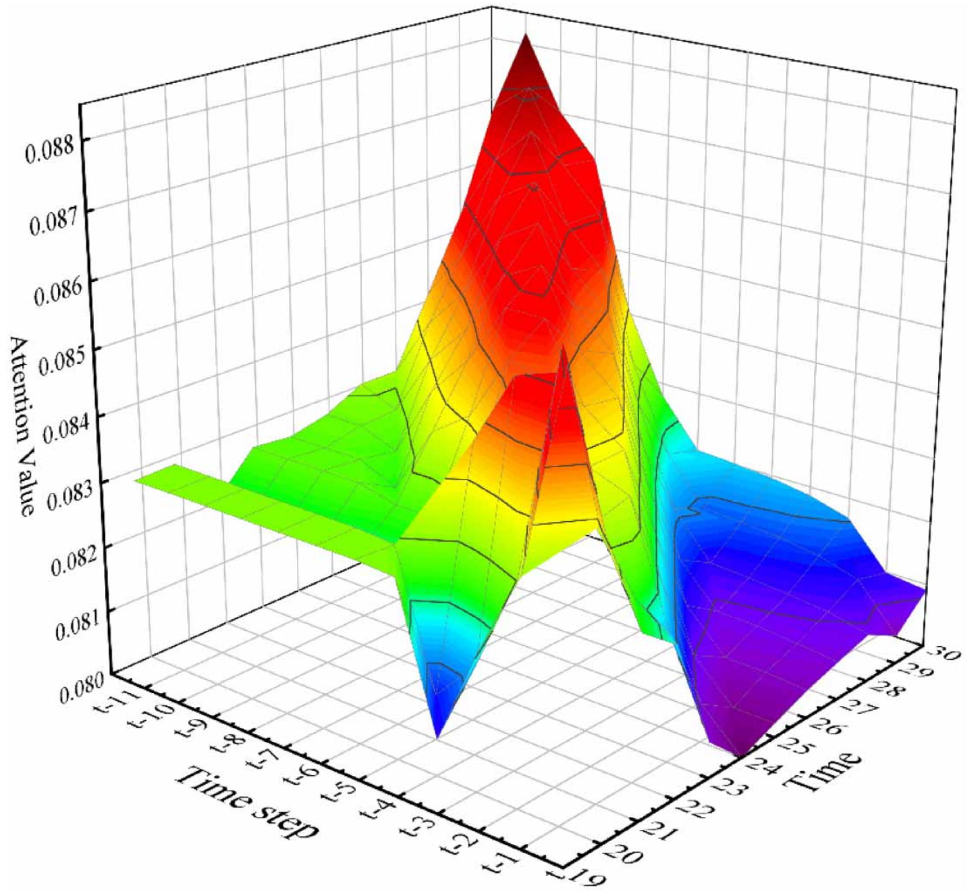


Figure 25 | MHAFFM model dimensionality reduction attention visualization.

Table 8 | Average computation time for each model

Model name	LSTM	BOA-LSTM	MHAM-LSTM	MHAFFM
Time cost (s)	57.94	1,266.98	98.07	60.59

3 h, and whether the superior performance of the MHAFFM model can be maintained beyond this duration needs further investigation. In addition, the hyperparameters of the MHAFFM model were not algorithmically optimized, and the potential improvement in model performance in this direction remains to be explored.

ACKNOWLEDGEMENTS

The authors are grateful for the support of the special project for collaborative innovation of science and technology in 2021 (No: 202121206) and Henan Province University Scientific and Technological Innovation Team (No: 18IRTSTHN009).

DATA AVAILABILITY STATEMENT

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

CONFLICT OF INTEREST

The authors declare there is no conflict.

REFERENCES

Abbas, A., Baek, S., Kim, M., Ligaray, M., Ribolzi, O., Silvera, N., Min, J.-H., Boithias, L. & Cho, K. H. 2020 Surface and sub-surface flow estimation at high temporal resolution using deep neural networks. *Journal of Hydrology* **590**, 125370. https://doi.org/10.1016/j.jhydrol.2020.125370.

Adnan, R. M., Petroselli, A., Heddam, S., Santos, C. A. G. & Kisi, O. 2021 Short term rainfall-runoff modelling using several machine learning methods and a conceptual event-based model. *Stochastic Environmental Research and Risk Assessment* **35** (3), 597–616. https://doi.org/10.1007/s00477-020-01910-0.

Bahdanau, D., Cho, K. & Bengio, Y. 2014 Neural machine translation by jointly learning to align and translate. arXiv:1409.0473. Available from: https://arxiv.org/abs/1409.0473.

Blöschl, G., Bierkens, M. F. P., Chambel, A., Cudennec, C., Destouni, G., Fiori, A., Kirchner, J. W., McDonnell, J. J., Savenije, H. H. G., Sivapalan, M., Stumpp, C., Toth, E., Volpi, E., Carr, G., Lupton, C., Salinas, J., Széles, B., Viglione, A., Aksoy, H., Allen, S. T., Amin, A., Andréassian, V., Arheimer, B., Aryal, S. K., Baker, V., Bardsley, E., Barendrecht, M. H., Bartosova, A., Batelaan, O., Berghuijs, W. R., Beven, K., Blume, T., Bogaard, T., Borges de Amorim, P., Böttcher, M. E., Boulet, G., Breinl, K., Brilly, M., Brocca, L., Buytaert, W., Castellarin, A., Castelletti, A., Chen, X., Chen, Y., Chen, Y., Chifflard, P., Claps, P., Clark, M. P., Collins, A. L., Croke, B., Dathe, A., David, P.C., de Barros, F. P. J., de Rooij, G., Di Baldassarre, G., Driscoll, J. M., Duethmann, D., Dwivedi, R., Eris, E., Farmer, W. H., Feiccabrino, J., Ferguson, G., Ferrari, E., Ferraris, S., Fersch, B., Finger, D., Foglia, L., Fowler, K., Gartsman, B., Gascoin, S., Gaume, E., Gelfan, A., Geris, J., Gharari, S., Gleeson, T., Glendell, M., Gonzalez Bevacqua, A., González-Dugo, M.P., Grimaldi, S., Gupta, A.B., Guse, B., Han, D., Hannah, D., Harpold, A., Haun, S., Heal, K., Helfricht, K., Herrnegger, M., Hipsey, M., Hlaváčiková, H., Hohmann, C., Holko, L., Hopkinson, C., Hrachowitz, M., Illangasekare, T.H., Inam, A., Innocente, C., Istanbulluoglu, E., Jarihani, B., Kalantari, Z., Kalvans, A., Khanal, S., Khatami, S., Kiesel, J., Kirkby, M., Knoben, W., Kochanek, K., Kohnová, S., Kolechkina, A., Krause, S., Kreamer, D., Kreibich, H., Kunstmann, H., Lange, H., Liberato, M. L. R., Lindquist, E., Link, T., Liu, J., Loucks, D. P., Luce, C., Mahé, G., Makarieva, O., Malard, J., Mashtayeva, S., Maskey, S., Mas-Pla, J., Mavrova-Guirguinova, M., Mazzoleni, M., Mernild, S., Misstear, B. D., Montanari, A., Müller-Thomy, H., Nabizadeh, A., Nardi, F., Neale, C., Nesterova, N., Nurtaev, B., Odongo, V. O., Panda, S., Pande, S., Pang, Z., Papacharalampous, G., Perrin, C., Pfister, L., Pimentel, R., Polo, M. J., Post, D., Prieto Sierra, C., Ramos, M.-H., Renner, M., Reynolds, J.E., Ridolfi, E., Rigon, R., Riva, M., Robertson, D. E., Rosso, R., Roy, T., Sá, J. H. M., Salvadori, G., Sandells, M., Schaefli, B., Schumann, A., Scolobig, A., Seibert, J., Servat, E., Shafiei, M., Sharma, A., Sidibe, M., Sidle, R. C., Skaugen, T., Smith, H., Spiessl, S. M., Stein, L., Steinsland, I., Strasser, U., Su, B., Szolgay, J., Tarboton, D., Tauro, F., Thirel, G., Tian, F., Tong, R., Tussupova, K., Tyralis, H., Uijlenhoet, R., van Beek, R., van der Ent, R. J., van der Ploeg, M., Van Loon, A. F., van Meerveld, I., van Nooijen, R., van Oel, P.R., Vidal, J.-P., von Freyberg, J., Vorogushyn, S., Wachniew, P., Wade, A. J., Ward, P., Westerberg, I.K., White, C., Wood, E. F., Woods, R., Xu, Z., Yilmaz, K. K. & Zhang, Y. 2019 Twenty-three unsolved problems in hydrology (UPH) – A community perspective. *Hydrological Sciences Journal* **64** (10), 1141–1158. https://doi.org/10.1080/02626667.2019.1620507.

Cai, H., Liu, S., Shi, H., Zhou, Z., Jiang, S. & Babovic, V. 2022 Toward improved lumped groundwater level predictions at catchment scale: Mutual integration of water balance mechanism and deep learning method. *Journal of Hydrology* **613**, 128495. https://doi.org/10.1016/j.jhydrol.2022.128495.

Ćalasan, M., Abdel Aleem, S. H. E. & Zobaa, A. F. 2020 On the root mean square error (RMSE) calculation for parameter estimation of photovoltaic models: A novel exact analytical solution based on Lambert W function. *Energy Conversion and Management* **210**, 112716. https://doi.org/10.1016/j.enconman.2020.112716.

Cao, Q., Zhang, H., Zhu, F., Hao, Z. & Yuan, F. 2022 Multi-step-ahead flood forecasting using an improved BiLSTM-S2S model. *Journal of Flood Risk Management* **15** (4), e12827. https://doi.org/10.1111/jfr3.12827.

Carreau, J. & Guinot, V. 2021 A PCA spatial pattern based artificial neural network downscaling model for urban flood hazard assessment. *Advances in Water Resources* **147**, 103821. https://doi.org/10.1016/j.advwatres.2020.103821.

Chadalawada, J. & Babovic, V. 2017 Review and comparison of performance indices for automatic model induction. *Journal of Hydroinformatics* **21** (1), 13–31. https://doi.org/10.2166/hydro.2017.078.

Chadalawada, J., Herath, H. M. V. V. & Babovic, V. 2020 Hydrologically informed machine learning for rainfall-runoff modeling: A genetic programming-based toolkit for automatic model induction. *Water Resources Research* **56** (4), e2019WR026933. https://doi.org/10.1029/2019WR026933.

Chen, X., Huang, J., Han, Z., Gao, H., Liu, M., Li, Z., Liu, X., Li, Q., Qi, H. & Huang, Y. 2020 The importance of short lag-time in the runoff forecasting model based on long short-term memory. *Journal of Hydrology* **589**, 125359. https://doi.org/10.1016/j.jhydrol.2020.125359.

Chomba, I. C., Banda, K. E., Winsemius, H. C., Eunice, M., Sichingabula, H. M. & Nyambe, I. A. 2022 Integrated hydrologic-hydrodynamic inundation modeling in a groundwater dependent tropical floodplain. *Journal of Human, Earth, and Future* **3** (2), 237–246. http://dx.doi.org/10.28991/HEF-2022-03-02-09.

Dean, J., Corrado, G. S., Monga, R., Chen, K., Devin, M., Le, Q. V., Mao, M. Z., Ranzato, M. A., Senior, A., Tucker, P., Yang, K. & Ng, A. Y. 2012 Large scale distributed deep networks. In: *Advances in Neural Information Processing Systems*. Curran Associates Inc., Lake Tahoe, Nevada, pp. 1223–1231. Available from: https://dl.acm.org/doi/10.5555/2999134.2999271.

Ditthakit, P., Pinthong, S., Salaeh, N., Weekaew, J., Thanh Tran, T. & Bao Pham, Q. 2023 Comparative study of machine learning methods and GR2M model for monthly runoff prediction. *Ain Shams Engineering Journal* **14** (4), 101941. https://doi.org/10.1016/j.asej.2022.101941.

Ekwueme, B. 2022 Machine learning based prediction of urban flood susceptibility from selected rivers in a tropical catchment area. *Civil Engineering Journal* **8**, 1857–1878. https://doi.org/10.28991/CEJ-2022-08-09-08.

Elman, J. L. 1990 Finding structure in time. *Cognitive Science* **14** (2), 179–211. https://doi.org/10.1207/s15516709cog1402_1.

Frame, J. M., Kratzert, F., Klotz, D., Gauch, M., Shalev, G., Gilon, O., Qualls, L. M., Gupta, H. V. & Nearing, G. S. 2022 Deep learning rainfall–runoff predictions of extreme events. *Hydrology and Earth System Sciences* **26** (13), 3377–3392. https://doi.org/10.5194/hess-26-3377-2022.

Gao, S., Huang, Y., Zhang, S., Han, J., Wang, G., Zhang, M. & Lin, Q. 2020 Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation. *Journal of Hydrology* **589**, 125188. https://doi.org/10.1016/j.jhydrol.2020.125188.

Gao, S., Zhang, S., Huang, Y., Han, J., Luo, H., Zhang, Y. & Wang, G. 2022 A new seq2seq architecture for hourly runoff prediction using historical rainfall and runoff as input. *Journal of Hydrology* **612**, 128099. https://doi.org/10.1016/j.jhydrol.2022.128099.

Herath, H. M. V. V., Chadalawada, J. & Babovic, V. 2021 Hydrologically informed machine learning for rainfall–runoff modelling: Towards distributed modelling. *Hydrology and Earth System Sciences* **25**, 4373–4401. https://doi.org/10.5194/hess-25-4373-2021.

Hochreiter, S. & Schmidhuber, J. 1997 Long short-term memory. *Neural Computation* **9** (8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Hu, C., Wu, Q., Li, H., Jian, S., Li, N. & Lou, Z. 2018 Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. *Water* **10** (11), 1543. https://doi.org/10.3390/w10111543.

Hwang, J. W., Park, R. H. & Park, H. M. 2021 Efficient audio-visual speech enhancement using deep U-Net with early fusion of audio and video information and RNN attention blocks. *IEEE Access* **9**, 137584–137598. https://doi.org/10.1109/ACCESS.2021.3118211.

Jäpel, R. C. & Buyel, J. F. 2022 Bayesian optimization using multiple directional objective functions allows the rapid inverse fitting of parameters for chromatography simulations. *Journal of Chromatography A* **1679**, 463408. https://doi.org/10.1016/j.chroma.2022.463408.

Jiang, S., Zheng, Y., Wang, C. & Babovic, V. 2022 Uncovering flooding mechanisms across the contiguous United States through interpretive deep learning on representative catchments. *Water Resources Research* **58** (1), e2021WR030185. https://doi.org/10.1029/2021WR030185.

Kingma, D. P. & Ba, J. 2014 Adam: a method for stochastic optimization. CoRR, abs/1412.6980. Available from: https://arxiv.org/abs/1412.6980.

Krisnayanti, D., Rozari, P., Garu, V., Damayanti, A., Legono, D. & Nurdin, H. 2022 Analysis of flood discharge due to impact of tropical cyclone. *Civil Engineering Journal* **8**, 1752–1763. https://doi.org/10.28991/CEJ-2022-08-09-01.

Kumar, P. S., Praveen, T. V. & Prasad, M. A. 2016 Artificial neural network model for rainfall-runoff–A case study. *International Journal of Hybrid Information Technology* **9** (3), 263–272. https://doi.org/10.14257/ijhit.2016.9.3.24.

Li, G., Li, F., Xu, C. & Fang, X. 2022a A spatial-temporal layer-wise relevance propagation method for improving interpretability and prediction accuracy of LSTM building energy prediction. *Energy and Buildings* **271**, 112317. https://doi.org/10.1016/j.enbuild.2022.112317.

Li, Y., Zhao, L., Zhang, Z., Li, J., Hou, L., Liu, J. & Wang, Y. 2022b Research on the hydrological variation law of the Dawen River, a tributary of the Lower Yellow River. *Agronomy* **12** (7), 1719. https://doi.org/10.3390/agronomy12071719.

Liang, Z., Li, Y., Hu, Y., Li, B. & Wang, J. 2018 A data-driven SVR model for long-term runoff prediction and uncertainty analysis based on the Bayesian framework. *Theoretical and Applied Climatology* **133** (1), 137–149. https://doi.org/10.1007/s00704-017-2186-6.

Liu, S., Zang, Z., Wang, W. & Wu, Y. 2019 Spatial-temporal evolution of urban heat Island in Xi'an from 2006 to 2016. *Physics and Chemistry of the Earth* **110**, 185–194. https://doi.org/10.1016/j.pce.2018.11.007.

Mao, R., Cao, C., Qian, J. J. Y., Wang, J. & Liu, Y. 2022 Mixture of Gaussian processes based on Bayesian optimization. *Journal of Sensors* **2022**, 7646554. https://doi.org/10.1155/2022/7646554.

Miao, C., Gou, J., Fu, B., Tang, Q., Duan, Q., Chen, Z., Lei, H., Chen, J., Guo, J., Borthwick, A. G. L., Ding, W., Duan, X., Li, Y., Kong, D., Guo, X. & Wu, J. 2022 High-quality reconstruction of China's natural streamflow. *Science Bulletin* **67** (5), 547–556. https://doi.org/10.1016/j.scib.2021.09.022.

Min, X., Hao, B., Sheng, Y., Huang, Y. & Qin, J. 2023 Transfer performance of gated recurrent unit model for runoff prediction based on the comprehensive spatiotemporal similarity of catchments. *Journal of Environmental Management* **330**, 117182. https://doi.org/10.1016/j.jenvman.2022.117182.

Moreno-Pino, F., Olmos, P. M. & Artés-Rodríguez, A. 2023 Deep autoregressive models with spectral attention. *Pattern Recognition* **133**, 109014. https://doi.org/10.1016/j.patcog.2022.109014.

Noh, S.-H. 2021 Analysis of gradient vanishing of RNNs and performance comparison. *Information* **12** (11). 442. https://doi.org/10.3390/info12110442.

Paudel, D., de Wit, A., Boogaard, H., Marcos, D., Osinga, S. & Athanasiadis, I. N. 2023 Interpretability of deep learning models for crop yield forecasting. *Computers and Electronics in Agriculture* **206**, 107663. https://doi.org/10.1016/j.compag.2023.107663.

Pelikan, M., 2005 Bayesian Optimization Algorithm. In: *Hierarchical Bayesian Optimization Algorithm: Toward a new Generation of Evolutionary Algorithms* (Pelikan, M., ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 31–48. https://doi.org/10.1007/978-3-540-32373-0_3.

Sarraf, A. P. 2015 Flood outlier detection using PCA and effect of how to deal with them in regional flood frequency analysis via L-moment method. *Water Resources* **42** (4), 448–459. https://doi.org/10.1134/S0097807815040132.

Tan, Q.-F., Lei, X.-H., Wang, X., Wang, H., Wen, X., Ji, Y. & Kang, A.-Q. 2018 An adaptive middle and long-term runoff forecast model using EEMD-ANN hybrid approach. *Journal of Hydrology* **567**, 767–780. https://doi.org/10.1016/j.jhydrol.2018.01.015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. 2017 Attention is all you need. arXiv e-prints: arXiv:1706.03762. Available from: https://arxiv.org/abs/1706.03762

Wang, H., Gao, X., Qian, L. & Yu, S. 2012 Uncertainty analysis of hydrological processes based on ARMA-GARCH model. *Science China Technological Sciences* **55** (8), 2321–2331. https://doi.org/10.1007/s11431-012-4909-3.

Wang, W.-C., Zhao, Y.-W., Chau, K.-W., Xu, D.-M. & Liu, C.-J. 2021 Improved flood forecasting using geomorphic unit hydrograph based on spatially distributed velocity field. *Journal of Hydroinformatics* **23** (4), 724–739. https://doi.org/10.2166/hydro.2021.135.

Xie, Y., Sun, W., Ren, M., Chen, S., Huang, Z. & Pan, X. 2023 Stacking ensemble learning models for daily runoff prediction using 1D and 2D CNNs. *Expert Systems with Applications* **217**, 119469. https://doi.org/10.1016/j.eswa.2022.119469.

Xu, B., Chen, B., Wan, J., Liu, H. & Jin, L. 2019 Target-aware recurrent attentional network for radar HRRP target recognition. *Signal Processing* **155**, 268–280. https://doi.org/10.1016/j.sigpro.2018.09.041.

Yuan, X., Wang, J., He, D., Lu, Y., Sun, J., Li, Y., Guo, Z., Zhang, K. & Li, F. 2022 Influence of cascade reservoir operation in the Upper Mekong River on the general hydrological regime: A combined data-driven modeling approach. *Journal of Environmental Management* **324**, 116339. https://doi.org/10.1016/j.jenvman.2022.116339.

Zhang, X., Peng, Y., Zhang, C. & Wang, B. 2015 Are hybrid models integrated with data preprocessing techniques suitable for monthly streamflow forecasting? Some experiment evidences. *Journal of Hydrology* **530**, 137–152. https://doi.org/10.1016/j.jhydrol.2015.09.047.

Zhang, X., Wang, H., Peng, A., Wang, W., Li, B. & Huang, X. 2020 Quantifying the uncertainties in data-driven models for reservoir inflow prediction. *Water Resources Management* **34** (4), 1479–1493. https://doi.org/10.1007/s11269-020-02514-7.

Zhao, L., Wang, J., Hu, Y. & Cheng, L. 2020 Conjoint feature representation of GO and protein sequence for PPI prediction based on an inception RNN attention network. *Molecular Therapy – Nucleic Acids* **22**, 198–208. https://doi.org/10.1016/j.omtn.2020.08.025.

Zhao, H., Chen, Z., Shu, X., Shen, J., Lei, Z. & Zhang, Y. 2023 State of health estimation for lithium-ion batteries based on hybrid attention and deep learning. *Reliability Engineering & System Safety* **232**, 109066. https://doi.org/10.1016/j.ress.2022.109066.

Zuo, G., Luo, J., Wang, N., Lian, Y. & He, X. 2020 Decomposition ensemble model based on variational mode decomposition and long short-term memory for streamflow forecasting. *Journal of Hydrology* **585**, 124776. https://doi.org/10.1016/j.jhydrol.2020.124776.

First received 24 June 2023; accepted in revised form 18 October 2023. Available online 30 October 2023