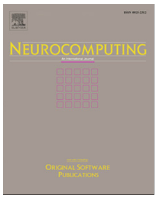
Neurocomputing 403 (2020) 348–359

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Interpretable spatio-temporal attention LSTM model for flood forecasting

Yukai Ding [a], Yuelong Zhu [a], Jun Feng [a,*], Pengcheng Zhang [a], Zirun Cheng [b]

[a] *College of Computer and Information, Hohai University, Nanjing, PR China*
[b] *School of Foreign Languages, Tongji University, Shanghai, PR China*

ARTICLE INFO

Article history:
Received 13 February 2020
Revised 25 April 2020
Accepted 28 April 2020
Available online 8 May 2020
Communicated by W.K. Wong

Keywords:
Attention mechanism
LSTM
Flood forecasting
Interpretable model

ABSTRACT

Modeling interpretable artificial intelligence (AI) for flood forecasting represents a serious challenge: both accuracy and interpretability are indispensable. Because of the uncertainty and nonlinearity of flood, existing hydrological solutions always achieve low prediction robustness while machine learning (ML) approaches neglect the physical interpretability of models. In this paper, we focus on the need for flood forecasting and propose an interpretable Spatio-Temporal Attention Long Short Term Memory model (STA-LSTM) based on LSTM and attention mechanism. We use dynamic attention mechanism and LSTM to build model, Max-Min method to normalize data, variable control method to select hyperparameters, and Adam algorithm to train the model. Emphasis is placed on the visualization and interpretation of attention weights. Experiment results on three small and medium basins in China suggest that the proposed STA-LSTM model outperforms Historical Average (HA), Fully Connected Network (FCN), Convolutional Neural Networks (CNN), Graph Convolutional Networks (GCN), original LSTM (LSTM), spatial attention LSTM (SA-LSTM), and temporal attention LSTM (TA-LSTM) in most cases. Visualization and interpretation of spatial and temporal attention weights reflect the reasonability of the proposed attention-based model.

1. Introduction

As one of the most widespread natural disasters, flood can be destructive and poses a great threat to society. Practical and effective flood forecasting methods are conducive to the rapid response of disaster prevention and relief, protecting lives and property, and upholding stability of society. However, technologies and methods for flood forecasting, especially for small and medium basins, are imperfect. Researchers around the world have made great efforts to lessen the damage caused by floods. At the beginning of the research on flood forecasting, hydrology experts mainly established models of the river basin with hydrodynamics, underlying surface analysis, and other hydrological theories. Despite the good performance of hydrological models, researchers find it still difficult to calibrate models and the portability of the model is poor. With abundant hydrological infrastructure and data, various data analysis techniques, such as signal processing methods wavelet transform [1] and probabilistic analysis methods Bayesian theory [2–4], have been used to analyze hydrological time series. Machine learning and neural network theories, including SVM [5,6] and neural network [7,8], also developed rapidly.

Flood forecasting models can be typically divided into two categories: hydrological models [9–14] and data-driven intelligent models [15,3,5,16,8]. Hydrologic modeling methods usually analyze hydrological features and describe runoff confluence physically. Based on the hydrodynamics theory, researchers usually derive confluence equations by combining the physical laws of mass, momentum, and energy conservation. Despite different applications of hydrological models (e.g., conceptual models, distributed models), parameters of hydrological models have certain physical significance. Calibration for models can be performed according to observation or data analysis, which puts forward high requirements for researchers.

Data-driven intelligent models mainly analyze existing observation data to build input-output mapping relations and predict specific hydrological quantities. With developing computing power and algorithms, hundreds of data analysis and mining technologies have marched rapidly. These technologies include not only regression analysis, EM algorithm, Bayesian theory [17,16], and other classical probability and statistical methods, but also SVM [5,6] and other classical machine learning methods. Novel

[*] Corresponding author.
E-mail address: fengjun@hhu.edu.cn (J. Feng).

https://doi.org/10.1016/j.neucom.2020.04.110
0925-2312/© 2020 Elsevier B.V. All rights reserved.

recurrent neural networks (RNN) [18,15], convolutional neural networks(CNN) [19], graph convolutional networks (GCN) [20], and other artificial intelligence methods [8,21,7,22] are also growing fast. Under the existing data conditions, the results of these studies are satisfactory. And with the further development of technology, many other fields' research results are of great significance for hydrological prediction. Despite the progress made by state-of-the-art approaches on intelligent flood forecasting, deep neural network (DNN) methods still suffer from the following two limitations:

1) Traditional DNN models, such as original LSTM, do not sufficiently and properly handle the 3D spatial and temporal information inherent in hydrological data.
2) As far as we know, existing data-driven (especially attention-based) intelligent flood forecasting models have not yet been proposed with physical interpretation.

In this paper, we propose a data-driven intelligent flood forecasting model based on spatio-temporal attention and LSTM. Hourly rainfall and flow data from three small and medium basins in China, i.e., Tunxi basin, Changhua basin, and Heihe basin, are used to train and evaluate the proposed model respectively. A series of testing experiments are conducted to realize a better setting of hyperparameter for models. We compare and analyze the accuracy of each model according to five evaluation metrics most used in hydrology and statistics fields. The results suggest the proposed STA-LSTM model outperforms baselines in most. We also preliminarily analyze the spatial and temporal attention weights for interpreting the physical reasonability of our model. The main contributions of this paper are summarized as follows:

1) An interpretable data-driven flood forecasting model is proposed based on LSTM and spatio-temporal attention mechanism.
2) Visualization and preliminary interpretation of spatial and temporal attention weights are presented with hydrological progress.
3) Hyperparameter selection and comparative experiments are conducted to optimize and evaluate the proposed STA-LSTM model, respectively.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the main concepts used in this paper. Section 4 introduces details of the STA-LSTM model. The experimental evaluation of STA-LSTM is presented in Section 5. Finally, Section 6 gives conclusions and suggestions for future work.

2. Related work

This part briefly reviews relevant researches inspired us to design the STA-LSTM model, mainly including flood forecasting models, LSTM-based models, and attention-based models.

2.1. Flood forecasting

Hydrological Models: Hydrological models can be divided into hydro-physical models and hydro-mathematical modes. Both of them mainly consider the physical process of a flood, i.e., runoff generation and confluence. In China, the most famous hydro-mathematical model is Xinanjiang model [9]. In addition, Paquet et al. [10] proposed a semi-continuous rainfall-runoff simulation method for extreme flood estimation. Kabir et al. [12] used a process-based distributed modeling approach to estimate the sediment budget at a river basin scale. And Nam et al. [14] predicted short-term flood inundation prediction using hydrological-hydraulic models forced with downscaled rainfall from global numeric weather prediction.

Data-Driven Models: Data-driven models for flood forecasting or time series analysis have developed for a long time accompanied by machine learning, neural network, and other technologies. Srivastava et al. [23] successfully predicted the inflow of Tarbela reservoir by using regression and neural network fusion model. Cheng et al. [24] proposed an artificial neural network model based on the quantum particle swarm optimization algorithm to predict the daily flow of reservoirs. Ma et al. [25] proposed a hybrid SVM-BP model and evaluated it on the Changhua river basin data. The experiments indicated the combination of SVM and BP model is beneficial to flood prediction. Yan et al. [18] performed small watershed streamflow forecasting based on LSTM, which achieved satisfactory results in river flow predicting and provided a new method for flood forecasting in a small watershed. Recently, Wu et al. [26] proposed a context-aware LSTM network model, which uses the context attention module in each step of LSTM to achieve a high prediction accuracy.

2.2. LSTM-based methods

Long Short Term Memory (LSTM) is a modified version of recurrent neural networks, which is proposed to solve the problem of long-distance (time) dependence by Hochreiter and Schmidhuber [27]. Liu et al. [28] proposed an LSTM network model integrating global situational awareness and attention to realize human 3D motion recognition. Yang et al. [21] used the LSTM model to realize traffic flow with feature enhancement. The proposed LSTM-based model performed well in the experiments. Further, Li et al. [22] combined LSTM and attention mechanism to predict stock price. The results suggested that attention-based LSTM can make a better prediction than baselines.

2.3. Attention-based methods

The attention mechanism in deep learning is similar to the human selective visual attention mechanism, which uses limit attention to select more critical information from numerous input features. Attention has been widely applied in various tasks, such as machine translation, image caption, and video motion recognition. Song et al. [29] proposed an end-to-end spatio-temporal attention model to realize the recognition and prediction of human actions in the video. Chen et al. [30] proposed a model of spatial and channel attention and image labeling combined with convolutional neural networks, which performed well in the experimental dataset. Zhai et al. [31] combined channel attention mechanism and dilated convolutional neural networks to address problems in optical flow estimation. And they also tackled that problem in [19] with a dual self-attention pyramid network. All experimental results showed that the attention mechanism is beneficial for optical flow estimation. Ran et al. [32] proposed an attention-based LSTM model to predict travel time. The experimental results showed that the attention-based models can achieve better accuracy than the baselines.

Inspired by the above models, we propose a spatio-temporal attention LSTM model for flood forecasting. In the next part, some preliminaries and the principle of the model will be introduced in detail.

3. Preliminaries

3.1. Long short term memory methods

Since the basic RNN usually retains the information of the last several moments, it can hardly handle the long-distance-dependence task. To tackle that problem, LSTM adapts basic RNN with three gates and preserve more useful information of input. As shown in Fig. 1, main structure of LSTM network includes:

Input gate. The input gate generates i_t based on the current input x_t and previous hidden layer state h_{t-1}. The input coefficient i_t determines how much information from x_t can be used to calculate cell state c_t.

Forget gate. Forget coefficient f_t is produced by forget gate and determines how much c_{t-1} is kept in c_t.

Output gate. $h_t = o_t \tanh(c_t)$ indicates how the output coefficient o_t controls final output h_t of the network.

These gates of LSTM help capture long-term as well as short-term dependencies of input time series data and prevent the gradient diminishing or exploding of information transmission. The key of LSTM to realize long-term memory lies in keeping the input information of each step in the memory unit. The hidden layer state of each output contains all input information before the current moment. Since the hidden layer state is usually represented by a vector of a certain length, the network gradually compresses all the information as time goes by. However, such indiscriminate compression will weaken the difference in time between input features to some extent and may fail to highlight important information in historical information. Hence, proper improvement is needed to enhance the discrimination of LSTM.

3.2. Attention mechanism

The most shared attention framework is demonstrated in Fig. 2. The attention mechanism is usually used to optimize sequence-handling models. The preprocessed input sequence is processed by a neural network or by mapping function to generate raw attention weights. Then with the softmaxed attention weights and raw input value, we can calculate the final output. According to the value of attention weights, attention models can be divided into two types: hard attention and soft attention. Hard attention refers to the one-hot selection of the input data features, which means attention weight can only be 0 or 1. The soft attention refers to that the weight is between 0 and 1, and the weight selection range is more flexible.

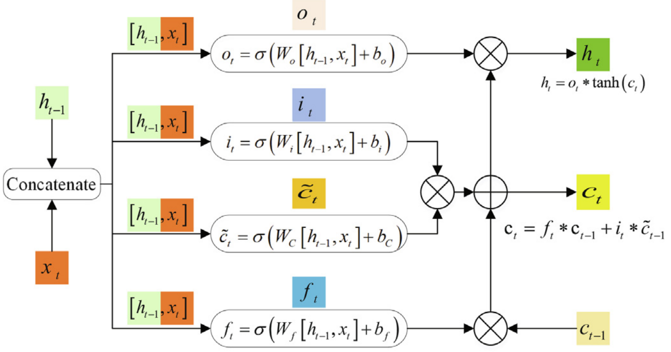

o_t
$[h_{t-1}, x_t]$
$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$
h_t
$h_t = o_t * \tanh(c_t)$
h_{t-1}
i_t
$[h_{t-1}, x_t]$
$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$
Concatenate
$\tilde{c}_t$
c_t
$[h_{t-1}, x_t]$
$\tilde{c}_t = \sigma(W_C[h_{t-1}, x_t] + b_C)$
$c_t = f_t * c_{t-1} + i_t * \tilde{c}_{t-1}$
x_t
f_t
$[h_{t-1}, x_t]$
$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$
c_{t-1}

Fig. 1. Illustration of LSTM structure.

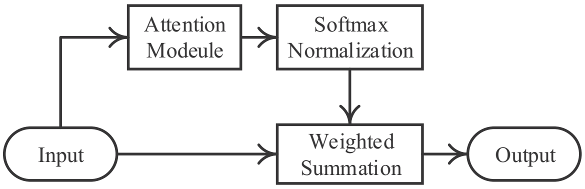


Fig. 2. Illustration of attention mechanism.

4. Methodology

To make full use of spatio-temporal information of input, we modify the original LSTM with attention mechanism. Firstly, spatial weights (or feature weights) are dynamically assigned to input features in a single time step. Afterward, the hidden layer state of each step of LSTM is fully utilized to dispatch temporal attention weights to the hidden layer state of each time step. The overall structure of the model is shown in Fig. 3. We take rainfall and streamflow as input features, and the output of our model is the prediction of the next n-time-step streamflow. Spatial and temporal attention weights affect the inputs and the outputs of LSTM cells. With the help of spatial attention module and temporal attention module, we can dynamically adjust the attention weights as well as improve the performance of LSTM cell. We adopt Adam algorithm [33] to train models. The spatial attention operation and temporal attention operation are demonstrated with details subsequently.

4.1. Spatial attention operation

Suppose there is a 2D spatio-temporal feature matrix $X \in R^{m \times k}$, in which m represents the number of features in a single time step and k is the number of time steps. The hydrology features usually include rainfall and flow of multiple stations. As shown in Fig. 3, the input feature matrix X can be divided into k m-dimension vectors like Eq. (1).

$$x_t = \left[f^t_1, f^t_2, \ldots, f^t_m\right]_{m \times 1} \tag{1}$$

$$\alpha_t = SA(x_t) = \left[\alpha^t_1, \alpha^t_2, \ldots, \alpha^t_m\right]^T_{1 \times m} \tag{2}$$

$$x'_t = \alpha_t \odot x_t = \left[\alpha^t_1 f^t_1, \alpha^t_2 f^t_2, \ldots, \alpha^t_m f^t_m\right]_{m \times 1} \tag{3}$$

Fig. 4 shows the calculation process of spatial dynamic attention weights. After the calculation of monolayer neurons, the input feature vector is activated by $Sigmoid(x) = \frac{1}{1+e^{-x}}$. And then with the normalization of $Softmax(x_i) = \frac{e^{-x_i}}{\sum_{i=1}^{n} e^{-x_i}}$, we can generate the spatial attention weight α_t (See Eq. (2)). Softmax is usually used for normalization to ensure the limited additivity of weights. Note that $\odot$ in Eq. (3) is Hadamard product, i.e., element-wise product operation.

4.2. Temporal attention operation

The spatial-attention-weighted sequence data is poured into LSTM cell step by step. The hidden layer state output sequence is obtained successively as Eq. (4) shown.

$$H = [h_1, h_2, \ldots, h_k]_{k \times s} \tag{4}$$

$$\beta = TA(H) = [\beta_1, \beta_2, \ldots, \beta_k]_{1 \times k} \tag{5}$$

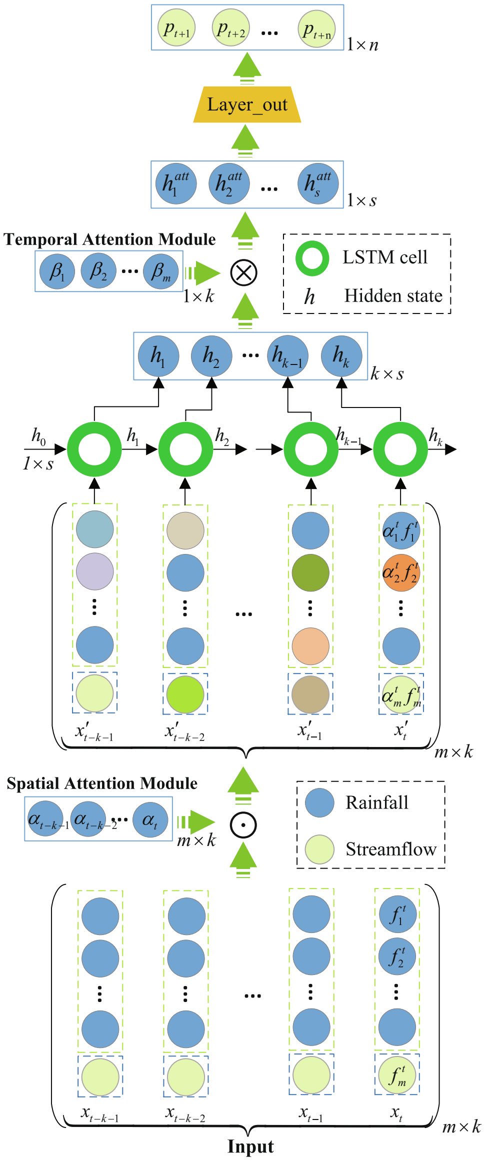

p_{t+1} p_{t+2} ... p_{t+n}
$1\times n$
Layer_out
h_1^{att} h_2^{att} ... h_s^{att}
$1\times s$
Temporal Attention Module
β_1 β_2 ... β_m
$1\times k$
⊗
LSTM cell
h Hidden state
h_1 h_2 ... h_{k-1} h_k
$k\times s$
h_0
$1\times s$
h_1 h_2 h_{k-1} h_k
$\alpha_1^t f_1^t$
$\alpha_2^t f_2^t$
$\alpha_m^t f_m^t$
x'_{t-k-1} x'_{t-k-2} x'_{t-1} x'_t
$m\times k$
Spatial Attention Module
α_{t-k-1} α_{t-k-2} ... α_t
$m\times k$
⊙
Rainfall
Streamflow
f_1^t f_2^t f_m^t
x_{t-k-1} x_{t-k-2} x_{t-1} x_t
$m\times k$
Input

Fig. 3. Framework of the proposed Spatio-Temporal Attention LSTM model. Spatial and temporal attention modules weigh the inputs and the outputs of LSTM cells. With the help of spatial attention module and temporal attention module, we can dynamically adjust the attention weights as well as improve the accuracy of LSTM cell.

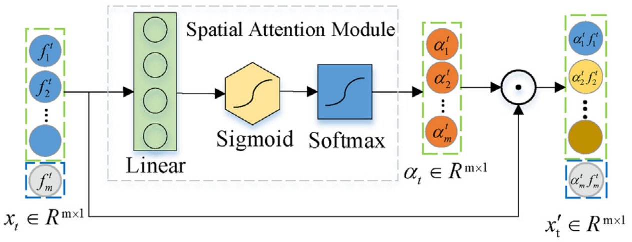


Fig. 4. Illustration of the spatial attention operation.

$$h_{att} = \beta \otimes H = \sum_{i=1}^{k} \beta_i h_i, h_{att} \in R^{1\times s} \quad (6)$$

$$p = O(h_{att}), p \in R^{1\times n} \quad (7)$$

Afterward, the temporal attention weight β is generated after ReLU activation ($ReLU(x) = max(0, x)$) and Softmax normalization as suggested in Fig. 5 and Eq. (5). ⊗ in Eq. (6) denotes matrix product. Output layer generates final prediction p without activation as shown in Eq. (7).

4.3. Interpretability

We introduce previously the overall model architecture as well as the principle of the spatial and temporal module. In this part, we will propose a validation scheme for the interpretation of model attention weights.

Fig. 6 shows diagram of a certain basin. Usually, rainfall influences the downstream flow with delay(confluence time). In a small basin, the average confluence time can be used to replace the confluence time of the whole basin to simplify the calculation. In theory, our spatio-temporal attention weight will also change with the change of input rainfall and flow data. For certain basins, the change law of temporal attention weight should conform to that of the confluence process.

$$q(x_d, y_d, t+1) \propto q(x_d, y_d, t-i)|_{i=0}^{k-1} \& p(x, y, t-\Delta t - j)|_{j=0}^{k-\Delta t} \quad (8)$$

The $t+1$ downstream flow $q(x_d, y_d, t+1)$ relates to $q(x_d, y_d, t-i)|_{i=0}^{k-1}$ and $p(x, y, t-\Delta t - j)|_{j=0}^{k-\Delta t}$ as presented in Eq. (8). The focus of temporal attention should translate over time while the Δt is related to the confluence time t_c as shown in Fig. 7. We conduct experiment to verify the interpretability of our STA-LSTM model. Details will be presented later in the experiment section.

5. Experiments

This part intends to show implementation details and results of our experiments. Datasets and Data Preprocessing introduce the used three datasets and preprocessing. Baselines and Implementation Setting show details of baselines and model structures and parameters. Evaluation Metrics lists five common evaluation metrics. Results and Discussions give experiment results and our discussions.

5.1. Datasets

We use data of Tunxi basin, Changhua basin, and Heihe basin in our experiments. The hydrological features include hourly rainfall data from all hydrological stations and hourly flow data from outlet stations in those basins. Please check more details of the datasets in Table 1. The input covers 12 h (including current) rainfall and streamflow of each basin. The output is the outlet flow in the next 6 h (each hour).

Tunxi is a 49532-samples dataset covers 12 rainfall stations and 1 flow station, as shown in Fig. 8(a). Tunxi basin has a catchment area of 2696.76 km^2. The period of the dataset is from June 27, 1981 to March 18, 2007.

Changhua contains 9354 samples from 8 rainfall stations and 1 flow station, as shown in Fig. 8(b). The period of the Changhua dataset is from April 7, 1998 to July 20, 2010. Changhua basin locates in Zhejiang province, China and has an area of 3444 km^2.

Heihe has 5423 samples from 10 rainfall stations and 1 flow station, as shown in Fig. 8(c). Heihe basin covers an area of 1350 km^2 and the period is from April 1, 2003 to November 10, 2010.

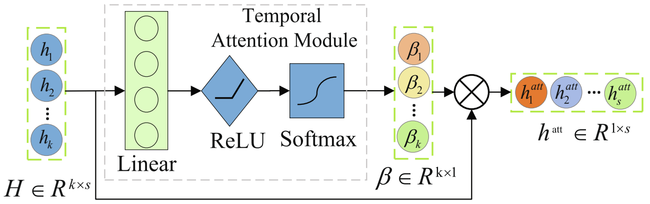

h_1
h_2
h_k
Temporal Attention Module
Linear
ReLU
Softmax
β_1
β_2
β_k
h_1^{att} h_2^{att} ··· h_s^{att}
$h^{att} \in R^{1\times s}$
$H \in R^{k\times s}$
$\beta \in R^{k\times 1}$

Fig. 5. Illustration of temporal attention operation.

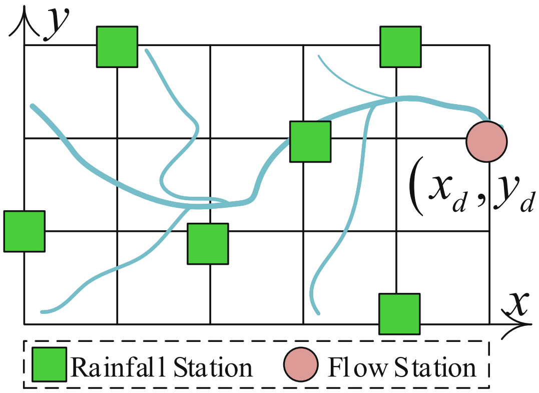


Fig. 6. Basin diagram.

5.2. Data preprocessing

Data Cleaning: We use the average completion to handle the missing or low-level outlier x_t as shown in Eq. (9). And we directly eliminate complex outliers from the dataset.

$$x'_t = \frac{x_{t-1} + x_{t+1}}{2} \tag{9}$$

where x_{t-1} and x_{t+1} are the values of the previous time and the next time.

Feature Extraction: We use all the spatial and temporal features available in the dataset, including the hourly rainfall of each station in the basin and the hourly flow of the outlet flow station.

Max-Min Normalization: Since different input features may have different magnitude, we standardize input features to the range [0,1] with Max-Min normalization as shown in Eq. (10).

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{10}$$

where x represents a value in the sequence of primitive variables, x_{max} and x_{min} represent the maximum and minimum values in variables, respectively.

5.3. Baselines

We compare the proposed STA-LSTM with the following 1 statistical method, 4 non-attention-based neural network models, and 2 attention-based models:

- **HA:** Historical Average. A statistical method for time series analysis. We use an average of 12 past steps to predict the next 1 step with 6 iterations.
- **FCN:** Fully Connected Networks. A simple and robust form of neural network. As shown in Fig. 9, the FCN model mainly consists of three parts, input layer, FCN layer, and output layer.
- **CNN:** Convolutional Neural Networks. As shown in Fig. 9, the CNN model mainly consists of three parts, input layer, CNN layer, and output layer.
- **GCN [20]:** Graph Convolutional Networks. A novel spectral graph convolutions solution with 1st-order Chebyshev approximation. As shown in Fig. 9, the GCN model mainly consists of three parts, input layer, GCN layer, and output layer.
- **LSTM [27]:** Long Short-Term Memory networks, a famous variant of RNN. LSTM in Fig. 9 shows the LSTM model.
- **SA-LSTM:** Spatial Attention LSTM Networks. As shown in Fig. 9, the SA-LSTM model adds a spatial attention module based on the original LSTM model. The main structure and parameters of the model are the same as LSTM.
- **TA-LSTM:** As shown in Fig. 9, the TA-LSTM model adds a temporal attention module on the basis of the original LSTM model. The main structure and parameters of the model are also the same as the LSTM.

5.4. Implementation settings

Parameters Settings: All experiments are implemented on a Linux server (CPU:Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20 GHz, GPU: NVIDIA GeForce TITAN Xp, 12 GB). We look backward 12 timesteps(12 h) and predict 6 timesteps (6 h). For all datasets, 80% of data is used for training, 20% for testing. Before we perform the final experimental evaluations, we pre-train our models to calibrate some hyperparameters. We assume that the learning rate, training times (epoch), and the number of neurons in the hidden layer (hidden dimension) make up the core hyperparameter space of the model. A robust solution can be obtained by searching for the hyperparameter space.

Fig. 10(a)–(c) show the relations between model performance and learning rate, hidden dimension, and epoch. According to the experiment result, we train all models 60 epochs with Adam optimizer to ensure convergence and efficiency. Adam weight decay is set to 1e−4 and random seed is fixed at 7. The initial learning rate is 5e−2 with a decay rate of 0.7 after every 20 epochs and batch size is 200. In addition, kernel size for CNN is set to 1 × 3, out channel number is 3. For GCN, the hidden dimension is set to $k \times m \times 2$ which is more adaptive to different datasets. Table 2 lists details of parameters and structures in the models.

Structures Settings: We explore the impact of activation of attention modules on our STA-LSTM in this part. Fig. 10(d) shows the impact of different combinations of activations.

The spatial attention module is at the shallow end of the model. Shallow features are usually concentrated in a narrow range of values. Hence, the spatial attention weights obtained from Sigmoid are relatively soft and comprehensive (because the weights are always in range (0,1)). Temporal attention operation, on the other hand, deals with the features after LSTM cell. ReLU can increase the differentiation of features and that makes the weights harder. Features less than zero will be compressed and features greater than zero will become more prominent after Softmax.

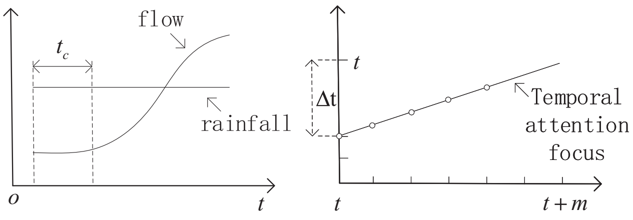


Fig. 7. Illustration of t_c and Δt.

Table 1
Dataset Characteristics.

Datasets	Dataset Size			Mean	Variance	Standard	Median	Kurtosis	Skewness	Fetures	Time
	Total	Training	Testing			Deviation					Resolution
Tunxi	49532	39625	9907	197.59	187150.70	432.60	75.56	40.88	5.60	rainfall, streamflow	1 h
Changhua	9354	7483	1871	146.80	41068.11	202.64	80.50	21.36	3.98	rainfall, streamflow	1 h
Heihe	5423	4338	1085	101.38	15214.98	123.34	59.83	28.15	4.21	rainfall, streamflow	1 h

The experimental results show the performance of the sig-relu model is better on multiple datasets with the most stable performance, followed by the sig-sig model. To keep better robustness of our model, we take the sig-relu combination as our activation setting.

5.5. Evaluation metrics

We adopt Root Mean Square Error (RMSE), Determination Coefficient (R^2), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and forecast Qualification Rate (QR) as evaluation metrics in experiments.

RMSE evaluates the accuracy of regression results, as shown in Eq. (11).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - q_i)^2} \tag{11}$$

where y_i represents the forecast value, q_i denotes the real flow value, and n is the number of test samples.

$\mathbf{R^2}$ indicates the correlation between two random variables, as shown in Eq. (12).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - q_i)^2}{\sum_{i=1}^{n}(q_i - \overline{q})^2} \tag{12}$$

where y_i represents the forecast value, q_i denotes the real value, $\overline{q}$ is the average value of the real value sequence, and n means the number of test samples.

MAE is the average of the absolute value of the sum of the deviations of all individual observations from the arithmetic mean. Eq. (13) is the formula of MAE.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - q_i| \tag{13}$$

MAPE is the mean absolute percentage error of predictions as shown in Eq. (14).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - q_i}{q_i}\right| \times 100\% \tag{14}$$

QR refers to the ratio of qualified predictions (m) to total samples (n) in the test dataset, as shown in Eq. (15). The qualified prediction refers to the prediction whose error is less than 20% of the real value.

$$QR = \frac{m}{n} \times 100\% \tag{15}$$

QR and R^2 are often used to judge the quality of the prediction model. As shown in Table 3, the determination coefficient of the predictions shall not be less than 0.5, and the forecast qualification rate shall not be less than 60%.

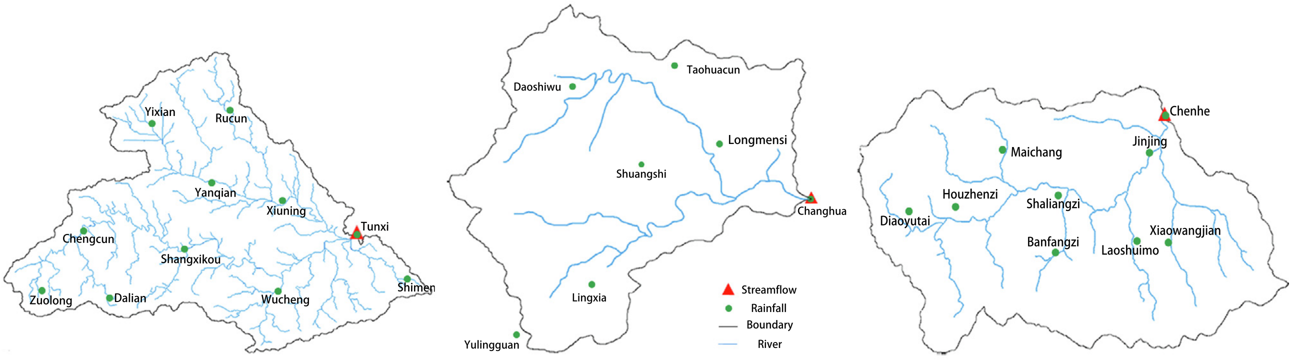
5.6. Results and discussions

The following subsections introduce accuracy comparison, error analysis, time-space consumption, and model interpretation with details respectively. Considering the actual application scenario, we only evaluate models on data with flow above 100. The experiments are conducted with the following four research targets:

1) Evaluating the accuracy of the proposed STA-LSTM model and exploring the benefits of spatio-temporal attention.
2) Analyzing the model error and finding out the direction for improving the accuracy of predictions.
3) Comparing the time and space consumption of models and finding the method of optimization and acceleration of intelligent models.
4) Visualizing spatial and temporal attention weights, and giving a preliminary physical interpretation from a hydrology perspective.

5.6.1. Benefits of spatio-temporal attention

Table 4 list the average performances of models in this 6-h flow prediction task. On all datasets, our STA-LSTM models outperform baselines in most cases. The basic statistical method HA gets the



(a) Tunxi (b) Changhua (c) Heihe

Fig. 8. Station maps of three basins. (a) is Tunxi Basin, (b) is Changhua Basin, and (c) is Heihe Basin.

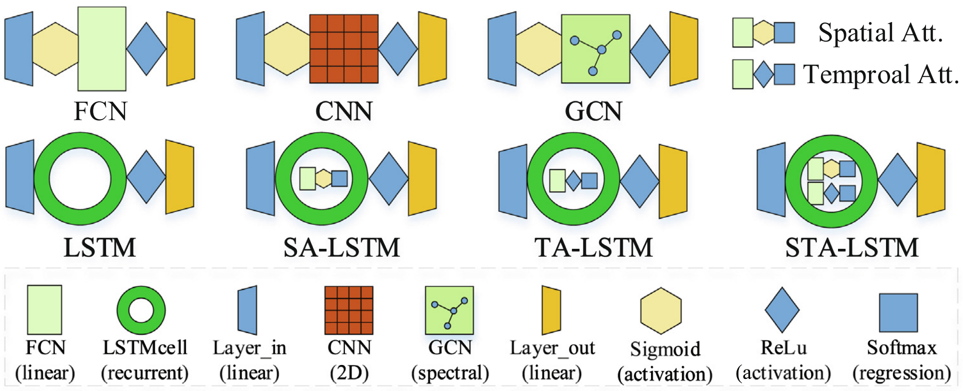


Fig. 9. Structures of models used in experiments.

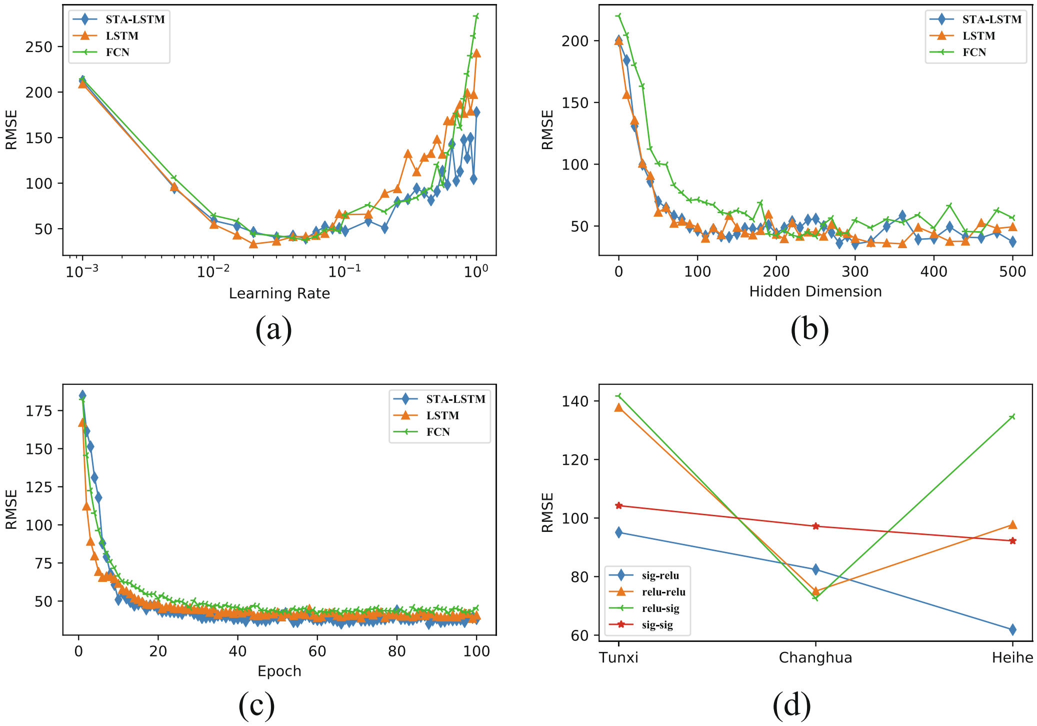


Fig. 10. Results of explore experiments.

Table 2
Structures and Parameters of Models.

Models	Layer_in		Layer_hidden		Layer_out		Parameters		
	dimension	activation	dimension	activation	dimension	activation	learning rate	epoch	hidden dimension
FCN	linear:$(k \times m, k \times m)$	sigmoid	linear:$(k \times m, s)$	relu	linear:(s, n)	–	0.05	60	300
CNN	linear:$(k \times m, k \times m)$	sigmoid	cnn:$(k \times m, s)$	relu	linear:(s, n)	–	0.05	60	$(k-2) \times m \times 3$
GCN	linear:$(k \times m, k \times m)$	sigmoid	gcn:$(k \times m, s)$	relu	linear:(s, n)	–	0.05	60	$k \times m \times 2$
LSTM	linear:$(k \times m, k \times m)$	–	lstm:$(k \times m, m, s)$	relu	linear:(s, n)	–	0.05	60	300
SA-LSTM	linear:$(k \times m, k \times m)$	–	lstm:$(k \times m, m, s)$ linear:(m, m)	relu sigmoid	linear:(s, n)	–	0.05	60	300
TA-LSTM	linear:$(k \times m, k \times m)$	–	lstm:$(k \times m, m, s)$ linear:$(k \times s, k)$	relu relu	linear:(s, n)	–	0.05	60	300
STA-LSTM	linear:$(k \times m, k \times m)$	–	lstm:$(k \times m, m, s)$ linear:(m, m) linear:(k, k)	relu sigmoid relu	linear:(s, n)	–	0.05	60	300

last position on all datasets. The results suggest the neural networks are more adaptive to the uncertainty of flood. The attention-based models are more accurate and robust than the original LSTM model. The attention modules decrease the RMSE and increase R^2 and QR of LSTM models. And in most cases, our proposed STA-LSTM outperforms the novel CNN and GCN models, which also reflects the benefits of the proposed spatial-temporal attention.

The RMSE of the predictions from models in experiments is shown in Fig. 11. On Tunxi dataset, the proposed STA-LSTM model has the best accuracy at all 6 moments. On Changhua and Heihe datasets, the STA-LSTM model has the best accuracy at most moments. At the same time, SA-LSTM and TA-LSTM also performed better, which is better than LSTM model mostly. At $t + 1$, all the models perform well, and errors increase to different degrees with time. Fig. 12 is the R^2 results of different models, which suggests our proposed STA-LSTM model outperform others in most times on all datasets. However, we can also infer that datasets size has a great impact on model accuracy according to Table 3 and Fig. 12. Models trained with Tunxi data have better performance while the average level of models trained by the other two datasets is lower.

As shown in Figs. 13 and 14, the attention-based model, especially the STA-LSTM model, performs well on different datasets. Smaller MAE and MAPE represent the error between the predicted value and the real value of the model is smaller, and the average predictive power of the model is more robust.

In most cases, our proposed STA-LSTM model outperforms other basic models. The performance of the LSTM model with attention module is better than the original LSTM model, which indicates the attention modules have a positive effect on models. With the existing experiment results, we can find the accuracy of the TA-LSTM model is slightly better than SA-LSTM on certain time steps. Considering Table 3 and Fig. 15, the predictions of NN models can reach at least B level on Tunxi dataset. On the Heihe and Changhua river basins, performances of models are less impressive while STA-LSTM can reach almost Level B.

5.6.2. Error analysis

Despite the good accuracy of models, there are prediction errors in testing. The errors also show obvious temporal variation trend. Because of the imperfect forecasting architecture and input information, the model loses accuracy gradually. In this part, we analyze and discuss the performance of models at a different time in different datasets. We can summarize the following conclusions:

1) The prediction error of each model will increase with time. As time goes on, the information provided by the input feature is no longer enough.
2) The causes of complicated floods are difficult to analyze, especially in the small and medium-sized river basins. In this paper, we only select the key rainfall and flow as model input, which may lead to incomplete input information. Part of the error may come from the incompleteness of the input information.
3) MAE or other average metrics cannot fully reflect the predictive performance of the model. Multiple indicators should be taken into account for comprehensive evaluation.
4) Among the models adopted in the experiments, the models based on attention performed well. Specifically, the proposed STA-LSTM model performs best in most cases while the error variation trend is similar to the others.
5) The performance of the model is related to the number of samples in the data set, that is, the training level. Tunxi dataset is the largest one used and the model trained on Tunxi dataset can give out better prediction. A large dataset is needed to optimize the parameters of complex models.

Table 3
Prediction level.

Level A	Level B	Level C
$R^2 \geq 0.90$ $QR \geq 85.0\%$	$0.90 > R^2 \geq 0.70$ $85.0\% > QR \geq 70.0\%$	$0.70 > R^2 \geq 0.50$ $70.0\% > QR \geq 60.0\%$

5.6.3. Time-space consumption

Table 5 lists training time and model size of the models used in experiments. FCN consumes the least time and GCN takes up the least storage space while our proposed STA-LSTM model is in the opposite end. The size of TA-LSTM model is the closest to STA-LSTM model, suggesting the temporal attention module is larger than the spatial attention module. The spatial attention module is small, however, it leads to a large increase in model training time.

Although the time-space consumption of the attention-based model is larger than that of other models (the actual value is still small), the performance of the model is greatly improved. Besides, the temporal and spatial consumption gap between the STA-LSTM and the original LSTM model is not large. In addition, the GCN model has good performance while takes less, which may spark important ideas for our next research.

5.6.4. Model interpretation

In previous subsections, we compare the performances of several models based on experimental details and generally analyze the difference between them. The conclusion can be drawn as attention-based models perform better. And in this section, we visualize the temporal and spatial attention weights of our proposed STA-LSTM model and mainly analyze the time-varying trend of weights.

Fig. 16 is the visualization of spatio-temporal attention weights from our STA-LSTM model. The six subfigures represent the six moments respectively. The input is data from Tunxi dataset. The X-coordinate is the input hydrology feature, and the Y-coordinate is the historical moment. The spatio-temporal attention weights are similar to the weight of time, which is also gradually moving forwards. However, because of the existence of spatial attention weight, the changing trend of the overall weight is relatively slow and not obvious.

Fig. 17 depicts the change of temporal attention weight in three datasets under actual flood input. The X-coordinate is the forecast time, and the Y-coordinate is the historical time. It can be seen that, as the forecast time goes on, the temporal attention weight also goes forwards gradually. The temporal attention weight goes forwards from time $t - 6$, and the advance speed synchronizes with the forecast speed.

With Figs. 17 and 16, we can preliminarily draw the following three conclusions:

1) The temporal attention module focuses on the different past moments at different prediction moments. The increasing trend of temporal attention weight is almost linear and is similar to the progress of confluence.
2) Different initial time delay of different basins may be related to basin area and topography. And this index may indicate the confluence time.
3) The error of spatio-temporal attention weight may lead to the error of prediction results to a certain extent even a large extent. As we find in Fig. 17, the trend of the curve is not strictly increasing. The TX curve is almost strictly increasing except for the last time step. CH and HH curves are much more fluctuant

Table 4
Average performance of models on Tunxi, Changhua and Heihe.

Models	Tunxi					Changhua					Heihe				
	RMSE	R2	MAE	MAPE	QR	RMSE	R2	MAE	MAPE	QR	RMSE	R2	MAE	MAPE	QR
HA	231.65	0.73	79.39	18.00%	81.84%	118.24	0.37	84.77	27.17%	66.25%	94.34	0.29	96.86	34.67%	48.24%
FCN	117.09	0.93	43.46	10.50%	92.99%	87.87	0.65	37.05	15.00%	79.95%	65.77	0.77	31.36	13.83%	85.14%
CNN	101.88	0.95	38.29	9.67%	95.15%	86.41	0.72	41.92	17.00%	80.57%	63.94	0.83	**30.51**	12.67%	93.06%
GCN	99.88	0.96	38.15	9.00%	95.22%	85.99	0.71	37.97	14.00%	85.00%	64.34	0.80	31.27	12.50%	94.37%
LSTM	106.65	0.96	38.31	8.67%	94.40%	88.46	0.73	**37.49**	14.50%	85.28%	61.95	0.77	33.84	13.50%	90.41%
SA-LSTM	102.04	0.95	40.17	8.67%	95.18%	88.31	0.67	37.86	**13.67%**	87.08%	65.23	0.83	35.29	13.50%	91.81%
TA-LSTM	103.23	0.95	40.23	8.17%	96.18%	84.71	0.69	38.8	15.50%	85.86%	62.86	0.79	31.11	12.17%	92.41%
STA-LSTM	**97.03**	**0.96**	**37.49**	**8.00%**	**96.19%**	**82.43**	**0.75**	40.13	14.17%	**87.82%**	**61.87**	**0.84**	33.53	**11.50%**	**94.88%**

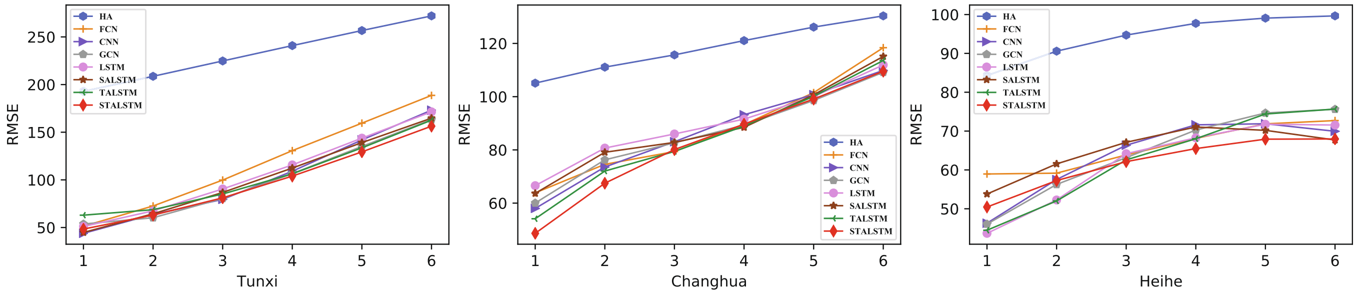


Fig. 11. RMSE comparison of prediction from different models on three datasets.

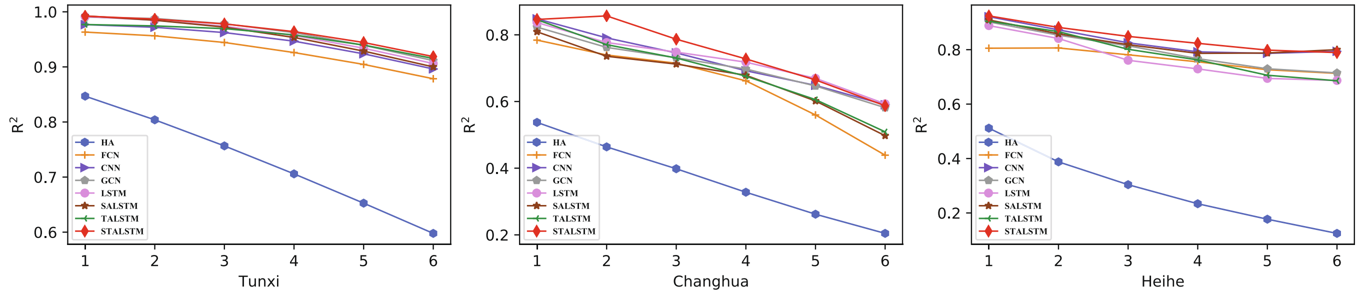


Fig. 12. R^2 comparison of prediction from different models on three datasets.

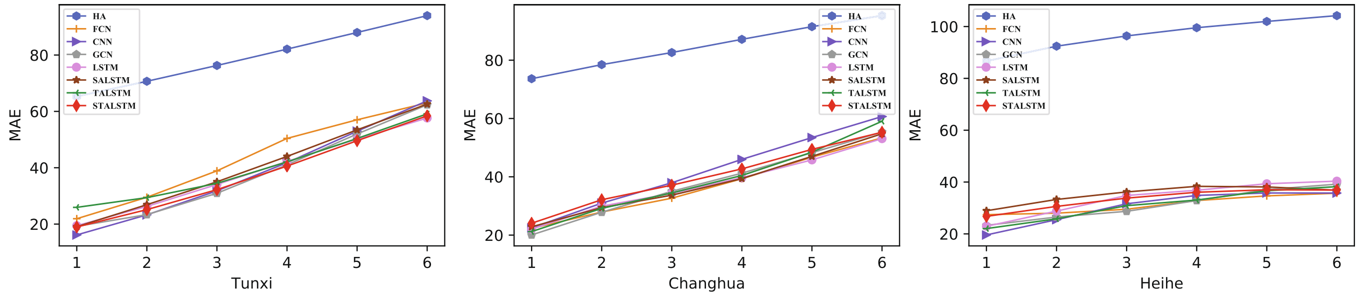


Fig. 13. MAE comparison of prediction from different models on three datasets.

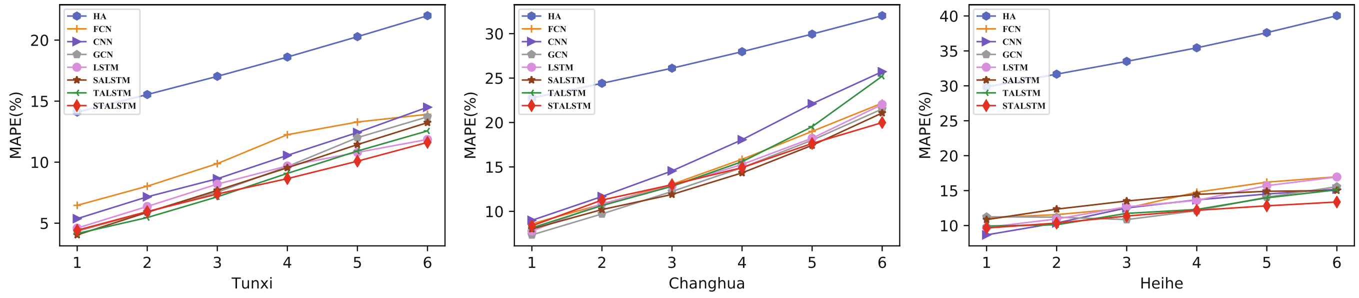


Fig. 14. MAPE comparison of prediction from different models on three datasets.

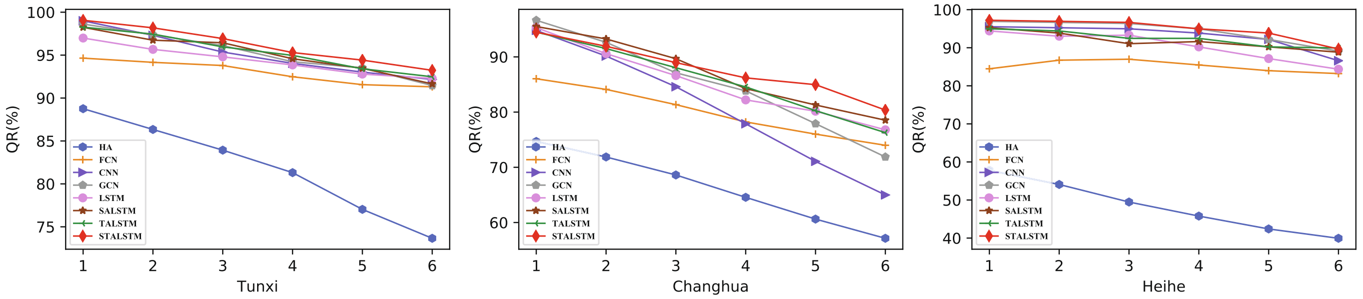


Fig. 15. QR comparison of prediction from different models on three datasets.

Table 5
Time-Space Consumption of Models.

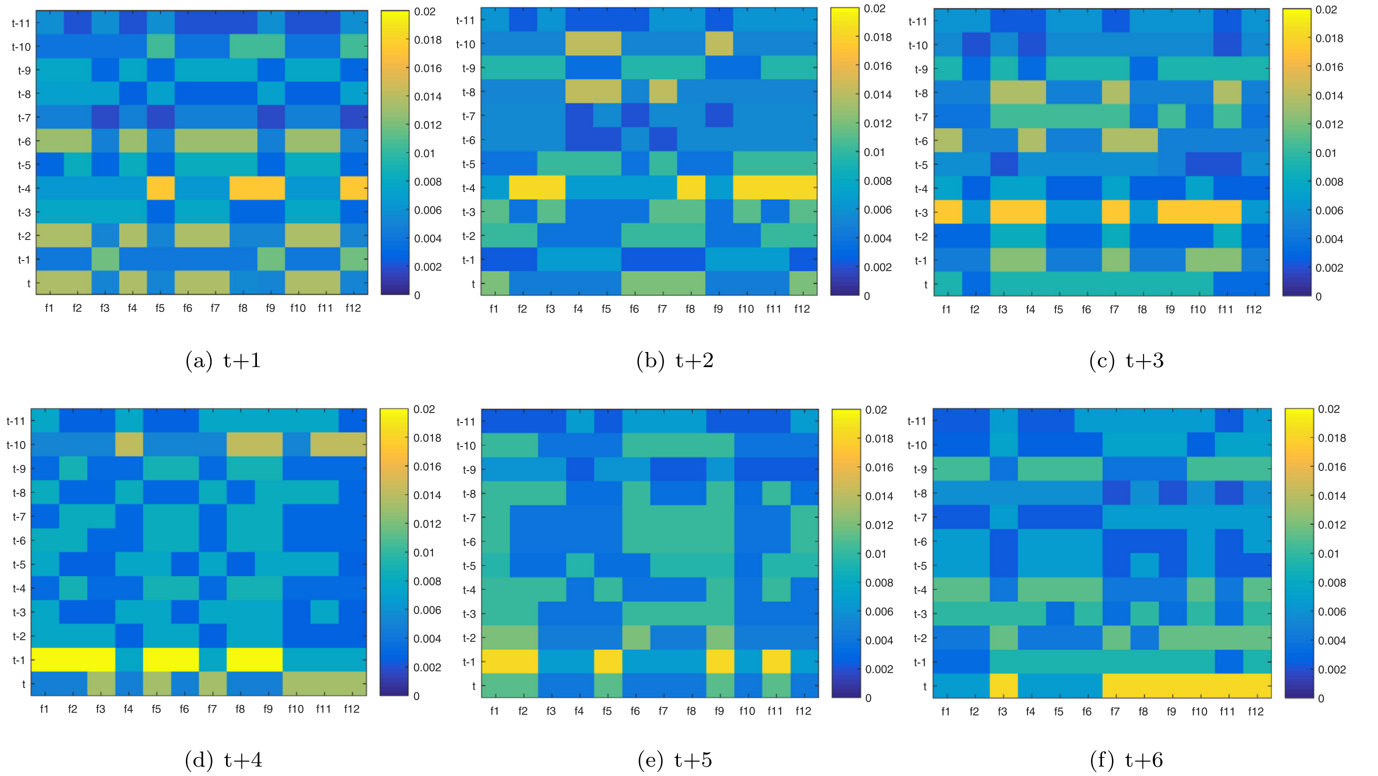
Models	Size (*KB*)	Training (*s*)	Testing (*s*)
HA	2	–	2
FCN	265	48	2
CNN	24	56	2
GCN	12	63	2
LSTM	1572	131	2
SA-LSTM	1574	203	2
TA-LSTM	1743	190	2
STA-LSTM	1744	267	2

epoch = 60, CPU: Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20 GHz, GPU: NVIDIA GeForce TITAN Xp, 12 GB.

which may lead to greater error. Models trained on Tunxi dataset can give out better predictions with the less-error temporal attention module.

6. Conclusions and future work

In this paper, we propose a flood forecasting LSTM model (STA-LSTM) based on the attention mechanism. HA, FCN, CNN, GCN, LSTM, SA-LSTM, TA-LSTM, and STA-LSTM models, are designed and evaluated in the experiment on Tunxi, Changhua, and Heihe datasets. Under the verification of RMSE, R^2, MAE, MAPE, and QR, the attention-based LSTM model used in the experiment perform better than the LSTM model and the FCN model. The STA-LSTM model with both spatial and temporal attention modules performs best among the models used, which reflects the benefits of the proposed spatio-temporal attention. Consequently, we analyze the



(a) t+1

(b) t+2

(c) t+3

(d) t+4

(e) t+5

(f) t+6

Fig. 16. This figure is the visualization of spatio-temporal attention weights of STA-LSTM model on Tunxi dataset. The six subgraphs represent the six moments respectively. The X-coordinate is the input hydrology features, and the Y-coordinate is the historical time. Different colors represent different weights of different sizes, with yellow indicating larger weights and blue indicating smaller weights.

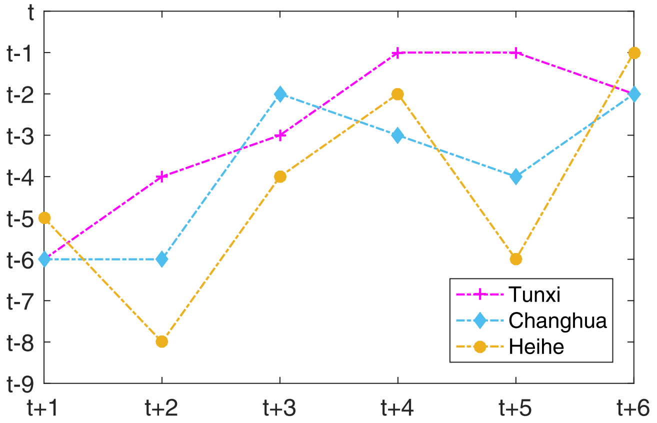


Fig. 17. This figure depicts the change of temporal attention weight in three datasets under actual flood input. The X-coordinate is the forecast time, and the Y-coordinate is the historical time.

temporal variation of the spatio-temporal attention weight and puts forward three inferences for model interpretation. Existing experiment results also suggest dataset size and quality may have a great relationship with the training level.

Inspired by the performance of the GCN model, our next research will consider further improving the performance of the model by utilizing the graph information of basins. Current and future works are aimed at flood data augmentation, flow trend control, and physical interpretation of data-driven models.

CRediT authorship contribution statement

Yukai Ding: Conceptualization, Methodology, Software, Investigation, Writing - original draft, Writing - review & editing. **Yuelong Zhu:** Funding acquisition, Resources, Supervision, Writing - review & editing. **Jun Feng:** Validation, Supervision, Writing - review & editing. **Pengcheng Zhang:** Writing - original draft, Writing - review & editing. **Zirun Cheng:** Writing - original draft, Writing - review & editing.

Acknowledgment

This work is supported by the National Key R&D Program of China (Grant No. 2018YFC0407901).

References

[1] P.K. de Macedo Machado, C.A.G. Freire, G.B.L. da Santos Silva, Analysis of the use of discrete wavelet transforms coupled with ANN for short-term streamflow forecasting, Appl. Soft Comput. 80 (2019) 494–505, https://doi.org/10.1016/j.asoc.2019.04.024.

[2] P.J. Darwen, Bayesian model averaging for river flow prediction, Appl. Intell. 49 (1) (2019) 103–111, https://doi.org/10.1007/s10489-018-1232-0.

[3] Y. Wu, Y. Ding, J. Feng, Sparse bayesian flood forecasting model based on smoteboost, in: 2019 International Conference on IEEE Cyber, Physical and Social Computing (CPSCom), Atlanta, GA, USA, July 14–17, 2019, 2019, pp. 279–284.https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00067.

[4] A. D'Addabbo, A. Refice, G. Pasquariello, F.P. Lovergine, D. Capolongo, S. Manfreda, A bayesian network for flood detection combining SAR imagery and ancillary data, IEEE Trans. Geosci. Remote Sens. 54 (6) (2016) 3612–3625, https://doi.org/10.1109/TGRS.2016.2520487.

[5] S. Li, K. Ma, Z. Jin, Y. Zhu, A new flood forecasting model based on SVM and boosting learning algorithms, in: IEEE Congress on Evolutionary Computation, CEC 2016, Vancouver, BC, Canada, July 24–29, 2016, 2016, pp. 1343–1348. https://doi.org/10.1109/CEC.2016.7743944.

[6] H. Azamathulla, A. Ab Ghani, C.K. Chang, Z. Abu Hasan, N. Zakaria, Machine learning approach to predict sediment load a case study, Clean Soil Air Water 38 (2010) 969–976, https://doi.org/10.1002/clen.201000068.

[7] Y. Tian, K. Zhang, J. Li, X. Lin, B. Yang, Lstm-based traffic flow prediction with missing data, Neurocomputing 318 (2018) 297–305, https://doi.org/10.1016/j.neucom.2018.08.067.

[8] F. Liu, F. Xu, S. Yang, A flood forecasting model based on deep learning algorithm via integrating stacked autoencoders with BP neural network, in: Third IEEE International Conference on Multimedia Big Data, BigMM 2017, Laguna Hills, CA, USA, April 19–21, 2017, 2017, pp. 58–61.https://doi.org/10.1109/BigMM.2017.29.

[9] R. Zhao, The xinanjiang model applied in china, J. Hydrol. 135 (1) (1992) 371–381, https://doi.org/10.1016/0022-1694(92)90096-E, URL: http://www.sciencedirect.com/science/article/pii/002216949290096E.

[10] E. Paquet, F. Garavaglia, R. Garon, J. Gailhard, The schadex method: a semi-continuous rainfallrunoff simulation for extreme flood estimation, J. Hydrol. 495 (15) (2013) 23–37, https://doi.org/10.1016/j.jhydrol.2013.04.045.

[11] M. Rogger, A. Viglione, J. Derx, G. Blschl, Quantifying effects of catchments storage thresholds on step changes in the flood frequency curve, Water Resour. Res. 49 (10) (2013) 6946–6958, https://doi.org/10.1002/wrcr.20553.

[12] M.A. Kabir, D. Dutta, S. Hironaka, Estimating sediment budget at a river basin scale using a process-based distributed modelling approach, Water Resour. Manage. 28 (12) (2014) 4143–4160, https://doi.org/10.1007/s11269-014-0734-8.

[13] N.A. Pierini, E.R. Vivoni, A. Robles-Morua, R.L. Scott, M.A. Nearing, Using observations and a distributed hydrologic model to explore runoff thresholds linked with mesquite encroachment in the sonoran desert, Water Resour. Res. 50 (10) (2015) 8191–8215, https://doi.org/10.1002/2014WR015781.

[14] D.H. Nam, T.M. Dang, K. Udo, A. Mano, Short-term flood inundation prediction using hydrologic-hydraulic models forced with downscaled rainfall from global nwp, Hydrol. Processes 28 (24) (2015) 5844–5859, https://doi.org/10.1002/hyp.10084.

[15] Y. Ding, Y. Zhu, Y. Wu, F. Jun, Z. Cheng, Spatio-temporal attention LSTM model for flood forecasting, in: 2019 International Conference on IEEE Cyber, Physical and Social Computing, CPSCom 2019, Atlanta, GA, USA, July 14–17, 2019, pp. 458–465.https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00095.

[16] S. Han, P. Coulibaly, Bayesian flood forecasting methods: a review, J. Hydrol. 551 (2017) 340–351, https://doi.org/10.1016/j.jhydrol.2017.06.004.

[17] K. Schrter, H. Kreibich, K. Vogel, C. Riggelsen, F. Scherbaum, B. Merz, How useful are complex flood damage models?, Water Resour. Res. 50 (4) (2014) 3378–3395, https://doi.org/10.1002/2013WR014396.

[18] L. Yan, J. Feng, T. Hang, Small watershed stream-flow forecasting based on LSTM, in: Proceedings of the 13th International Conference on Ubiquitous Information Management and Communication, IMCOM 2019, Phuket, Thailand, January 4–6, 2019, 2019, pp. 1006–1014.https://doi.org/10.1007/978-3-030-19063-7_79.

[19] M. Zhai, X. Xiang, R. Zhang, N. Lv, A. El Saddik, Optical flow estimation using dual self-attention pyramid networks, IEEE Trans. Circuits Syst. Video Technol. (2019) 1–1.https://doi.org/10.1109/TCSVT.2019.2943140.

[20] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017.

[21] B. Yang, S. Sun, J. Li, X. Lin, Y. Tian, Traffic flow prediction using LSTM with feature enhancement, Neurocomputing 332 (2019) 320–327, https://doi.org/10.1016/j.neucom.2018.12.016.

[22] H. Li, Y. Shen, Y. Zhu, Stock price prediction using attention-based multi-input LSTM, in: Proceedings of The 10th Asian Conference on Machine Learning, ACML 2018, Beijing, China, November 14-16, 2018, pp. 454–469. URL: http://proceedings.mlr.press/v95/li18c.html.

[23] P.K. Srivastava, D. Han, M.A. Rico-Ramirez, M. Bray, T. Islam, Selection of classification techniques for land use/land cover change investigation, Adv. Space Res. 50 (9) (2012) 1250–1265, https://doi.org/10.1016/j.asr.2012.06.032.

[24] C. Cheng, W. Niu, z. Feng, J. Shen, Daily reservoir runoff forecasting method using artificial neural network based on quantum-behaved particle swarm optimization, Water 7 (8) (2015) 4232–4246, https://doi.org/10.3390/w7084232.

[25] K. Ma, S. Li, J. Wang, Y. Yu, Comparative study of data-driven intelligent flood forecasting methods for small- and medium-sized rivers, J. Univ. Sci. Technol. China 46 (2016) 774–779, https://doi.org/10.3969/j.issn.0253-2778.2016.09.009.

[26] Y. Wu, W. Xu, J. Feng, S. Palaiahnakote, T. Lu, Local and global bayesian network based model for flood prediction, in: 24th International Conference on Pattern Recognition, ICPR 2018, Beijing, China, August 20–24, 2018, pp. 225–230.https://doi.org/10.1109/ICPR.2018.8546257.

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

[28] J. Liu, G. Wang, P. Hu, L. Duan, A. C. Kot, Global context-aware attention LSTM networks for 3d action recognition, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017, 2017, pp. 3671–3680.https://doi.org/10.1109/CVPR.2017.391.

[29] S. Song, C. Lan, J. Xing, W. Zeng, J. Liu, An end-to-end spatio-temporal attention model for human action recognition from skeleton data, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4–9, 2017, San Francisco, California, USA, 2017, pp. 4263–4270. URL: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14437.

[30] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, T. Chua, SCA-CNN: spatial and channel-wise attention in convolutional networks for image captioning, in:

2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017, 2017, pp. 6298–6306.https://doi.org/10.1109/CVPR.2017.667.

[31] M. Zhai, X. Xiang, R. Zhang, N. Lv, A. El-Saddik, Optical flow estimation using channel attention mechanism and dilated convolutional neural networks, Neurocomputing 368 (2019) 124–132, https://doi.org/10.1016/j.neucom.2019.08.040.

[32] X. Ran, Z. Shan, Y. Fang, C. Lin, An lstm-based method with attention mechanism for travel time prediction, Sensors 19 (4) (2019) 861, https://doi.org/10.3390/s19040861.

[33] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015.URL: http://arxiv.org/abs/1412.6980.

Yukai Ding was born in 1996. He is currently pursuing M.S. degree in computer science and technology, Hohai University, Nanjing, China. His research interests include pattern recognition and data mining.

Yuelong Zhu was born in 1959. He is currently a Professor with the School of Computer and Information, Hohai University. His main research interests include intelligent information processing, data mining, and water resources informatization.

Jun Feng was born in 1969. She received the B.S. and M. S. degrees in computer science and technology from Hohai University, China, in 1991 and 1994, respectively, and the Ph.D. degree in information engineering from the University of Nagoya, Japan, in 2004. She is currently a Professor with the School of Computer and Information, Hohai University. Her research interests include data management, spatiotemporal indexing and search methods, knowledge engineering, and domain data mining.

Pengcheng Zhang was born in 1981. He received the Ph.D. degree in computer science from Southeast University in 2010. He is currently an associate professor in College of Computer and Information, Hohai University, Nanjing, China, and was a visiting scholar at San Jose State University, USA. His research interests include water Informatics, software engineering, service computing, and data mining. He co-authored more than 70 peer-reviewed conference and journal papers, and has served as technical program committee member on various international conferences.

Zirun Cheng was born in 1997. She is currently pursuing M.A. degree in translation, Tongji University, Shanghai, China. Her research interests include translation between English and Chinese.