



Research papers

A novel multi-step ahead forecasting model for flood based on time residual LSTM



Yongsong Zou^a, Jin Wang^{a,b,*}, Peng Lei^a, Yi Li^a

^a School of Hydraulic & Environmental Engineering, Changsha University of Science & Technology, Changsha 410014, China

^b School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha 410014, China

ARTICLE INFO

Keywords:

Residual long short-term memory

Flood forecasting

Rainfall-runoff

Time series task

Probability distribution

ABSTRACT

Climate change has significantly impacted hydrology, including extreme precipitation, and changing precipitation patterns that could lead to an increase in flooding. Life and property benefit from accurate and reliable multi-step flood forecasting. Recently, Recurrent Neural Network (RNN) have become increasingly popular among hydrology researchers for their ability to capture historical dependencies, simplify computations by ignoring intermediate hydrological processes, and provide higher prediction accuracy than traditional models. However, RNN-based flood prediction models face two significant challenges. Firstly, due to their strict time-serial, RNN suffer from gradient issues such as vanishing and exploding gradients, which can make training RNN models difficult. To address this issue, we propose a Residual Long Short-Term Memory (ResLSTM) model that incorporates time residual connections into the time connections of LSTM. Secondly, most flood prediction models output a deterministic value, but the natural hydrological characteristics of the basin are a nonlinear and complex system with many influencing factors that have some randomness. This requires the use of probabilistic methods to modeling. Thus, we introduce the probabilistic forecasting model Autoregressive Recurrent Networks (DeepAR) into our flood prediction model, which outputs a prediction interval rather than a deterministic value. Then, we build four flood probability prediction models by combining DeepAR and four enhanced RNN, including ResLSTM (ours), Long Short-Term Memory (LSTM), Gate Recurrent Unit (GRU), and Time Feedforward Connections Simple Gate Recurrent Unit (TFC-SGRU). The performance of these models is evaluated by the long-term hydrologic data of the Passaic and Ramapo River basins in the United States. The results demonstrate that the prediction interval of the four models is more adaptive to flood uncertainties. And the accuracy of peak flow prediction is nearly 100% within a 90% prediction probability interval. The temporal residual-based model is more accurate and robust than the original LSTM and GRU. We believe this study fills a research gap in multi-step-ahead flood probability prediction and improves the accuracy and reliability of flood prediction models.

1. Introduction

According to the United Nations World Water Development Report (UNESCO, 2020), 74% of the world's disasters over the past two decades were water-related. Effective flood forecasting is essential for mitigating flood risk and managing water resource systems. Hydrodynamic models based on physics and Machine Learning (ML) models based on data have been developed to study flood risk. Hydraulic and hydrodynamic models are commonly used in flood research (Wen et al., 2013; Shrestha et al., 2020). Although hydraulic and hydrodynamic models are incredibly efficient, they require precise boundary conditions and complex calculations (Jodhani et al., 2021).

Thus, more and more researchers have recently focused on ML. ML has been applied to the analysis of various natural hazard types, such as floods (Campolo et al., 1999; Li et al., 2011), utilizing techniques such as support vector regression (Choubin et al., 2019; Yu et al., 2006), decision trees (Bui et al., 2019; Khosravi et al., 2018), random forest (Tang et al., 2021; Zahura et al., 2020), genetic algorithms (Arturo et al., 2021; Wang et al., 2011), Optimally Pruned extreme Learning Machine (OP-ELM) (Adnan et al., 2019; Adnan et al., 2020), and two ML training strategies for self-organizing maps to address the issue of non-convergence during clustering of high-dimensional datasets (Chang et al., 2021). Unlike physics-driven models, ML does not require precise boundary conditions and complicated physical calculations.

* Corresponding author.

E-mail address: jinwang@csust.edu.cn (J. Wang).

Nonetheless, flood analysis is a dynamic process with a temporal variation. ML-type models lack mechanisms to capture long-term dependencies. Deep Learning (DL) based on ML has come to the attention of researchers.

The Recurrent Neural Network (RNN) Is a type of DL used to perform time series tasks. RNN can extract spatial-temporal features from data and is advantageous for predicting flood risks (Yaseen et al., 2015). RNN predicts flooding by mining historical data for driver factors. Thus, RNN has been successfully applied to many Earth science problems (Reichstein et al., 2019; Wang et al., 2020; Ikram et al., 2023) and related applications such as hydrology (Munawar et al., 2021). Le et al. (Le et al., 2019) proposed a model for flood prediction based on Long Short-Term Memory (LSTM). The results demonstrated that the LSTM's predictive ability was quite remarkable. Fang et al. used the LSTM to predict flood susceptibility with dependable accuracy (Fang et al., 2021). The authors proposed a hybrid neural network model for flood forecasting by combining a Convolutional Neural Network (CNN) with an LSTM (Chen et al., 2022). Google provides a neural network alternative for hydraulic modeling of flood inundations. The LSTM model is used for real-time flood warnings (Nevo et al., 2022). The results showed that the LSTM is more accurate than the conventional model. In most catchments, the RNN models provide a more accurate functional representation of rainfall-runoff than the ML models (Feng et al., 2020; Kratzert et al., 2019).

Although RNN has significant advantages in various time series tasks (Liu et al., 2019; Mikolov et al., 2010; Singh et al., 2017), RNN remains a huge challenge because the time-serial dependencies limit the accuracy. The time-serial of RNN refers to the process of unraveling the recurrent connections of the network over time. By unrolling the network over time, we create a temporal sequence of connected neural network layers that can be trained with backpropagation through time to learn complex temporal dependencies. RNN's time-series dependencies can cause difficulties in training flood prediction models based on RNN, as the repeated multiplication of gradients over time can result in the vanishing and exploding gradients issues. (Bengio et al., 1994). LSTM (Gers et al., 2000; Hochreiter and Schmidhuber, 1997) and Gate Recurrent Unit (GRU) (Cho et al., 2014) have been proposed to address gradient issues with "gates" strategies. An LSTM cell consists of an input gate, an output gate, and a forget gate. Consequently, a GRU cell comprises an update gate and a reset gate. Although LSTM and GRU improve the gradient issues in RNN by employing "gates" strategies, the problem of vanishing and exploding gradients resulting from the time-serial dependencies of RNN still persists and has not been fundamentally resolved. However, most RNN-based flood prediction models utilize the original model without addressing its limitations, such as hybrid long short-term memory neural network and ant lion optimizer model (LSTM-ALO) (Yuan et al., 2018), LSTM-ED multi-step-ahead flood forecasting by combining the advantages of Encoder-Decoder and LSTM (Kao et al., 2020), SAE-RNN by combining stacked autoencoders (SAE) with RNN (Kao et al., 2021). Adnan proposed an improved adaptive neuro-fuzzy system (ANFIS) model called ANFIS-GBO, which enhances the model's performance through various model heterogeneity methods using gradient-based optimization techniques (Adnan et al., 2022). (Wang et al., 2023) proposed the Time Feedforward Connections Simple Gate Recurrent Unit (TFC-SGRU) by introducing Highway Networks (HN) (Srivastava et al., 2015) to improve the time-series dependencies. The TFC-SGRU model incorporates shortcut connections between time steps t-2 and t to enable direct information flow without the need for nonlinear transformations at time step t, thereby altering the time-series dependencies of traditional RNN. HN is a time residual scheme that incorporates parameters. However, due to the high computational complexity of LSTM (Pascanu et al., 2013), a no-parameter residual strategy may be superior to HN. Inspired by TFC-SGRU and ANFIS-GBO, we propose a novel RNN architecture, called time Residual LSTM (ResLSTM), to address the gradient problem caused by the time-series dependencies of RNN. Unlike TFC-SGRU, ResLSTM adopts a no-

parameter residual scheme by introducing a no-parameter residual ResNet (He et al., 2016; Yue et al., 2018).

The natural temporal systems are often impacted by multiple uncertain factors and randomness, which makes accurate predictions challenging. Nevertheless, the probabilistic models enable the prediction of the range and probability distribution. To address the probabilistic forecasting issue in RNN, researchers proposed a probabilistic forecasting approach based on autoregressive recurrent networks (DeepAR) (Salinas et al., 2020). DeepAR is a deep learning-based time series probabilistic forecasting model. The aim of flood prediction is to get the future water level or peak streamflow at a specific time. However, basin river systems' hydrological characteristics are nonlinear and complex, influenced by several factors. Future values are often affected by numerous uncertainties such as future rainfall, infiltration, terrain, basin characteristics, etc. As a result, accurate single value of future flood peak streamflow is often challenging. A more appropriate approach would be to estimate the possible range of values and probability distribution of future flood peak streamflow using a probabilistic model, which would better reflect the system characteristics. Here, the DeepAR model is introduced to solve the problem of flood probability distribution. We propose a novel temporal residual long-short-term memory model for flood probability prediction based on DeepAR and ResLSTM. Finally, we employ three powerful optimization techniques, namely data cleaning, normalization, and feature filtering, to enhance the performance of the flood probability prediction model. Based on these approaches, our model is not only capable of capturing the attribution information of flood conditioning factors and the local spatial information of flood data, but it also has powerful sequential modeling capabilities to handle the spatial-temporal relationship of floods.

The main contributions of this paper are summarized as follows.

- We proposed a ResLSTM that can improve the gradient problem of LSTM by breaking the time-serial.
- We introduced the probability forecasting DeepAR to ResLSTM to solve the flood probability distribution.
- We presented a multi-step ahead model for predicting the probability of flooding.
- We constructed four flood probability prediction models by integrating DeepAR and four enhanced RNN, including ResLSTM, LSTM, GRU, and TFC-SGRU.

The rest of this paper is organized as follows. Section 2 presents the RNN, DeepAR models, and computational methods. Section 3 describes the studied basin and data. Section 4 introduces our ResLSTM and the structure of the flood probability prediction model. Section 5 provides the experimental results based on the hydrologic data of the Passaic River and Ramapo River basin in the United States. Section 6 concludes this paper.

2. Methodology

2.1. Recurrent neural network

RNN is a common Deep Neural Network (DNN) used to handle time series tasks (Chung et al., 2014). RNN was first proposed in 1990 (Elman, 1990). We can express it as follows:

$$h^t = \tanh(Wx^t + Vh^{t-1} + b) \quad (1)$$

where t is the time step. x^t and h^t represent the input matrix and hidden state, respectively. h^{t-1} denotes the previous hidden state. W, V, b represent the input weight, the hidden weight, and the bias, respectively. $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

In contrast to DNN, the input of each RNN cell comprises input from the current time as well as output from the previous time. Therefore, RNN are more powerful than DNNs in learning sequences. However,

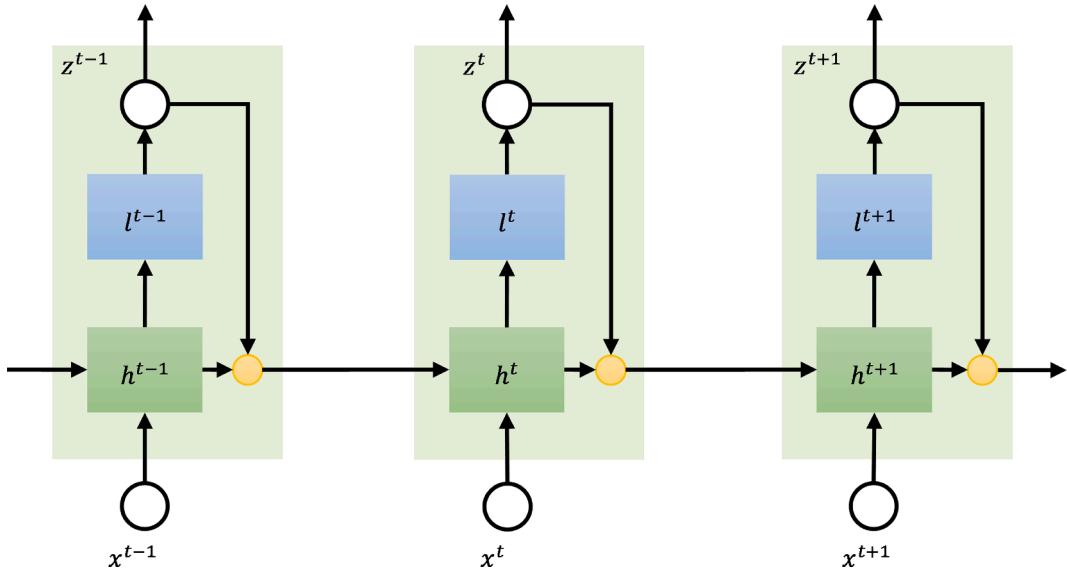


Fig. 1. The framework of DeepAR. x^t , l^t , z^t is the input data, likelihood (Eq. (17)) and output at time step t , respectively.

traditional RNN are difficult to train due to gradients that vanish and explode. To solve this issue, LSTM is proposed.

$$f^t = \text{sigmoid}(W_f x^t + V_f h^{t-1} + b_f) \quad (2)$$

$$i^t = \text{sigmoid}(W_i x^t + V_i h^{t-1} + b_i) \quad (3)$$

$$o^t = \text{sigmoid}(W_o x^t + V_o h^{t-1} + b_o) \quad (4)$$

$$g^t = \tanh(W_g x^t + V_g h^{t-1} + b_g) \quad (5)$$

$$c^t = i^t \otimes g^t + f^t \otimes c^{t-1} \quad (6)$$

$$h^t = o^t \otimes \tanh(c^t) \quad (7)$$

where f^t , i^t , o^t , c^t , h^t represent the forget gate, the input gate, the output gate, the memory state, and the hidden output state, respectively. Memory state is intended to capture long-term memory. The input gate is used to filter the input information. The forget gate can filter out noise from the past. The output gate is the final output checkpoint. W , V , b represent the input weight, the hidden weight, and the bias, respectively. \otimes is the Hadamard product. The activate function $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$.

While the performance of LSTM has improved, its computational complexity has increased proportionally. Then, a simplified gated LSTM, denoted by GRU, is proposed.

$$z^t = \text{sigmoid}(W_z x^t + V_z h^{t-1} + b_z) \quad (8)$$

$$r^t = \text{sigmoid}(W_r x^t + V_r h^{t-1} + b_r) \quad (9)$$

$$\tilde{h}^t = \tanh(Wx^t + r^t \otimes h^{t-1}) \quad (10)$$

$$h^t = (1 - z^t) \otimes h^{t-1} + z^t \otimes \tilde{h}^t \quad (11)$$

where z^t , r^t represent the update gate and the reset gate, respectively.

h^t , \tilde{h}^t represent the hidden output state and the transition state, respectively. W , V , b represent the input weight, the hidden weight, and the bias, respectively. GRU reduces training parameters by eliminating LSTM's output gate.

Nonetheless, the performance of LSTM and GRU is still constrained by the time-serial dependence alone. The RNN models are trained using

Backpropagation Through Time (BPTT). Continuous multiplication is used to update gradients, which limits the performance of the 'gate' method. Thus, the authors proposed a time residual simple GRU by introducing the HN named TFC-SGRU, as expressed as follows:

$$z^t = \text{sigmoid}(W_z x^t + V_z h^{t-1} + b_z) \quad (12)$$

$$s^t = \text{sigmoid}(W_s x^t + V_s h^{t-2} + b_s) \quad (13)$$

$$\tilde{h}^t = \tanh(Wx^t + z^t \otimes h^{t-1}) \quad (14)$$

$$y^t = (1 - z^t) \otimes h^{t-1} + z^t \otimes \tilde{h}^t \quad (15)$$

$$h^t = (1 - s^t) \otimes h^{t-2} + s^t \otimes y^t \quad (16)$$

where z^t , s^t represent the update gate and carry gate, respectively. h^t , h^{t-1} , and h^{t-2} represent hidden output at time t , $t-1$, and $t-2$, respectively.

Due to the time shortcut of TFC-SGRU, gradient update is achieved by a sum expression instead of continuous multiplication. TFC-SGRU enhances feed-forward connections between time steps, thereby enabling the gradient from $t-2$ to t without the nonlinear transformation at time t . TFC-SGRU can improve the gradient issue in horizontal RNN propagation.

2.2. Probabilistic forecasting model

DeepAR is a probabilistic model for forecasting time series data. We can receive both a predicted value and a probability distribution. DeepAR is an embedded autoregressive recurrent network (RNN). Set a sequential series $z^t (t \in [0, T])$, and the index t_0 . $(1, 2, \dots, t_0 - 1)$ are the conditioning range, and $(t_0, \dots, T-1, T)$ are the prediction range. The DeepAR is shown in Fig. 1. h^t is the hidden output of RNN at time t . x^t and z^t are the input series and output values at time t , respectively. l^t represents the likelihood function. The Gaussian likelihood for real-valued data. Then, we set the likelihood as follows:

$$l(z|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(z - \mu)^2}{2\sigma^2}\right) \quad (17)$$

$$\mu^t = \omega_\mu^T h^t + b_\mu \quad (18)$$

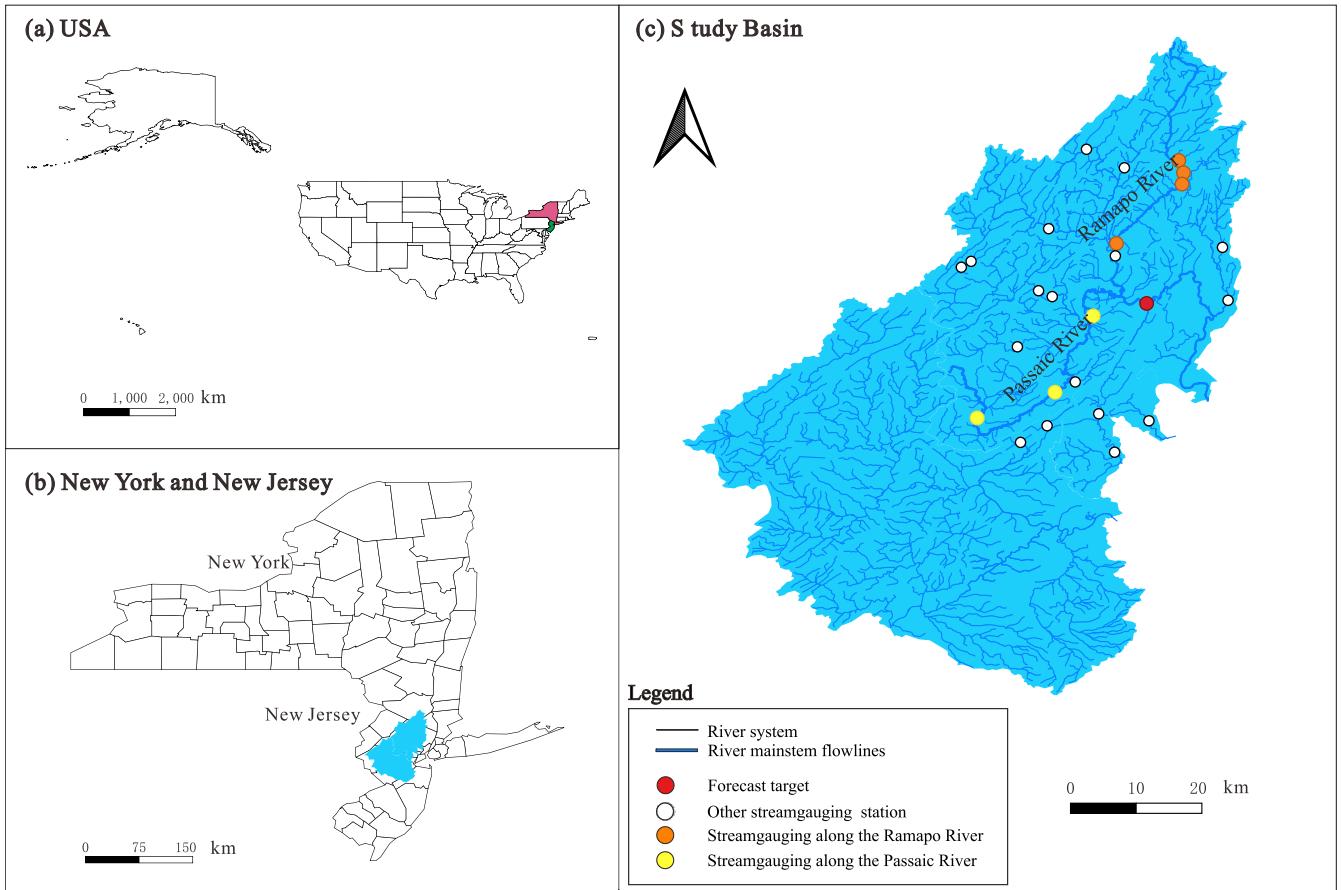


Fig. 2. Location of the Passaic River and Ramapo River basin. (a) The map of the United States. (b) The map of New Jersey and New York, United States. (c) The study basin. There are hydrological stations situated on both rivers – represented by the yellow circles for the Passaic River stations, and the orange circles for the Ramapo River stations – as well as several other stations denoted by white circles. The red circle is the objective station. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\sigma^t = \omega_\sigma^T h^t + b_\sigma \quad (19)$$

where μ is the mean, σ is the variance, ω is the weight matrix, and b is the bias.

DeepAR learns the internal characteristics of the different periods to improve the accuracy of predictions. In contrast to non-probabilistic models, this network predicts distribution parameters rather than a single value. There are two benefits to output distribution: (1) Since most nonlinear systems have random properties, the output probability distribution is closer to nature. (2) It can evaluate the forecast's uncertainty and risk.

2.3. Evaluation of model performance

In this paper, the aggregate performance of our models is evaluated using statistical error metrics, such as the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), and the Mean Absolute Scaled Error (MASE), as illustrated in Eqs. (20) to (22).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\tilde{y}_i - y_i)^2} \quad (20)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |\tilde{y}_i - y_i| \quad (21)$$

$$MASE = \frac{MAE}{\frac{1}{T-1} \sum_{t=2}^T |y_t - y_{t-1}|} \quad (22)$$

where n represents the number of samples, and \tilde{y}_i , y_i denote the predicted and observed values, respectively. $t = 1, 2, \dots, T$ is the time of training sets.

RMSE also referred to as the standard error, is calculated as the square root of the ratio between the sum of squared deviations of observed values from true values and the number of observations. It is commonly used to quantify the deviation between observed and true values and is particularly sensitive to both large and small errors within a set of measurements, thus making it an effective measure of precision. However, one major limitation of RMSE is also its susceptibility to outliers, which can distort the results and lead to misleading conclusions. In addition, RMSE may be less intuitive to interpret since it measures errors in squared units, which may not be directly comparable to the original units of the data. MAE is a commonly used metric for assessing prediction accuracy. It calculates the average of the absolute differences between predicted values and actual values. MAE is a robust measure that is not affected by outliers since it considers only the absolute error between the predicted and actual values, regardless of the direction of the error. For stable prediction models, MAE provides a reliable measure of the error. However, the disadvantage of MAE is that it treats all errors equally, irrespective of their magnitude. This drawback can lead to scenarios where MAE does not effectively capture the magnitude of the prediction error. Moreover, MAE does not indicate the direction of the error. MASE is calculated by dividing the mean error by a scaling factor, which is determined by the seasonal value m selected based on the forecasting frequency. A lower MASE value indicates a higher accuracy of the forecasting model. MASE is particularly suitable for datasets that exhibit periodic or seasonal characteristics. Considering

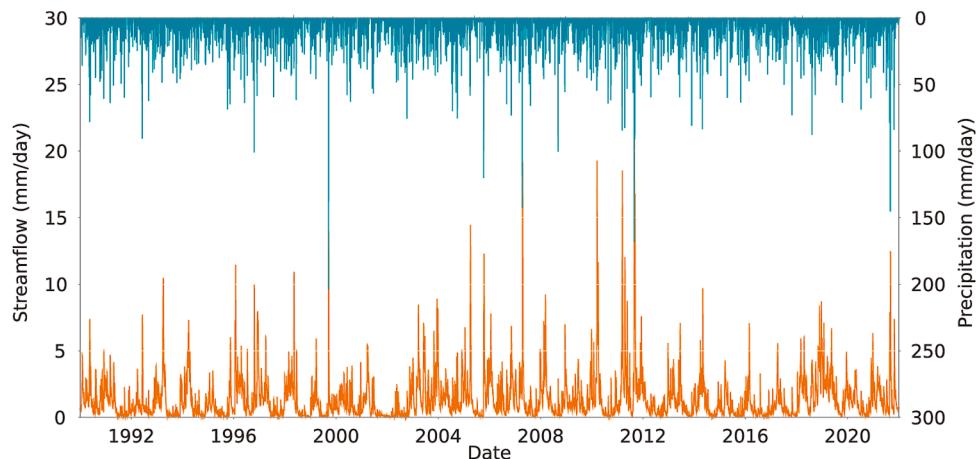


Fig. 3. The streamflow observations for USGS-01389500. The left y-axis represents the streamflow, and the right y-axis represents the precipitation.

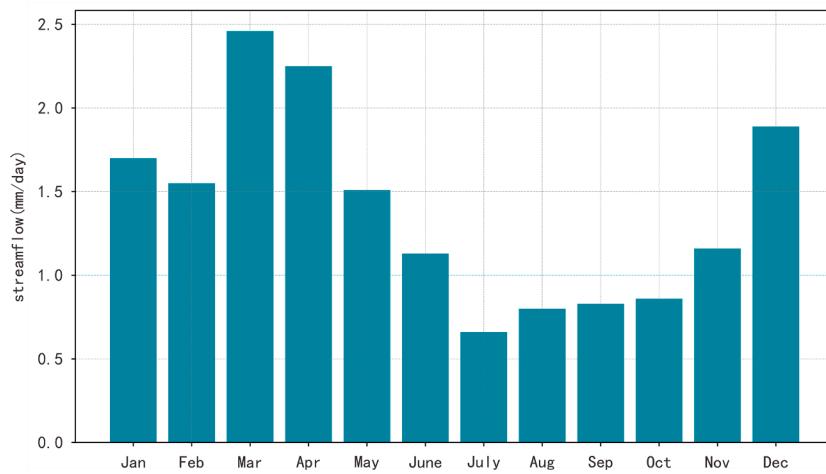


Fig. 4. The multi-year monthly average streamflow for USGS-01389500, presented on a month-by-month basis.

the seasonal effects can improve the accuracy of forecasting floods with low rainfall during the rainy season and less rainfall during the dry season.

3. Studied basin and data

Our research is focused on a watershed located at the confluence of the Passaic River and Ramapo River. We use data from the Passaic River and the Ramapo River in New Jersey and New York in the United States, as shown in Fig. 2(a – c). The Passaic River Basin is located below Millington Canyon and covers about 881 square miles. The United States Geological Survey (USGS) monitors this river from four different hydrologic observation stations along the Passaic River, as shown in Fig. 2(c) by the yellow circles. The Ramapo River is monitored by four distinct USGS hydrologic observation stations, as depicted by the orange circles in Fig. 2(c). The red circle in Fig. 2(c) represents the forecast objective station (USGS-01389500) on the left bank of the Passaic River at Little Falls, New Jersey, 0.6 miles downstream of Beatties Dam in Little Falls and 1.0 mile upstream of Peckman River.

Fig. 3 depicts the streamflow and precipitation for USGS-01389500. The green line on the left y-axis represents the streamflow, while the orange line on the right y-axis represents the precipitation. The hydrological features include daily precipitation and streamflow from 25 hydrological stations spanning the 32-year record from 1990/01/01 to 2021/12/31 which are downloaded from the USGS website. The average annual discharge of the objective station is $17.98 \text{ m}^3/\text{s}$. According to

Fig. 3, the first half of the year is the wet season, and the second half is the dry season. Typically, April, May, and June are the busiest months of the year. Consequently, the streamflow of USGS-01389500 is zero every November.

Based on the data collected from hydrological stations, we have determined that the average annual precipitation in the Passaic and Ramapo River watershed between 1990 and 2021 was 1258 mm/day, while the target station's average annual flow rate was 1.4 mm/day. To gain a deeper understanding of the station's multi-year monthly flow rates, we used statistical calculations to determine the multi-year average flow rate for each month. Specifically, by analyzing the watershed streamflow monthly average distribution chart in Fig. 4, we discovered that the multi-year average streamflow in March was the highest at 2.46 mm/day, while June had the lowest average streamflow at 0.66 mm/day. These findings indicate that the watershed faces significant rainfall pressure in March and requires measures to prevent flooding. Fig. 5.

4. Our proposed flood forecast model

4.1. Residual long short-term memory

ResNet is a CNN architecture that is well-known for its ability to train CNNs effectively, with the capability to train networks with up to 152 layers. Training CNNs with many layers before ResNet was difficult due to vanishing gradients, leading to poor performance and making it

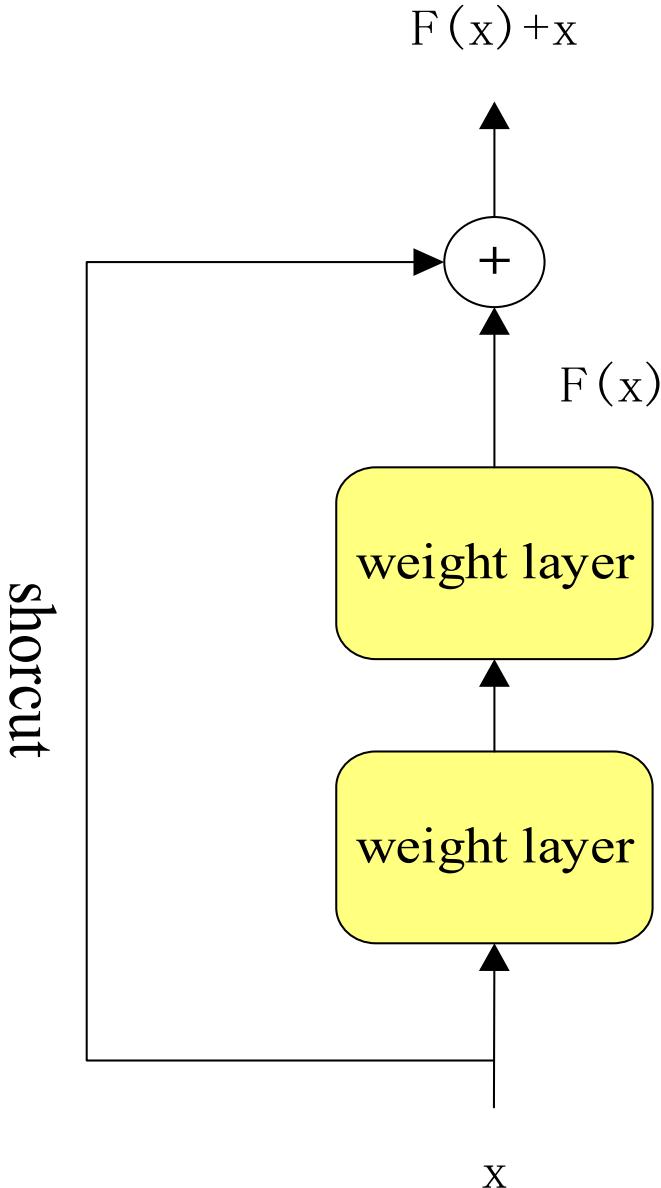


Fig. 5. The structure of ResNet.

challenging for networks to learn from input data. To address this issue, residual learning was proposed, which involves allowing the network to learn the residual mapping between the input and output by adding a shortcut connection on top of the original single connection between the layers. The shortcut connection is a simple identity mapping that directly passes inputs to outputs without modification by weight layers.

Despite the emergence of many RNN models, LSTM remains the best-performing one (Jozefowicz et al., 2015). However, LSTM has

limitations such as too many training parameters and time-serial dependencies that can decrease its accuracy. To address this, we propose a time residual LSTM by incorporating shortcut connections from ResNet into the time-serial connections of LSTM, as shown in Fig. 6. This allows the hidden output h^{t-2} to be directly inputted to the cell at time t without the nonlinear transformation at time $t-1$.

To maximize the spatial-temporal information of streamflow and precipitation, we enhance the residual network structure of the LSTM cell. As with the GRU model, we begin by erasing the LSTM forget gate to reduce the number of parameters and accelerate convergence. The internal structure of the ResLSTM cell is illustrated in Fig. 7. The time residual of ResLSTM is reflected in Eq. (27).

The ResLSTM cell consists of an input gate and an output gate. Each gate of the ResLSTM cell connects the information of the input and the output (h^t) of the hidden state. Eqs. (23)–(27) provide the principle of the ResLSTM model.

$$i^t = \text{sigmod}(W_i x^t + V_i h^{t-1} + b_i) \quad (23)$$

$$o^t = \text{sigmod}(W_o x^t + V_o h^{t-1} + b_o) \quad (24)$$

$$g^t = \tanh(W_g x^t + V_g h^{t-1} + b_g) \quad (25)$$

$$c^t = i^t \otimes g^t + f^t \otimes c^{t-1} \quad (26)$$

$$h^t = o^t \otimes \tanh(c^t) + (1 - o^t) \otimes \tanh(h^{t-2}) \quad (27)$$

where i^t, o^t, c^t, g^t , and h^t represent the input gate, the output gate, the memory state, the internal state, and the hidden output state at time t , respectively. W, V , and b represent the input weight, the hidden weight, and the bias, respectively. \otimes is the Hadamard product. Like the LSTM model, the input gate filters the input information, and c^t is responsible for capturing long-term memory.

We remove the forget gate of LSTM and add the short-term information h^{t-2} as the time shortcut to break the time-serial dependency. Significantly, the time shortcut permits the gradient at step $t-1$ to be transferred to step $t+1$ without the nonlinear transformation at step t . More importantly, we use the no-parameter residual method instead of the parametric residuals in TFC-SGRU, so there are no training parameters.

4.2. Flood probability forecast model design

Suppose there is an $n+k$ steps time series task, where m represents the number of features in a single time step (one day), n represents the input time steps, s denotes the predicted start time, k represents the predicted time steps, and $p^{s+1:s+k}$ is the predicted sequence.

As shown in Fig. 8, our flood probability forecast model consists of four layers: the input layer, the RNN layer, the likelihood layer, and the output layer. The RNN layer captures the temporal dependencies of the input array and identifies a more accurate functional representation of rainfall-runoff.

The RNN layers depicted in Fig. 8 utilize ResLSTM, but can be substituted with other models such as LSTM and GRU. The output of the

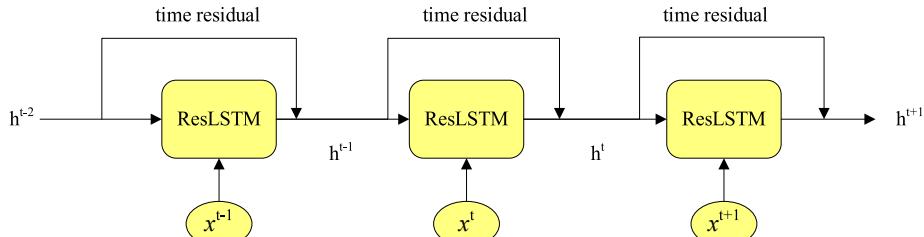


Fig. 6. The structure of ResLSTM network. x^t, h^t represent the input data and hidden output at time t , respectively.

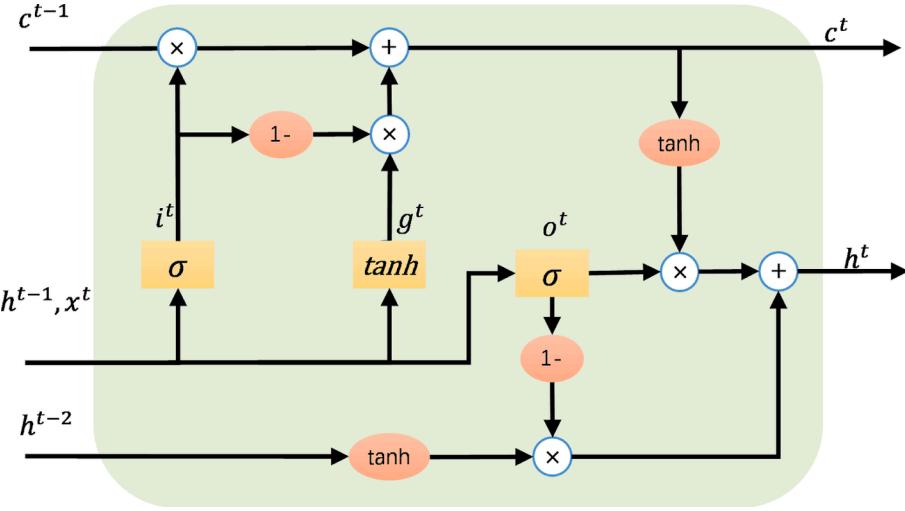


Fig. 7. The structure of ResLSTM cell.

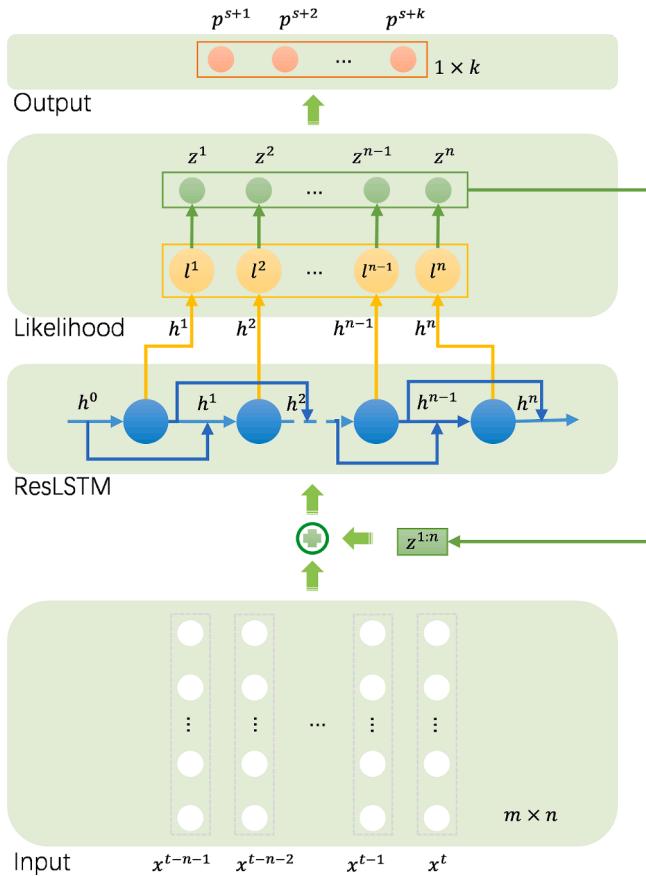


Fig. 8. The framework of the multi-step flood probabilistic forecasting model is based on ResLSTM and DeepAR.

RNN is transferred to the likelihood layer, which can search for possible output expression intervals. The output layer outputs the prediction result $p^{(s+1:s+k)}$ and probability intervals.

4.3. Input datasets

The hydrological features include daily precipitation and discharge from 25 hydrological stations. Three powerful optimization techniques, data cleaning, normalization, and features filter, are adapted to further

enhance the model's performance for predicting the probability of a flood. The dataset is then divided into three subsets.

- (1) Data Cleaning: Due to equipment failures, the original dataset contains incomplete or missing data. Missing data, which is a common occurrence, can have a significant impact on the model's performance (Brunner et al., 2021). As shown in Eq. (28), we use the mean of the still noise to handle the missing or low-level outlier.

$$\tilde{x}_t = \frac{x_{t-1} + x_{t+1}}{2} \quad (28)$$

where x_{t-1} and x_{t+1} are the values of the time $t-1$ and the time $t+1$. \tilde{x}_t is the missing value.

- (2) Normalization: We standardize input features to the range [0,1] with Max-Min normalization, as shown in Eq. (29).

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (29)$$

where x_{max} and x_{min} represent the maximum and minimum, respectively.

- (3) Features filter: We evaluate the correlation between each input feature and the streamflow of USGS-01389500 using the Pearson coefficient. The Pearson coefficient $\rho_{r,p}$ of array R and P is defined as: "The covariance cov(R, P) between array R and array P divided by the product of their respective standard deviations $\sigma_r \sigma_p$," as is shown in Eq. (30).

$$\rho_{r,p} = \frac{cov(R, P)}{\sigma_r \sigma_p} \quad (30)$$

where $\rho_{r,p}$ is the Pearson coefficient, $cov(R, P)$ represents the covariance of R and P, and σ_r is the standard deviation. The Pearson correlation coefficient is used to measure the strength and direction of the linear relationship between two variables, and it ranges from -1 to 1. A value of 1 indicates a perfect positive correlation between the variables, while a value of -1 indicates a perfect negative correlation. A value of 0 indicates no linear relationship between the variables.

In neural network feature engineering, the Pearson correlation coefficient is commonly used in the steps of feature selection and feature engineering to help identify the most relevant features to the target variable. It can be used to identify highly correlated feature variables and to help exclude those feature variables that have no linear relationship with the target variable.

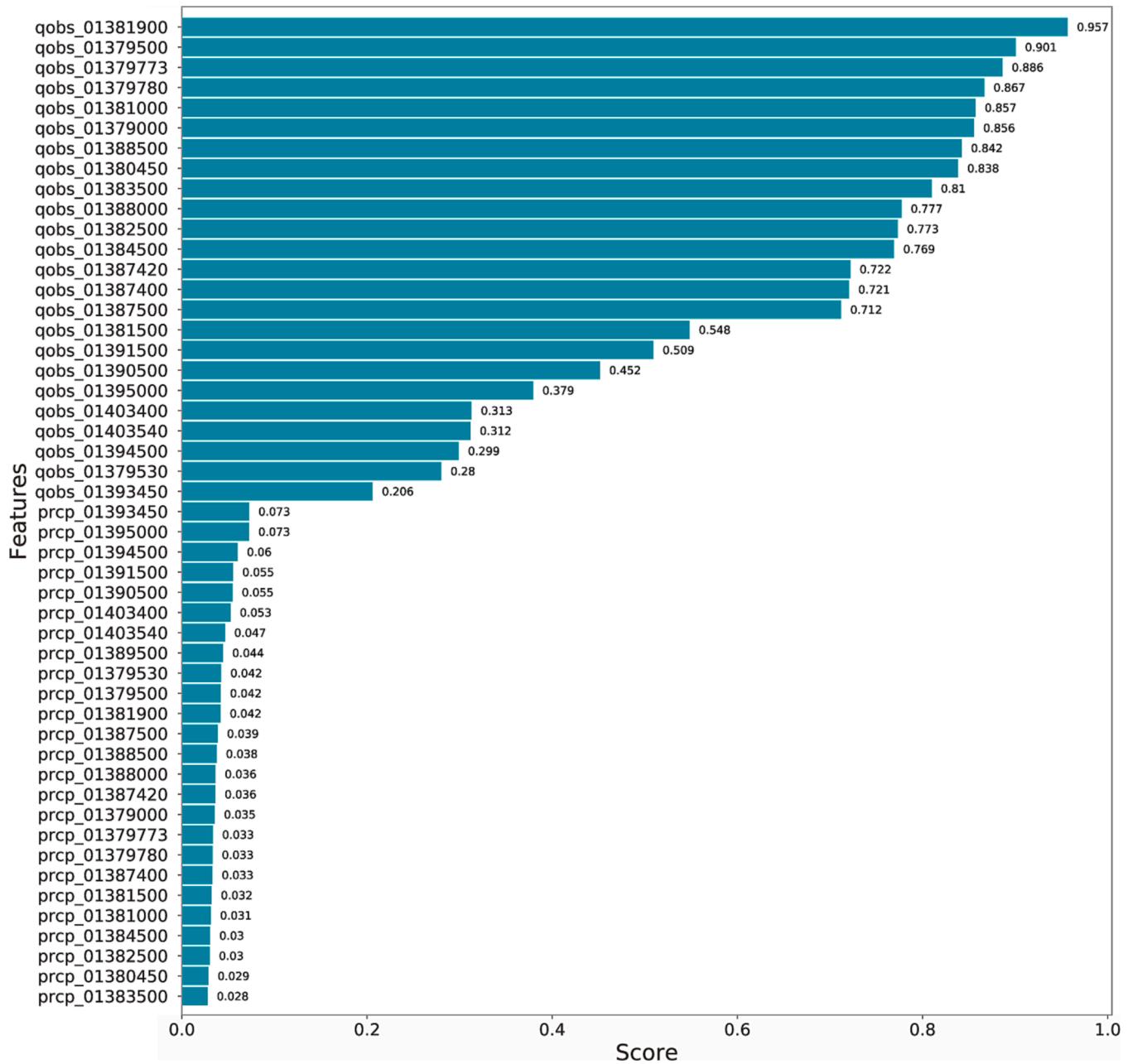


Fig. 9. The result of the Pearson coefficient. “qobs” and “prcp” present the streamflow and precipitation. The numerical values appended to “qobs” or “prcp” denote the identification number of the hydrological station.

Fig. 9 shows the result of the Pearson coefficient of 50 input features of 25 hydrological stations. The numbers on the y-axis are the hydrological station ID. The higher score, the greater the significance of the feature. Then, we use the threshold $score > 0.04$ to filter features with low correlation.

- (4) Split datasets: The datasets cover daily precipitation and streamflow at each station. The output is USGS-01389500 streamflow. About 80% of the data in each dataset is used for training, 10% for validating, including 1080 steps, and 10% for testing. The test set contains 1080-time steps (days) of hydrologic data from 2019/01/17 to 2021/12/31. We determined two output strategies for our experiments: one-step prediction (1 day) with 1080 repetitions and eighteen-step prediction (18 days) with 60 repetitions.

5. Experimental results and discussion

5.1. Parameters setting

Our model for predicting flood probability is based on RNN and DeepAR. The RNN layer is essential for obtaining the flood model's long-term dependencies. This study compares ResLSTM to three modified RNN models: LSTM, GRU, and TFC-SGRU.

Training an RNN model requires selecting appropriate hyperparameters. Hyperparameters are manually set parameters that cannot typically be learned from the training data and require experience and trial-and-error to determine their optimal values. Common hyperparameters include the context length, the number of RNN layers, the number of RNN cells in each layer, the learning rate, the learning rate decay factor, and the number of training epochs.

However, due to the large number and complexity of

Table 1

The best hyperparameters of models.

Output steps	Model	Context	Layers	Cells	Learning	Decay	Epochs
One	ResLSTM-DeepAR	36	6	160	0.00335	0.95	160
	TFC-SGRU-DeepAR	72	4	256	0.00205	0.95	200
	LSTM-DeepAR	180	4	192	0.00370	0.65	180
	GRU-DeepAR	36	6	208	0.00160	0.95	170
Eighteen	ResLSTM-DeepAR	72	4	192	0.00210	0.75	60
	TFC-SGRU-DeepAR	72	2	176	0.00260	0.95	50
	LSTM-DeepAR	36	4	256	0.00170	0.55	60
	GRU-DeepAR	36	2	224	0.00200	0.75	170

Table 2

Evaluation metrics of four models for flood forecasting on the test data set.

Output steps	Model	RMSE	MASE	MAE
One	ResLSTM-DeepAR	0.203	0.374	0.080
	TFC-SGRU-DeepAR	0.213	0.377	0.080
	LSTM-DeepAR	0.159	0.358	0.076
	GRU-DeepAR	0.165	0.388	0.083
Eighteen	ResLSTM-DeepAR	0.248	0.667	0.142
	TFC-SGRU-DeepAR	0.264	0.774	0.165
	LSTM-DeepAR	0.262	0.710	0.151
	GRU-DeepAR	0.291	0.760	0.162

hyperparameters, the search space can be vast. To select the best hyperparameter combination, techniques such as grid search, random search, and Bayesian hyperparameter optimization algorithms can be used. In this paper, we utilize the Optuna open-source Python library to search for the optimal hyperparameter combination using the Bayesian optimization algorithm. The Bayesian optimization algorithm is a method for optimizing neural network model hyperparameters by

continuously updating the prior distribution of hyperparameters during the training process to find the optimal hyperparameters. Table 1 shows the optimal hyperparameters for the four probabilistic prediction models under the two output strategies.

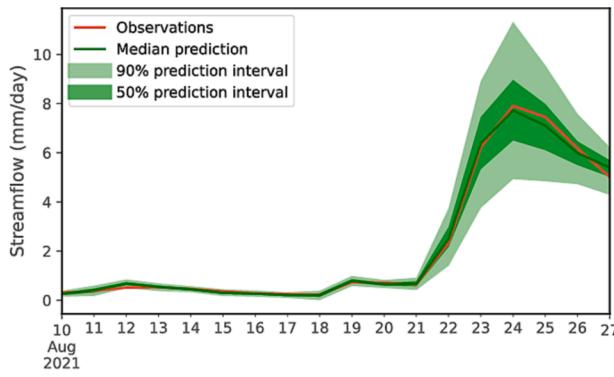
5.2. Performance evaluation

5.2.1. The two output strategies results

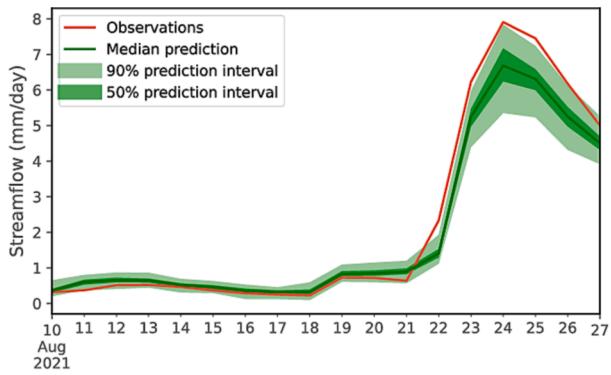
For the experiments in this paper, the model error is evaluated using the aggregate performance metrics RMSE, MASE, and MAE. The smaller the RMSE, MAE, and MASE, the smaller the error and the greater the aggregate performance. Therefore, these three metrics can effectively evaluate the robustness and regression accuracy of the four models.

Table 2 illustrates the evaluation metrics for streamflow prediction at various output stages for the four models. The accuracy of all models is strongly correlated with the output time step. As the number of output time steps increases, the accuracy of the four models decreases.

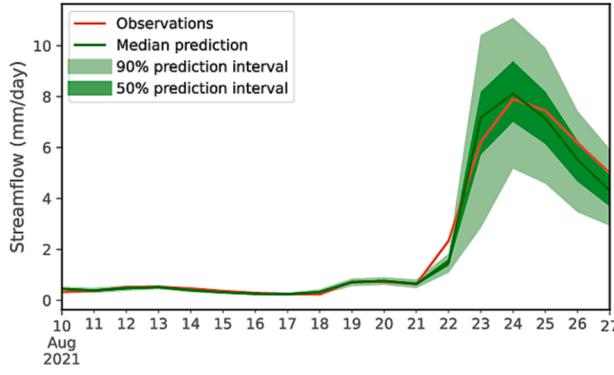
The results show that ResLSTM achieves RMSE of 0.213 for one-step output, but drops to 0.248 for eighteen steps, MASE drops from 0.374 to 0.667, and MAE drops from 0.080 to 0.142. The same applies to the



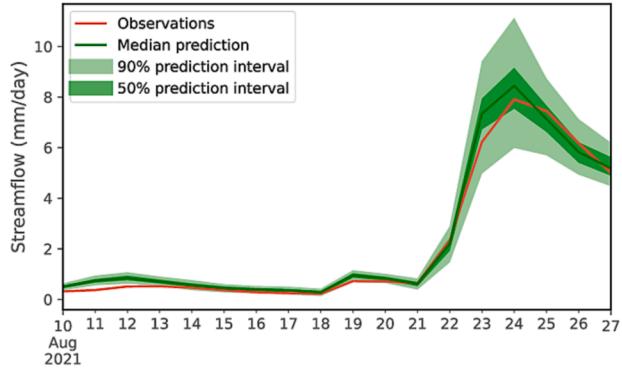
(a) ResLSTM 52nd forecast



(b) TFC-SGRU 52nd forecast



(c) LSTM 52nd forecast



(d) GRU 52nd forecast

Fig. 10. The multi-step forecast results of the 10th-27th August 2021 flood event. Each hydrograph shows the observation, median prediction, 50%, and 90% intervals. (a) The ResLSTM results. (b) The TFC-SGRU result. (c) The LSTM results. (d) The GRU results.

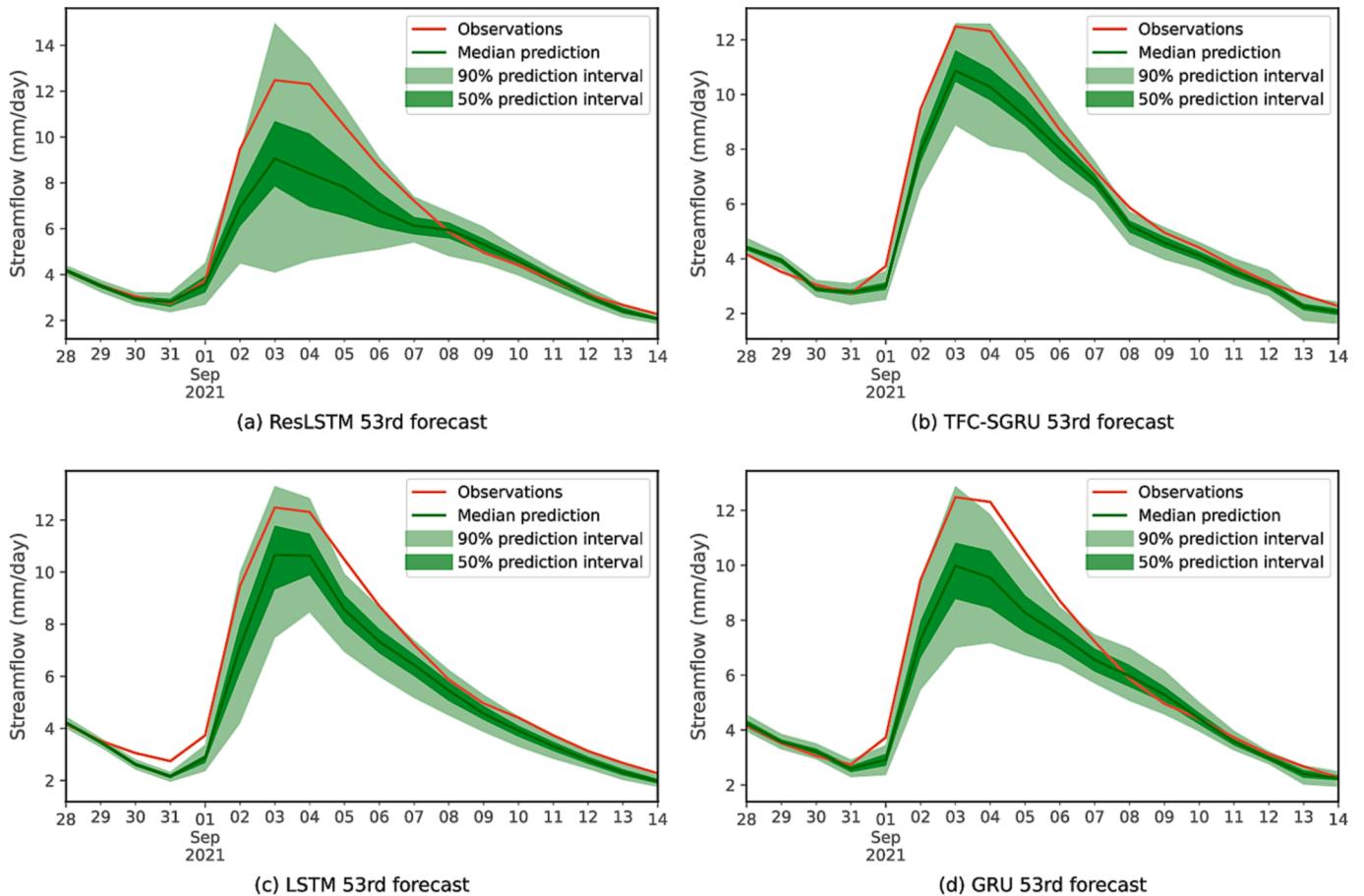


Fig. 11. The multi-step forecast results for the flood event that occurred from August 28 to September 14 in 2021, as predicted by four models. Each hydrograph shows the observation, median prediction, 50%, and 90% prediction intervals.

remaining models. The reason is that the longer the prediction step, the more memory the RNN needs. ResLSTM performs marginally better than TFC-SGRU for single-step output, but there is no significant difference. In addition, the aggregate performance of the LSTM model outperforms that of all other models across all metrics.

However, the ResLSTM model performs better than the others for the eighteen-step output. ResLSTM and TFC-SGRU are residual-based models that incorporate the ResNet and Highway networks. The residual-based model is more accurate and robust than the original LSTM and GRU for long-term prediction. The residual network can reduce the RMSE, MAE, and MASE of LSTM and GRU, indicating that the residual module influences the RNN positively. In multi-step prediction, our proposed ResLSTM outperforms competing models.

MAE is the mean absolute value of the sum of all individual observations' deviations from the arithmetic mean. Since RMSE is sensitive to outliers, MAE can assess the network's robustness more accurately. Table 2 demonstrates that the RMSE is greater than the MAE for both output strategies. This can be due to the large variation in monthly stream flows and the extremely inhomogeneous temporal distribution of precipitation within the study basin.

5.2.2. Multi-step ahead flood peak probability intervals prediction

The steps of test datasets are 1080 while the output steps are 18 of multi-step prediction, so we need to run 60 times. During the time period of the test datasets, there are three floods. The first flood occurred at the 52nd forecast on 2021/08/24, the second at the 53rd forecast on 2021/09/03, and the third at the 56th forecast on 2021/10/27. Our model is intended to generate probabilistic predictions, rather than point forecasts. The output is a prediction interval. Figs. 10–12 display the

eighteen-step prediction results for three flood events from the ResLSTM, TFC-SGRU, LSTM, and GRU models. Each of these plots shows the observations, the median prediction, the 50% prediction interval, and the 90% prediction interval.

Fig. 10(a–d) depict the 90% prediction interval, the 50% prediction interval, and the observed streamflow by the four models for the 10th–27th August 2021 flood event. The four models predict a reliable occurrence time of the flood peak. The TFC-SGRU model forecasted a flood peak that was one day late and underestimated the streamflow in the rising period (see Fig. 10(b)). During ascent, the ResLSTM and GRU models perform better than the LSTM and TFC-SGRU models. However, the ResLSTM, LSTM, and GRU models outperform the TFC-SGRU model during the recession. The ResLSTM model outperforms the other models in terms of flood peak flow. The observation and 50% prediction interval are in excellent agreement during periods of expansion and contraction (see Fig. 10(a)).

Fig. 11(a–d) display the 90% prediction interval, the 50% prediction interval, and the observed streamflow for the flood event that occurred from August 28 to September 14 in 2021, as predicted by four models. The observed flood hydrographs are higher than the first flood due to the relatively concentrated precipitation. There may be a correlation between the irregular fluctuations of peak prediction intervals and the spatial-temporal distribution of precipitation. The four models severely underestimate the high flow rate due to the small number of training samples of high flow rate peaks. During the rise, the 90% prediction interval of the TFC-SGRU model is less than the observed flow. Nevertheless, the 90% prediction interval of the GRU is less than the observed value during the recession. The 90% prediction interval of ResLSTM and LSTM covers the observed flood peaks. The LSTM model

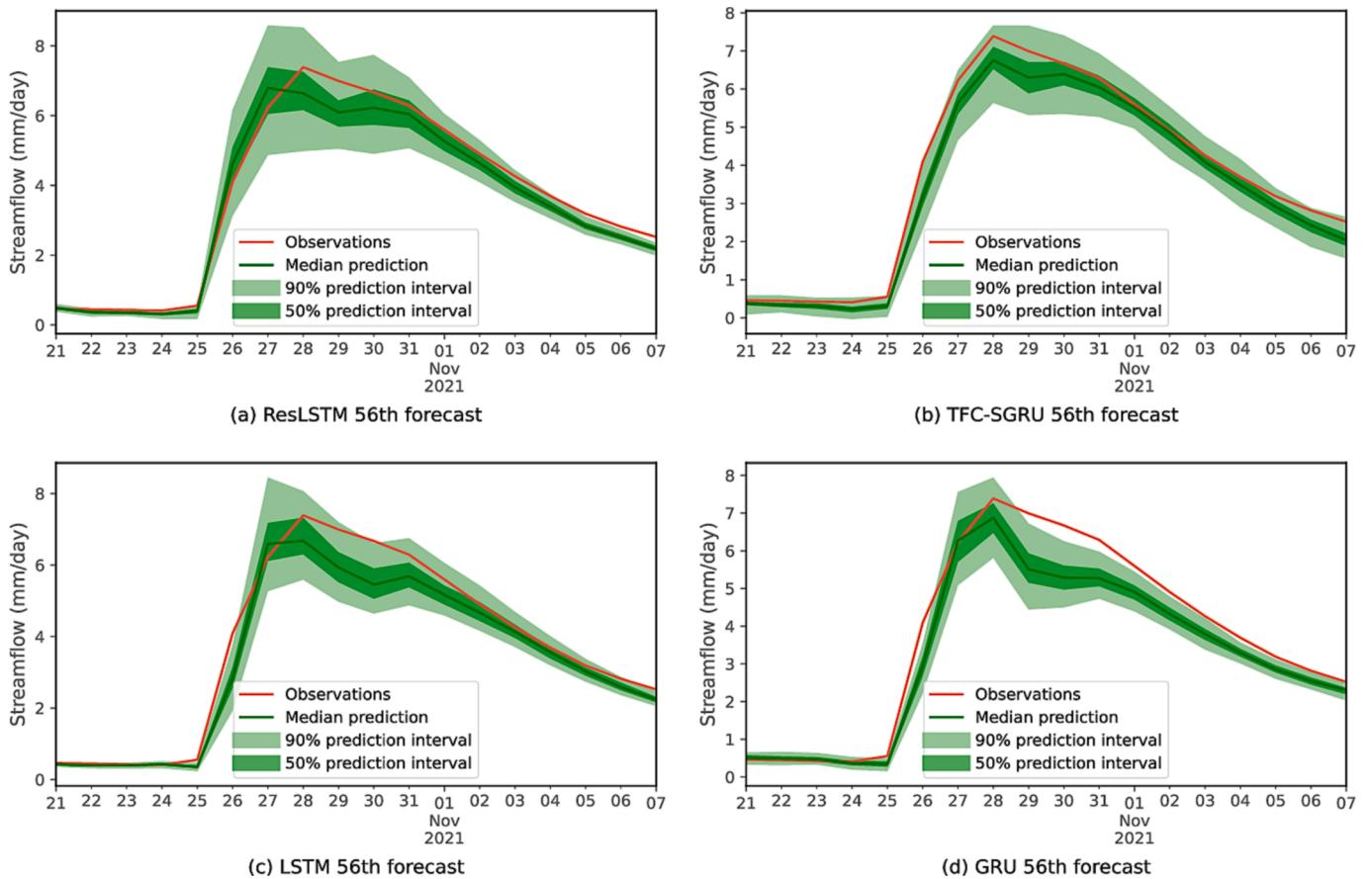


Fig. 12. The multi-step forecast results by the four models for the 2021 flood event that occurred from October 21 to November 7. Each hydrograph shows the observation, median prediction, 50%, and 90% prediction intervals.

Table 3
The optimal quantiles.

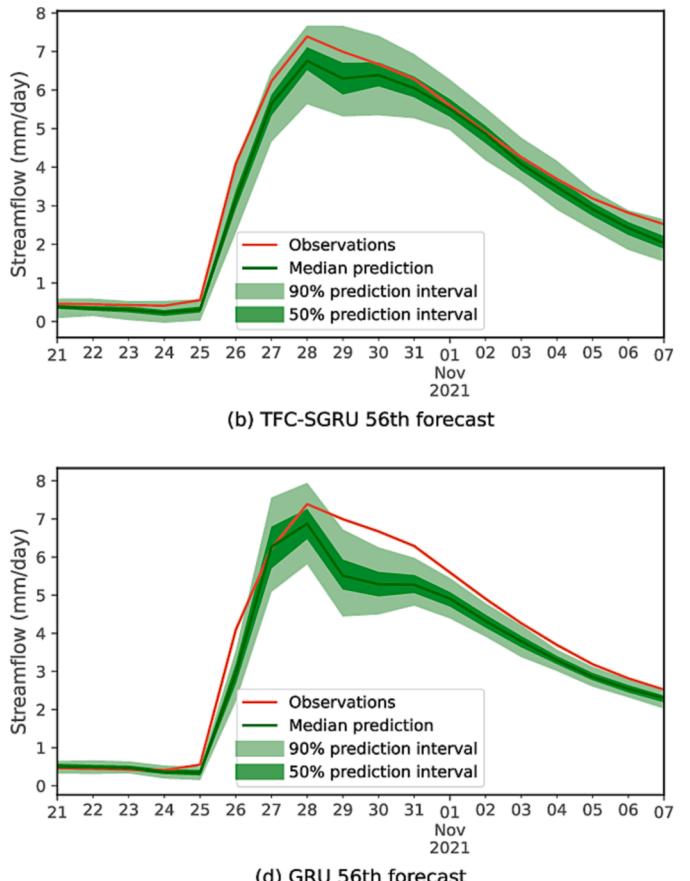
Output steps	Model	52nd	53rd	56th
eighteen	ResLSTM-DeepAR	0.56	0.78	0.88
	TFC-SGRU-DeepAR	0.56	0.90	0.92
	LSTM-DeepAR	0.52	0.82	0.90
	GRU-DeepAR	0.40	0.74	0.92

performs better than the other models in terms of flood peak magnitude, whereas the ResLSTM model performs better overall.

Fig. 12(a – d) depict the 90% prediction interval, the 50% prediction interval, and the observed streamflow by the four models for the 2021 flood event that occurred from October 21 to November 7. The 50% prediction interval of the ResLSTM and TFC-SGRU models covers the observed hydrograph of the flood peak, whereas the LSTM and GRU models underestimate the flood peak. During the ascent, the ResLSTM and TFC-SGRU models perform reasonably well. The LSTM and GRU models perform adequately well during a recession. The ResLSTM and LSTM models perform reasonably well during the flood peak occurrence time.

5.2.3. Comparison of the impact of different quantiles on multi-step ahead flood peak prediction

Probability intervals in flood prediction offer accurate and comprehensive information, indicating uncertainty and aiding decision-making. Our model, using DeepAR, generates probability intervals with an 18-step output strategy, showing the potential range of streamflow values over the next 18 days. Derived from historical data, these intervals reveal the expected range of future flood flow values.



Different quantiles have varying accuracy for probability distribution. To further analyze model precision, we have identified optimal quantiles for four probability prediction models in three flood predictions, as shown in Table 3.

Under optimal quantile conditions, the peak flood prediction errors for three floods were compared across four flood prediction models in Fig. 13. All models performed well with peak flood prediction error less than 8.31%. ResLSTM had the smallest error, less than 2%.

TFC-SGRU had lower errors than LSTM and GRU in the first and third peak flood predictions, but had a higher error than GRU in the second prediction. LSTM and GRU had similar errors in the three peak flood predictions, with GRU slightly outperforming LSTM, indicating GRU's slightly better performance on small data sets such as hydrological data.

Fig. 14 illustrates the impact of different quantiles on the accuracy of flood event predictions, comparing three events using ResLSTM models. (a)-(c) display 18-step predictions at the 0.56, 0.78, and 0.88 quantiles. For the first flood event, the 0.56 quantile matches well with observed streamflow during the rising period, while the 0.78 quantile outperforms other results during the recession period (Fig. 14(a)). For the second flood event, the 0.88 quantile performs reasonably well during both rising and recession periods (Fig. 14(b)). The accuracy of the 0.56 and 0.78 quantiles for the third flood event is not significantly different (Fig. 14(c)). ResLSTM results indicate that for the first flood, the peak of the 0.56 quantile is closer to observed peak streamflow compared to the 0.88 quantile. For the second flood, the peak value of the 0.88 quantile is closer to the observed peak streamflow. For the third flood, the peak streamflow of the 0.78 quantile is closer to the observed peak streamflow.

Fig. 14(d – f) depict the prediction quantiles 0.56, 0.90, and 0.92 for the three flood events by the TFC-SGRU models. In Fig. 14(d), the

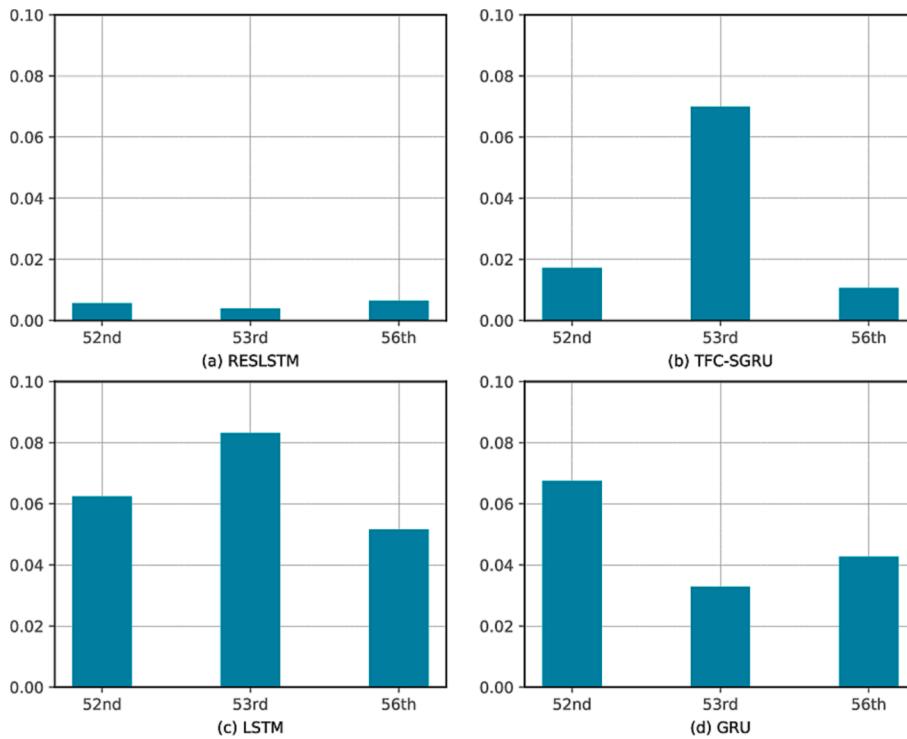


Fig. 13. Peak flood prediction errors for three flood events among the four models.

forecast interval for the first flood's rising period is narrow, indicating that there are few uncertainties. Thus, each quantile-specific result is highly consistent. In contrast, during the recession, the result of quantile 0.92 corresponds with observation (see Fig. 14(d)). As with ResLSTM, the 0.95 quantile of TFC-SGRU yields the best results for the second flood (see Fig. 14(e)). For the third flood, all three quantiles are close to the observed streamflow during the rising period, whereas the 0.92 quantile is closer than the others during the recession period (see Fig. 14(f)). The TFC-SGRU results indicate that the 0.90 quantile result is closer to the observed peak streamflow of the initial flood. Quantile 0.92 is closer to the observed peak streamflow for the second flood, while quantile 0.90 is closer to the observed peak streamflow for the third flood.

Fig. 14(g – i) illustrate the prediction quantiles 0.52, 0.82, and 0.90 for the three flood events by LSTM models. The quantiles 0.52 and 0.82 are close to the observed streamflow than the quantile 0.90 in the rising and recession periods for all flood events. The results of the LSTM algorithm indicate that the peak of the quantile 0.52 is closer to the observed peak flow of the initial flood. Quantile 0.82 is closer to the observed peak streamflow for the second flood; quantile 0.82 is closer to the observed peak streamflow for the third flood.

Fig. 14(j – l) illustrate the prediction quantiles 0.40, 0.74, and 0.92 for the three flood events by the GRU models. For the first and third floods, the results of the GRU model are comparable to those of the LSTM model, with the quantiles of 0.40 and 0.74 being closer to the observed streamflow than the quantile of 0.92 for the rising and recessions. But for the second flood, the result of quantile 0.92 is in good agreement with the observed streamflow (see Fig. 14(k)). The GRU results show that the peak of the quantile 0.40 is closer to the observed peak streamflow of the first flood. The peak of quantile 0.92 is closer to the observed peak streamflow for the second flood. The peak streamflow of quantile 0.74 is closer to the observed peak streamflow for the third flood.

Fig. 15 presents 12 violin plots that exhibit the 18-step multi-step flood prediction results for three floods in the test set, using four distinct flood prediction models. A violin plot is a graphical tool used to

compare the distribution of numerical variables across multiple groups. By plotting several violins on the same graph, differences between different groups can be readily visualized. Fig. 14 includes 12 violin plots where each subplot contains four violins. The leftmost violin displays the distribution of observed values, while the other three violins show the distribution of predictions at different quantiles.

Different quantiles used by different models for predicting different flood events are sourced from Table 3. In the first flood prediction (a, d, g, j), the predictions of all four models based on the smallest quantile are very close to the observed values, including the median and maximum values. This suggests that the predictions based on the smallest quantile are more accurate and the models have reliable prediction capability with high accuracy in peak flood prediction. In the second flood prediction (b, e, h, k), the largest quantile violins of all four models were very close to the observed value violin plot. The violin shapes were very similar, indicating that the predicted results had a consistent data distribution with the observed values. However, when comparing the peaks of the quantile violins and the observed value violin plot, it was found that TFC-SGRU and GRU models underestimated the peak, while the LSTM model overestimated it, indicating that all three models had large prediction errors for the flood peak. The ResLSTM model produced the best violin plot. In the third flood prediction (c,f,i,l), the peaks of the violin plots for all three quantiles' results from the ResLSTM model are relatively close to the observed values, indicating accurate flood peak prediction. Additionally, the shapes of the ResLSTM's violin plots for all three quantiles' results are similar to the observed values, suggesting reliable prediction of river streamflow during the flood process. The peaks of the last two quantiles' results of LSTM and GRU's violin plots are roughly similar to the observed values. But the distribution of the result is not consistent with the observed values. Once again, ResLSTM performs the best.

Our findings are in agreement with the previous analysis and provide additional evidence supporting the effectiveness of our improved ResLSTM model compared to other models. Specifically, our results suggest that the flood probability prediction model we proposed, which is based on RNN and DeepAR, is better suited for handling flood

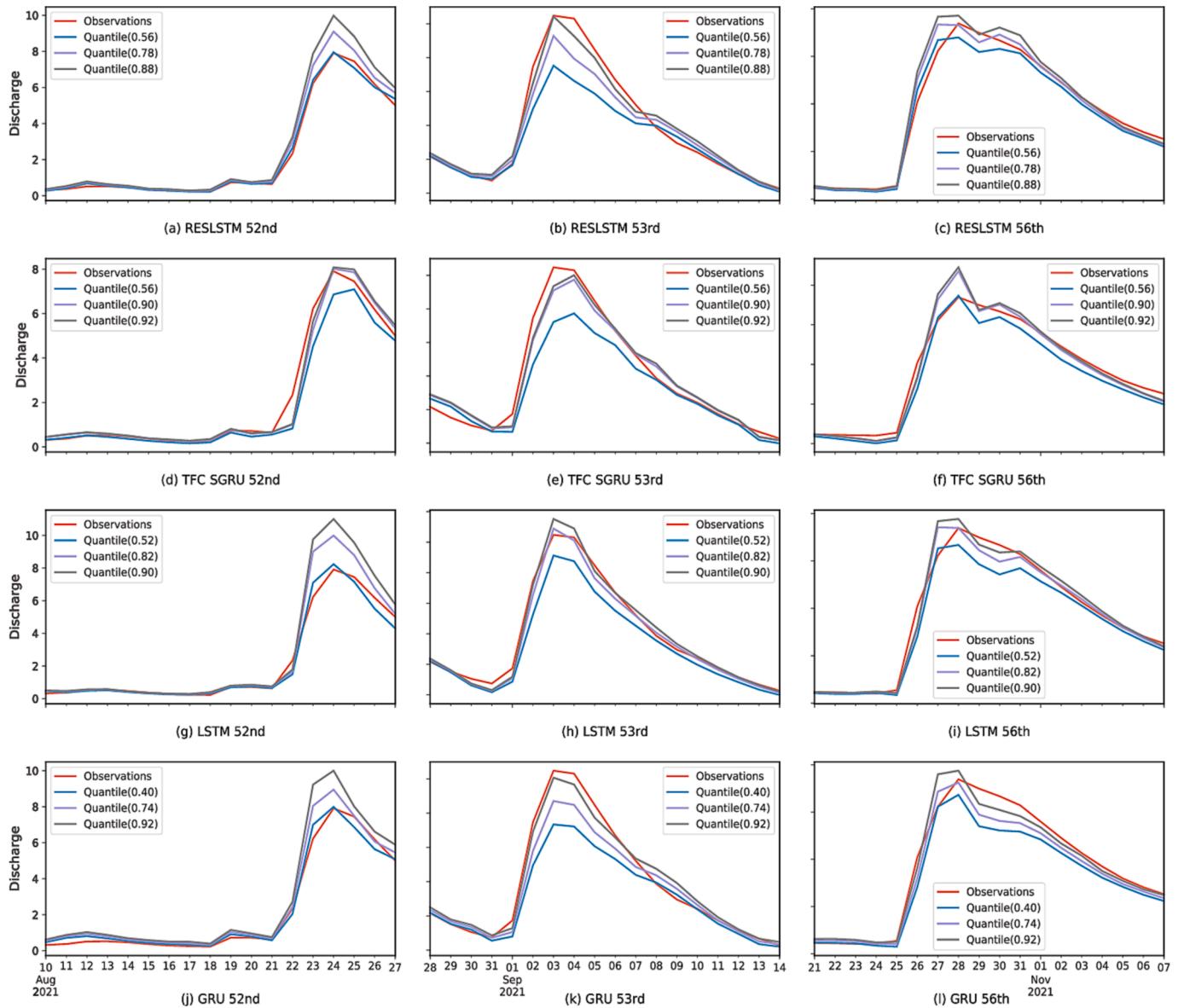


Fig. 14. The influence of different quantiles on the prediction accuracy is compared for the three flood events.

uncertainty due to its superior adaptability.

6. Conclusion and future work

This study proposes an effective multi-step model for predicting the probability of flooding based on RNN and DeepAR. First, a ResLSTM is proposed to improve the LSTM for flood prediction. The temporal shortcut of ResLSTM can improve the gradient problem of LSTM. Second, the probabilistic model DeepAR is introduced to make flood uncertainty more adaptable. The results demonstrate that probabilistic forecasting models are more adaptable to flood uncertainty, and the temporal residual-based model is more precise and robust than the original LSTM and GRU models.

Although ResLSTM has shown promising results in solving the gradient problem, its performance in parallel computing is still limited, resulting in slow training speeds. Therefore, it is worth exploring new algorithms to build flood prediction models, such as the multi-head attention mechanism. In addition, the use of advanced optimization algorithms, such as Adam and Adagrad, can also potentially improve the training speed of ResLSTM. Future research can focus on combining

ResLSTM with these advanced techniques to further improve the performance of flood prediction models.

CRediT authorship contribution statement

Yongsong Zou: Data curation, Methodology, Visualization, Software, Writing – original draft. **Jin Wang:** Conceptualization, Funding acquisition, Validation, Writing – review & editing. **Peng Lei:** Investigation, Writing – review & editing, Resources. **Yi Li:** Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

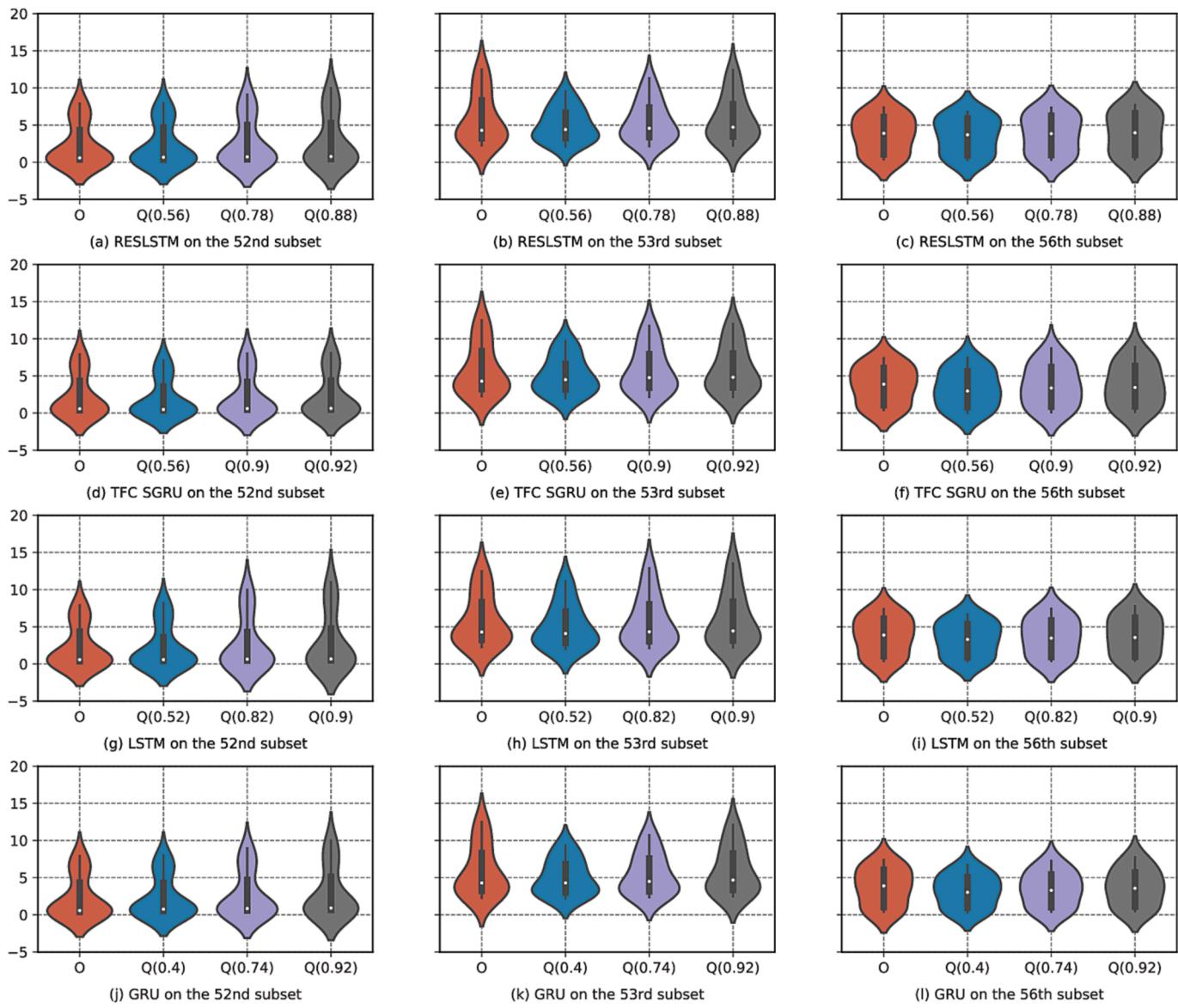


Fig. 15. The violin plots illustrate the results of 18-step multi-step flood prediction for three floods in the test set using four distinct flood prediction models. (a)-(c) ResLSTM-DeepAR, (d)-(f) TFC-SGRU-DeepAR, (g)-(i) LSTM-DeepAR, and (j)-(l) GRU-DeepAR.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (62072056).

References

- Adnan, R.M., Liang, Z., Trajkovic, S., Zounemat-Kermani, M., Li, B., Kisi, O., 2019. Daily streamflow prediction using optimally pruned extreme learning machine. *Journal of Hydrology*. 577, 123981.
- Adnan, R.M., Liang, Z., Heddam, S., Zounemat-Kermani, M., Kisi, O., Li, B., 2020. Least square support vector machine and multivariate adaptive regression splines for streamflow prediction in mountainous basin using hydro-meteorological data as inputs. *Journal of Hydrology* 586, 124371.
- Adnan, R.M., Mostafa, R.R., Elbeltagi, A., Yaseen, Z.M., Shahid, S., Kisi, O., 2022. Development of new machine learning model for streamflow prediction: Case studies in Pakistan. *Stochastic Environmental Research and Risk Assessment*. 36, 999–1033. <https://doi.org/10.1007/s00477-021-02111-z>.
- Arturo, S.L., Linlong, B., Yun, T., 2021. Comparison of the genetic algorithm and pattern search methods for forecasting optimal flow releases in a multi-storage system for flood control. *Environmental Modelling and Software*. 145, 105198 <https://doi.org/10.1016/j.envsoft.2021.105198>.
- Bengio, Y., Simard, P.Y., Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*. 5, 157–166. <https://doi.org/10.1109/72.279181>.
- Brunner, M.I., Slater, L., Tallaksen, L.M., Clark, M., 2021. Challenges in modeling and predicting floods and droughts: A review. *Wiley Interdisciplinary Reviews: Water*. 8, e1520.
- Bui, D.T., Tsangaratos, P., Ngo, P.T.T., Pham, T.D., Pham, B.T., 2019. Flash flood susceptibility modeling using an optimized fuzzy rule based feature selection technique and tree based ensemble methods. *Science of The Total Environment*. 668, 1038–1054. <https://doi.org/10.1016/j.scitotenv.2019.02.422>.
- Campolo, M., Andreussi, P., Soldati, A., 1999. River flood forecasting with a neural network model. *Water Resources Research*. 35, 1191–1197. <https://doi.org/10.1029/1998WR900086>.
- Chang, L.C., Wang, W.H., Chang, F.J., 2021. Explore training self-organizing map methods for clustering high-dimensional flood inundation maps. *Journal of Hydrology* 595, 125655. <https://doi.org/10.1016/j.jhydrol.2020.125655>.
- Chen, C., Jiang, J., Liao, Z., Zhou, Y., Wang, H., Pei, Q., 2022. A short-term flood prediction based on spatial deep learning network: A case study for Xi County, China. *Journal of Hydrology*. 607, 127535 <https://doi.org/10.1016/J.JHYDROL.2022.127535>.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. doi: 10.48550/arXiv.1406.1078.
- Choubin, B., Moradi, E., Golshan, M., Adamowski, J., Sajedi-Hosseini, F., Mosavi, A., 2019. An ensemble prediction of flood susceptibility using multivariate discriminant

- analysis, classification and regression trees, and support vector machines. *Science of the Total Environment*. 651, 2087–2096. <https://doi.org/10.1016/j.scitotenv.2018.10.064>.
- Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555. doi: 10.48550/arXiv.1412.3555.
- Elman, J.L., 1990. Finding structure in time. *Cognitive Science*. 14, 179–211. [https://doi.org/10.1016/0364-0213\(90\)90002-E](https://doi.org/10.1016/0364-0213(90)90002-E).
- Fang, Z., Wang, Y., Peng, L., Hong, H., 2021. Predicting flood susceptibility using LSTM neural networks. *Journal of Hydrology*. 594, 125734 <https://doi.org/10.1016/j.jhydrol.2020.125734>.
- Feng, D., Fang, K., Shen, C., 2020. Enhancing streamflow forecast and extracting insights using long-short term memory networks with data integration at continental scales. *Water Resources Research*. 56, e2019WR026793. doi: 10.1029/2019wr026793.
- Gers, F.A., Schmidhuber, J., Cummins, F., 2000. Learning to forget: Continual prediction with LSTM. *Neural Computation*. 12, 2451–2471. <https://doi.org/10.1049/cp:19991218>.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Computation*. 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735.
- Ikram, R. M. A., Mostafa, R. R., Chen, Z., Parmar, K. S., & Zounemat-Kermani, M., 2023. Water temperature prediction using improved deep learning methods through reptile search algorithm and weighted mean of vectors optimizer. *Journal of Marine Science and Engineering*. 11, 259. doi: 10.3390/jmse11020259.
- Jodhani, K.H., Patel, D., Madhavan, N., 2021. A review on analysis of flood modelling using different numerical models. *Materials Today: Proceedings*.
- Jozefowicz, R., Zaremba, W., Sutskever, I., 2015. An empirical exploration of recurrent network architectures. In: *International Conference on Machine Learning* (pp. 2342–2350).
- Kao, I.F., Zhou, Y., Chang, L.C., Chang, F.J., 2020. Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting. *Journal of Hydrology*. 583, 124631 <https://doi.org/10.1016/j.jhydrol.2020.124631>.
- Kao, I.F., Liou, J.Y., Lee, M.H., Chang, F.J., 2021. Fusing stacked autoencoder and long short-term memory for regional multistep-ahead flood inundation forecasts. *Journal of Hydrology*. 598, 126371. <https://doi.org/10.1016/j.jhydrol.2021.126371>.
- Khosravi, K., Pham, B.T., Chapi, K., Shirzadi, A., Shahabi, H., Revhaug, I., Prakash, I., Bui, D.T., 2018. A comparative assessment of decision trees algorithms for flash flood susceptibility modeling at Haraz watershed, northern Iran. *Science of the Total Environment*. 627, 744–755. <https://doi.org/10.1016/j.scitotenv.2018.01.266>.
- Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A.K., Hochreiter, S., Nearing, G.S., 2019. Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*. 55, 11344–11354. <https://doi.org/10.1029/2019WR026065>.
- Le, X.H., Ho, H.V., Lee, G., Jung, S., 2019. Application of Long Short-Term Memory (LSTM) neural network for flood forecasting. *Water*. 11, 1387. <https://doi.org/10.3390/w11071387>.
- Li, J., Heap, A.D., Potter, A., Daniell, J.J., 2011. Application of machine learning methods to spatial interpolation of environmental variables. *Environmental Modelling Software*. 26, 1647–1659. <https://doi.org/10.1016/j.envsoft.2011.07.004>.
- Liu, P., Wang, J., Sangaiah, A.K., Xie, Y., Yin, X., 2019. Analysis and prediction of water quality using LSTM deep neural networks in IoT environment. *Sustainability*. 11, 2058. <https://doi.org/10.3390/su11072058>.
- Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., Khudanpur, S., 2010. Recurrent neural network based language model. In: *Interspeech* (Vol. 2, No. 3, pp. 1045–1048).
- Munawar, H.S., Hammad, A.W., Waller, S.T., 2021. A review on flood management technologies related to image processing and machine learning. *Automation in Construction*. 132, 103916 <https://doi.org/10.1016/j.autcon.2021.103916>.
- Nevo, S., Morin, E., Gerzi Rosenthal, A., Metzger, A., Barshai, C., Weitzner, D., Voloshin, D., Kratzert, F., Elidan, G., Dror, G., Begelman, G., Nearing, G., Shalev, G., Noga, H., Shavit, I., Yuklea, L., Royz, M., Giladi, N., Peled Levi, N., Reich, O., Gilon, O., Maor, R., Timnat, S., Shechter, T., Anisimov, V., Gigi, Y., Levin, Y., Moshe, Z., Ben-Haim, Z., Hassidim, A., Matias, Y., 2022. Flood forecasting with machine learning models in an operational framework. *Hydrology and Earth System Sciences*. 26 (15), 4013–4032.
- Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks. In: *International Conference on Machine Learning* (pp. 1310–1318).
- Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., 2019. Deep learning and process understanding for data-driven Earth system science. *Nature*. 566, 195–204. <https://doi.org/10.1038/s41586-019-09121>.
- Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T., 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*. 36, 1181–1191. <https://doi.org/10.1016/j.ijforecast.2019.07.001>.
- Shrestha, A., Bhattacharjee, L., Baral, S., Thakur, B., Joshi, N., Kalra, A., Gupta, R., 2020. Understanding suitability of MIKE 21 and HEC-RAS for 2d floodplain modeling. In: *World Environmental and Water Resources Congress 2020: Hydraulics, Waterways, and Water Distribution Systems Analysis* (pp. 237–253).
- Singh, S.P., Kumar, A., Darbari, H., Singh, L., Rastogi, A., Jain, S., 2017. Machine translation using deep learning: An overview. In: *2017 International Conference on Computer, Communications and Electronics* (pp. 162–167).
- Srivastava, R.K., Greff, K., Schmidhuber, J., 2015. Highway networks. arXiv preprint arXiv:1505.00387. doi: 10.48550/arXiv.1505.00387.
- Tang, X., Machimura, T., Liu, W., Li, J., Hong, H., 2021. A novel index to evaluate discretization methods: A case study of flood susceptibility assessment based on random forest. *Geoscience Frontiers*. 12 (6), 101253.
- Wang, Y., Wang, H., Lei, X., Jiang, Y., Song, X., 2011. Flood simulation using parallel genetic algorithm integrated wavelet neural networks. *Neurocomputing*. 74, 2734–2744. <https://doi.org/10.1016/j.neucom.2011.03.018>.
- Wang, J., Zou, Y., Lei, P., Sherratt, R.S., Wang, L., 2020. Research on crack opening prediction of concrete dam based on recurrent neural network. *Journal of Internet Technology*. 21, 1161–1169. <https://doi.org/10.3966/160792642020072104023>.
- Wang, J., Zou, Y., Lim, S.J., 2023. An improved time feedforward connections recurrent neural networks. *Intelligent Automation & Soft Computing*. 36, 2743–2755. <https://doi.org/10.32604/iasc.2023.033869>.
- Wen, L., Macdonald, R., Morrison, T., Hameed, T., Saintilan, N., Ling, J., 2013. From hydrodynamic to hydrological modelling: Investigating long-term hydrological regimes of key wetlands in the Macquarie Marshes, a semi-arid lowland floodplain in Australia. *Journal of Hydrology*. 500, 45–61. <https://doi.org/10.1016/j.jhydrol.2013.07.015>.
- Yaseen, Z.M., El Shafie, A., Jaafar, O., Afan, H.A., Sayl, K.N., 2015. Artificial intelligence based models for stream-flow forecasting: 2000–2015. *Journal of Hydrology*. 530, 829–844. <https://doi.org/10.1016/j.jhydrol.2015.10.038>.
- Yu, P.S., Chen, S.T., Chang, I.F., 2006. Support vector regression for real-time flood stage forecasting. *Journal of Hydrology*. 328, 704–716. <https://doi.org/10.1016/j.jhydrol.2006.01.021>.
- Yuan, X., Chen, C., Lei, X., Yuan, Y., Muhammad Adnan, R., 2018. Monthly runoff forecasting based on LSTM–ALO model. *Stochastic Environmental Research and Risk Assessment*. 32, 2199–2212. <https://doi.org/10.1007/s00477-018-1560-y>.
- Yue, B., Fu, J., Liang, J., 2018. Residual recurrent neural networks for learning sequential representations. *Information*. 9, 56. <https://doi.org/10.3390/info9030056>.
- Zahura, F.T., Goodall, J.L., Sadler, J.M., Shen, Y., Morsy, M.M., Behl, M., 2020. Training machine learning surrogate models from a high-fidelity physics-based model: Application for real-time street-scale flood prediction in an urban coastal community. *Water Resources Research*. 56, e2019WR027038. doi: 10.1029/2019WR027038.