

Application of temporal convolutional network for flood forecasting

Yuanhao Xu^{a,b}, Caihong Hu^{a,b,*}, Qiang Wu^{a,b}, Zhichao Li^{a,b}, Shengqi Jian^{a,b} and Youqian Chen^{a,b}

^a School of Water Conservancy Science and Engineering, Zhengzhou University, Zhengzhou 450000, China

^b Yellow River Institute for Ecological Protection & Regional Coordinated Development, Zhengzhou 450001, China

*Corresponding author. E-mail: hucaihong@zzu.edu.cn

ABSTRACT

Rainfall–runoff modeling is a complex nonlinear time-series problem in the field of hydrology. Various methods, such as physical-driven and data-driven models, have been developed to study the highly random rainfall–runoff process. In the past 2 years, with the advancement of computing hardware resources and algorithms, deep-learning methods, such as temporal convolutional network (TCN), have been shown to be good prospects in time-series prediction tasks. The aim of this study is to develop a prediction model based on TCN structure to simulate the hourly rainfall–runoff relationship. We use two datasets in the Jingle and Kuye watersheds to test the model under different network structures and compare with the other four models. The results show that the TCN model outperforms the Excess Infiltration and Excess Storage Model (EIESM), artificial neural network, and long short-term memory and improves the flood forecasting accuracy at different foreseeable periods. It is shown that the TCN has a faster convergence rate and is an effective method for hydrological forecasting.

Key words: deep learning, flood forecasting, rainfall–runoff, temporal convolutional network

HIGHLIGHTS

- Propose a TCN model for flood simulation and forecast.
- Explore the optimal hyperparameter combination of TCN at different forecast periods in flood forecast.
- Compare the performance of the models including physical and neural network models.
- Apply the TCN model in different watersheds to evaluate the practicality.

ABBREVIATIONS

TCN	Temporal convolutional network
LSTM	Long short-term memory
ANN	Artificial neural network
NSE	Nash–Sutcliffe efficiency
RMSE	Root-mean-square error

INTRODUCTION

Floods endanger human lives, hinder sustainable socioeconomic development, and cause inestimable damage to densely populated areas in floodplains or downstream from major rivers (Hu *et al.* 2018). Precise flood forecasting can better reduce the risk of flooding and provide timely and efficient environmental information for management decisions (Le *et al.* 2019; Sahoo *et al.* 2019).

In the past decades, multiple flood forecasting models have been proposed. According to the difference in principle, models for flood forecasting can be divided into process-driven models based on physical mechanisms and data-driven models based on machine learning methods (Douglas-Mankin *et al.* 2010; Qin *et al.* 2018; Yuan *et al.* 2018). Commonly used process-driven models, such as the Xinanjiang model, have been regarded as common techniques for flood process simulation and forecasting (Beven *et al.* 1984; Zhao 1992; Wang *et al.* 2012). They usually require complex mathematical formulas, a large amount of hydrological and meteorological data, and an accurate understanding of runoff mechanisms. However, with the diversification of hydrological data and in-depth research on the mechanism of runoff generation and convergence, these process-driven models are restricted by many factors (Kratzert *et al.* 2018; Tian *et al.* 2018).

This is an Open Access article distributed under the terms of the Creative Commons Attribution Licence (CC BY 4.0), which permits copying, adaptation and redistribution, provided the original work is properly cited (<http://creativecommons.org/licenses/by/4.0/>).

The data-driven models focus on the statistical relationship between input and output data and do not consider the physical mechanism of the hydrological process, but rather establish a mathematical analysis of the time series and use the given sample to discover the statistical or causal relationship between the hydrological variables (Adamowski & Sun 2010; Yunpeng *et al.* 2017). The data-driven method has unique advantages for solving numerical prediction problems, reconstructing highly nonlinear functions, and analyzing time series (Miao *et al.* 2019).

With the rapid development of computer technology, data-driven models have gained more applications in the field of hydrological forecasting (Bafitlhile & Li 2019). The artificial neural networks (ANNs), a common data-driven model based on artificial intelligence, were applied in flood forecasting at the end of the last century already (Hsu *et al.* 1995; Shamseldin 1997; Abdellatif *et al.* 2013).

Recent breakthroughs in the field of computational science have created growing interest in learning methods in academic and applied scientific circles (Chang *et al.* 2002; Kisi 2011). One of the most active research points in deep learning is modeling sequential data through a recurrent neural network (RNN), which is particularly suitable for hydrological prediction and gives a precise and timely prediction of time series in systems (Chang *et al.* 2002; Kumar *et al.* 2004).

Since the late 1990s, more modern RNN architectures have been proposed, particularly the long short-term memory (LSTM), which has been successfully applied to image recognition, the Internet of Things, text translation, and stock prediction (Hochreiter & Schmidhuber 1997; Donahue *et al.* 2015; Crivellari & Beinat 2020; Wu *et al.* 2020). While the LSTM neural network is designed to overcome the problem of gradient disappearance in long-term dependent time series of simple RNN neural network, each LSTM unit contains four affine transformations and each time step needs to be run once, which can easily use up the memory available.

However, the development of deep learning is changing with each passing day, and there are still some networks worthy of applying for hydrological modeling and flood forecasting (Gao *et al.* 2020; Lin *et al.* 2020). Although CNNs were originally designed for computer vision tasks, they are also suitable for time-series data since they can extract high-level features from data with a grid topology. Temporal convolutional network (TCN) is a recently proposed convolutional neural network, which combines the 1-dimensional fully convolutional network (1D FCN) and causal convolutions (Bai *et al.* 2018). The 1D FCN keeps the network producing an output of the same length as the input. The causal convolution guarantees that future information will not affect the past. The dilated convolutions and residual modules in TCN can obtain a large receptive field by a few convolutional network layers and allows the network to transmit information in a cross-layer manner (Bai *et al.* 2018). Recently, TCN has been proved superior for time-series data modeling tasks to LSTM (Chen *et al.* 2020). Several works have already successfully used TCNs for time-series forecasting tasks such as the stock trend prediction, anomaly detection, and recognition of sepsis (Deng *et al.* 2019; Yujie *et al.* 2019; Lara-Benitez *et al.* 2020). In the field of hydrology, especially the prediction of rainfall-runoff relationship contains a large number of time-series data, resulting in an urgent demand for modeling. Up to now, people have not applied this new technology to flood forecasting, and more advanced deep-learning technologies such as TCN can provide support for more efficient and accurate forecasting. Therefore, TCN is introduced first in flood forecasting based on deep learning.

The objective of this study is to explore the ability and stability of TCN for flood forecasting with multi-step lead times. To achieve this goal, we process the rainfall, flow, and NDVI data of the Jingle watershed to a size between (0, 1) through normalization to facilitate neural network analysis. Next, TCN is used to predict the future 1–12 h flow process. Additionally, we use the Nash–Sutcliffe efficiency (NSE), root-mean-square error (RMSE), and bias to evaluate the performance of the prediction model and to compare the results with those of EIESM, ANN, and LSTM to experimentally verify the model effectiveness (Shiri *et al.* 2015; Wen *et al.* 2020). In addition, in order to explore the influence of NDVI data as input data on the TCN model, TCN without NDVI is also used as a comparison model. Finally, we also conducted experiments on these models in another basin to prove the stability of TCN in dealing with flood forecasting tasks.

The following section introduces the TCN model structure, model design, settings and parameterization, comparison models, and performance evaluation criteria. The case study, hydrological data, and the data preprocessing method are given in the ‘Case study’ section. The experimental results and thorough discussion are presented in the ‘Result and discussion’ section. The conclusions of this study provide a summary of the work and recommendations for future research in the ‘Conclusions’ section.

METHOD

Temporal convolutional neural network

TCN, a type of convolutional neural network, is applied in many time-series forecasting tasks (Liu *et al.* 2019). It is deliberately kept simple, combining some of the best practices of modern convolutional architectures (Bai *et al.* 2018). TCN has two specific designs: 1D FCN architecture is used in TCN to keep the network output the same length as the input sequence (Long *et al.* 2015); The outputs are only influenced by information of the present and past inputs in each layer by using causal convolutions. Causal convolution is different from standard convolution in that the output at time t is not convolved with future values (Wan *et al.* 2019).

However, simple causal convolution still has the problem of traditional convolutional neural networks, and the modeling length of time is limited by the size of the convolution kernel (van den Oord *et al.* 2016). In this case, if you want to learn longer dependencies between data, you need to stack many layers linearly. To solve this problem, TCN uses one-dimensional expansion convolution. The difference between expansion convolution and traditional convolution is that it allows the input of convolution to have interval sampling. Without the pooling operation, this convolution increases the acceptable range of the network, so there is no loss of resolution.

The complete dilated causal convolution operation F over consecutive layers for a 1-D sequence of a given hydrology input $q = (q_0^1, \dots, q_t^1)$ and a filter $f: \{0, \dots, k-1\}$, on element s of the sequence, is defined by the following equation:

$$F(s) = (q *_{df})(s) = \sum_{i=0}^{k-1} f(i) \cdot q_{s-di} \quad (1)$$

where d is the dilation factor, k is the filter size, and $s - d \cdot i$ accounts for the direction of the past.

As shown in Figure 1, the sampling standard is selected according to d , the first layer $d=1$ means that each number is selected, and the second layer $d=2$ means that the data are selected at intervals (Lara-Benitez *et al.* 2020). Choosing a larger filter size k or increasing the expansion factor d can obtain a larger network receptive field. Similar to the common dilated convolutions, d increases exponentially with the depth of the network layers, which allows the network to use a larger effective history while receiving each input. Another common method to further increase the network acceptance domain is to connect several TCN blocks. To avoid the deeper network structure from complicating the learning process,

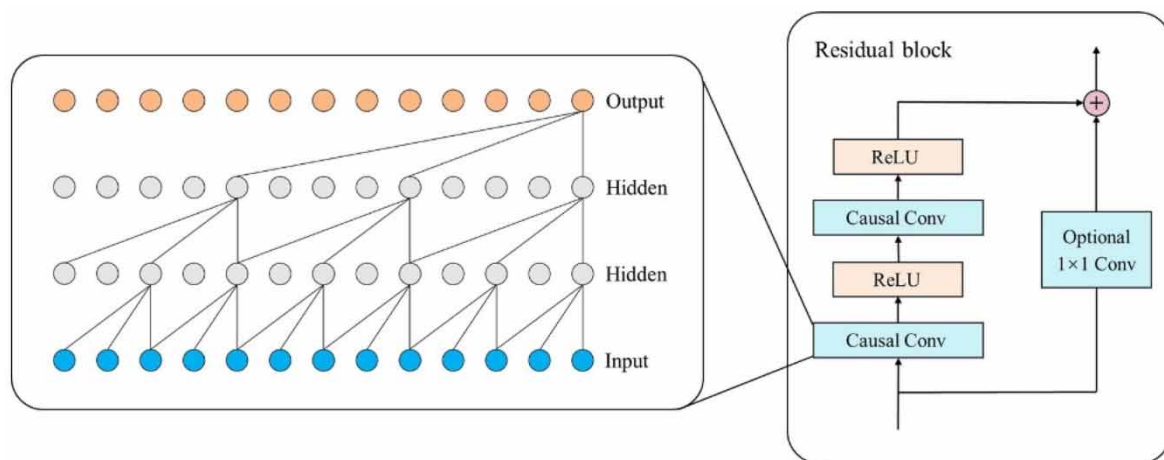


Figure 1 | TCN architectural elements, including a dilated causal convolution with dilation factors of $d = 1, 2$, and 4 and filter size of $k = 3$. The receptive field is able to cover all values from the input sequence. And a 1×1 convolution (residual block) is added when residual input and output have different dimensions. The sampling standard is selected according to d , the first layer $d=1$ means that each number is selected, and the second layer $d=2$ means that the data are selected at intervals (Lara-Benitez *et al.* 2020). Choosing a larger filter size k or increasing the expansion factor d can obtain a larger network receptive field. To avoid the deeper network structure from complicating the learning process, a residual connection is added to the output (He *et al.* 2016). The residual connection uses 1×1 convolution to ensure that the addition operation receives the same tensor (Bai *et al.* 2018).

a residual connection is added to the output (He *et al.* 2016). Since the input and output have different widths, the residual connection uses 1×1 convolution to ensure that the addition operation receives the same tensor (Bai *et al.* 2018).

Flood forecasting model based on TCN

TCN is designed for time-series forecasting tasks. The rainfall–runoff process contains a series of time-series data, such as rainfall evapotranspiration and runoff. To explore the application of TCN in flood forecasting, this paper established a multi-step time-series forecasting model based on TCN for flood forecasting in the next 1–12 h. In TCN, a moving window scheme is used to create input and output pairs, which will be fed into the neural network (Bandara *et al.* 2020). In the prediction process at time t , the input data from time $t - n$ to t are used to simulate the value at time $t + m$, where n is the length of the input data, and m is the prediction time in the future. When the window slides to the next moment (time $t + 1$), the value at time $t + m + 1$ is simulated using the data at time $t - n + 1$ to $t + 1$.

In this paper, the factors that affect the process of runoff generation and convergence, such as rainfall, evapotranspiration, and runoff data in the watershed, are selected as the input of the model. In addition, since the middle reaches of the Yellow River where the study area is located, the nature of the underlying surface changes significantly within the data time span, which affects the rainfall–runoff process in the basin. As a piece of data that can reflect the characteristics of the underlying surface, NDVI plays an indispensable role in simulating the relationship between rainfall and runoff. At the same time, deep learning requires a great deal of data as support, and only rainfall and flow data for input are not comprehensive. Although the relationship between rainfall and flow is the most obvious and direct, it is difficult for the model to grasp the subtle differences in the flood process. Moreover, predecessors have done similar studies. Therefore, we consider NDVI as another input to enhance the TCN's ability to learn data; flood flow data that needs to be predicted are used as output. These data constitute the dataset for training and testing the model.

Model setting and parameterization

The programming language of choice is Python 3.7, and the libraries used for preprocessing and managing our data are NumPy and pandas. We use the Google Keras deep-learning framework with TensorFlow backend and the NVIDIA RTX 2080Ti GPU to train the models.

First, the normalization operation is performed. Normalization is a method of rescaling the data in the original range so that all values range from 0 to 1. If the machine learning model uses the gradient descent method to find the optimal solution, dimensions are not uniform, and normalization is often necessary to allow convergence. For efficient learning, all input features (meteorological and hydrological variables) and output data (the discharge) are normalized by subtracting the mean value and dividing by the standard deviation. The output of the network is retransformed to obtain the final discharge prediction, which is an inverse data scaling process. The expression is described as follows:

$$x' = \frac{x - \mu}{\sigma} \quad (2)$$

where x' and x represent the normalized result and sample data, respectively; μ and σ are the mean and standard deviation of the sample data, respectively.

The purpose of this research is to establish a TCN-based deep-learning flood forecasting model to fully explore the ability of TCN to perceive hydrological data. Due to the high complexity of deep-learning models, finding the optimal TCN network structure and hyperparameters is a crucial task. Therefore, under the premise of different foreseeable periods, we conducted multiple experiments on the same dataset and combined them into models containing different network structures by changing the value of the TCN hyperparameters. The kernel size, filter, and residual block are selected as {4, 6, 8}, {32, 64, 128}, and {1, 2, 3}, respectively. In addition, the expansion factor is fixed to [1, 2, 4, 8, 16, 32]. And the length of the input data is fixed to six, which means that a total of six time periods of current and past data are used to predict the future value. All these architectures are then tested for all combinations of the following parameters: the batch size and the epochs are selected as {32, 64, 128} and {20, 50, 100}, respectively. In general, we conducted $(3 \times 3 \times 3 \times 3 \times 3 = 243)$ experiments on the dataset in the research basin and found the best network structure and hyperparameter values under different forecast periods.

Performance evaluation criteria

In this study, the performance of different models is evaluated by statistical error measures, including the RMSE, NSE, and bias (Nash & Sutcliffe 1970; Hu *et al.* 2018; Kratzert *et al.* 2018). The mathematical expressions of these metrics can be described as follows:

$$\text{NSE} = 1 - \frac{\sum_{i=1}^n (Q_0 - Q_c)^2}{\sum_{i=1}^n (Q_0 - \overline{Q_0})^2} \quad (3)$$

where Q_0 (m^3/s) and Q_c (m^3/s) represent the discharge of the observed and simulated hydrographs, respectively; $\overline{Q_0}$ is the mean value of the observed discharge, and n is the data point number.

NSE measures the ability of the model to predict variables different from the mean, gives the proportion of the initial variance accounted for by the model, and ranges from 1 (perfect fit) to $-\infty$. Values closer to 1 provide more accurate predictions.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (Q_0 - Q_c)^2}{n}} \quad (4)$$

The RMSE is very sensitive to the maximum and minimum errors, enabling it to effectively reflect the accuracy of the prediction results. RMSE values closer to 0 provide more accurate predictions.

$$\text{Bias} = \frac{\sum_{i=1}^n (Q_0 - Q_c)}{\sum_{i=1}^n (Q_0)} \quad (5)$$

The bias can evaluate the accuracy of the overall water balance of the simulation results and range from -100 to 100% . A value close to 0 means more accurate predictions.

Model benchmarks and methods

We used several benchmarks to evaluate the performance of the TCN model. These benchmarks include LSTM, ANN, and EIESM (physical model). In addition, in order to explore the influence of NDVI as an input on the TCN model, we also added the TCN model without NDVI as a benchmark. The forecast period was set to 1–12 h, which is the premise of comparing the performance of different benchmarks. We simultaneously calculate and compare the performance evaluation indicators of all benchmarks.

LSTM is an RNN with a special memory unit structure (Hochreiter & Schmidhuber 1997). In the memory cell unit, the input information passes through the forget gate, input gate, and output gate in turn. Some information is selected to be forgotten, while other information is selected to be added to the memory, which overcomes the disadvantages of gradient explosion and gradient disappearance of traditional recurrent neural networks. Research in recent years has shown that LSTM is widely used in the modeling of time-series forecasting tasks in the hydrological field, such as flow forecasting and water quality forecasting assessment (Kratzert *et al.* 2018). LSTM was used as a benchmark in this study.

ANN is a black-box simplified model used to solve several water resources problems and can be trained with datasets to identify complex nonlinear relationships between inputs and outputs (Tokar & Johnson 1999). The representative feedforward neural network is composed of an input layer, hidden (containing neurons), and output layers. Recent studies have shown that using ANN is one of the most significant methods to simulate hydrological processes (Ahmad & Hussain 2019). ANN was also used as another benchmark in this study.

EIESM is a physical, improved Xinanjiang model (Hu *et al.* 2003). Compared with the Xinanjiang model, the excess infiltration runoff mode of EIESM is based on the infiltration curve and infiltration capacity distribution curve of the watershed, and the storage runoff mode is based on the water storage capacity distribution curve of the watershed. The two types of runoff modes are organically combined. Recent studies have applied the runoff simulated by EIESM within the range of

acceptable accuracy, which is reflected by the goodness-of-fit measure (Wen *et al.* 2020). EIESM was used as a benchmark in this study.

CASE STUDY

The Yellow River – the fifth largest in the world – often suffers from flood disasters. In recent years, a large number of water and soil conservation measures have been applied in the middle reaches of the Yellow River. This study selected the representative Jingle and Guxian watersheds. The first TCN model was developed for the Jingle watershed of the Fenhe River in Shanxi Province, a relatively small watershed that covers 2,799 km². Jingle hydrological station is the primary stream control station on the upper Fenhe River and is located at 111°55' east longitude and 38°20' north latitude. The annual mean precipitation in the Jingle watershed is approximately 538.38 mm. Devastating frequent flooding in the last few decades has been widely researched.

An additional assessment was conducted in the Kuye watershed to discover if the proposed model architecture operates in different watersheds after training. The watershed covers 8,706 km², spans Shanxi and Henan provinces, and is narrow with a long concentration time. The Wenjiachuan station is located at 110°45' east longitude and 38°26' north latitude. Annual precipitation in the two watersheds varies greatly, and both are severely affected by flooding. Figure 2 shows the locations of the Jingle and Kuye watersheds.

The underlying data for our study of the Jingle watershed include hourly discharge data from the Jingle station and hourly rainfall data from 14 gauges in the area. Complete records of 98 flood events from 1971 to 2013 were obtained. Of these, 78 flood events (1971–2000, 3,986 datasets) were used for calibration, and 20 events (2000–2013, 1,366 datasets) were used for verification. The type of data used in the Kuye watershed is the same as that of Jingle, but the Kuye watershed contains 19 rainfall stations and the data contain 86 events (1973–2016, 3,479 datasets), of which 66 events (1973–2003, 2,764 datasets)

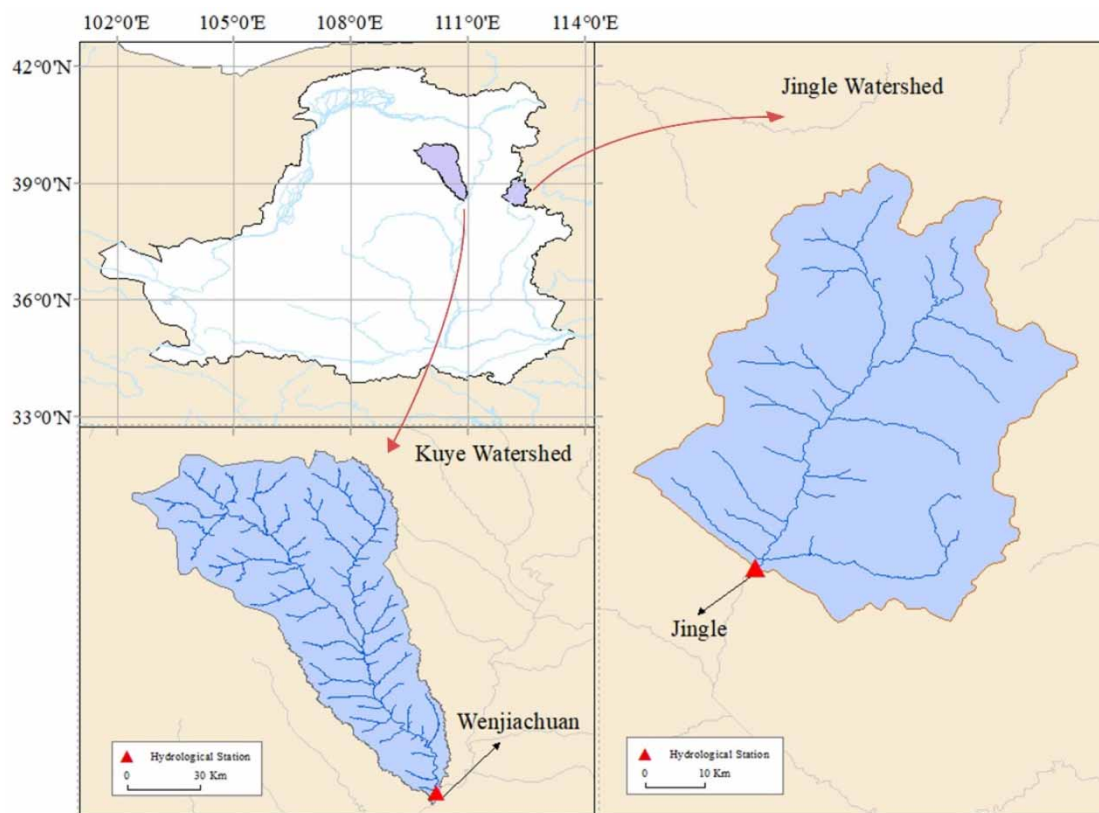


Figure 2 | Two watersheds and their locations in the provinces (top left). The detailed streamflow map of the Kuye watershed (bottom left) and the Jingle watershed (right) with the sub-watershed boundary (red lines) and hydrological stations (red markers). Please refer to the online version of this paper to see this figure in color: <http://dx.doi.org/10.2166/nh.2021.021>.

are used as the training set and 20 events (2003–2016, 3,479 datasets) as the validation set. In this paper, a typical flood process, with a large volume flow and duration, is selected to verify the performance of the established model.

RESULTS AND DISCUSSION

TCN network hyperparameter experiment

In this study, after experiments with different combinations of TCN models on the Jingle watershed dataset, we finally selected the best TCN model construction scheme based on the loss function (Loss) of the prediction results on the verification set for different forecast periods, including kernel size, filters, residual blocks, batch size, and epochs. Table 1 illustrates the best model construction in each forecast period.

As can be seen from Table 1, there is no obvious difference between the constructions of different forecast periods. The filters and kernel size are mostly selected as 128 and 8, respectively. The number of residual blocks always works best when the value is 2. These results are related to the characteristics of the given input hydrological data. TCN adjusts the receptive field by changing the above three hyperparameters, and the internal learning matches the rainfall–runoff process. The batch size remains unchanged at 128, and the epochs increase with the forecast periods. As the increases of time interval between output data and input data, the convolutional network requires more iterative epochs to capture the relationship in the data.

Understanding TCN in hydrology with model evaluations

In the above research, we discussed the optimal network structure and hyperparameters for TCN to model the rainfall–runoff process. To deeply explore and compare the hydrological process simulation performance of the TCN model, we used the other four models, including TCN (without NDVI input), LSTM, ANN, and EIESM models, for the same experiment and performed a simulation at a forecast period of 1–12 h as TCN. Table 2 illustrates the evaluation index of runoff forecasting at different forecast periods (1–12 h) by the four models. The changes in the three indicators reflect the flood simulation accuracy on the training and testing set. At the same time, in order to more intuitively reflect the performance of the model on the test set, we also calculate the evaluation indicators of all flood events on the test set. Figure 3 shows the Boxplots of different tests. It can be seen that the results of all models are closely related to the forecast periods, and the prediction accuracy decreases with the increase of the forecast period. In the modeling process, the forecast period represents the time interval between input and output data. A longer forecast period increases the difficulty of the prediction of the target value. In terms of flow formation reasons, future flow will be affected by current or earlier rainfall and other factors. The existing data gradually cannot provide an effective reference when it is hoped to obtain more distant future flow.

In the physical model, the results show that the NSEs of EIESM are 0.8848 (training set) and 0.8610 (testing set) at a forecast period of 1 h but fall to 0.7203 (training set) and 0.7106 (testing set) at a forecast period of 12 h. The physical models use the principle of rainfall–runoff to calculate the flow, but the overall accuracy is low due to incomplete consideration and systematic error. The results of ANN show that NSE varies from 0.9 to 0.6 for forecast periods from 1 to 12 h. The simulation effect shows a rapid downward trend with the increase of the foreseeable period. ANN is a relatively simple ANN and uses a backpropagation supervised learning technique for training. As a machine learning model proposed earlier, ANN cannot capture the information in the input required to process sequence data. Compared with the previous two models, LSTM, TCN, and TCN (NDVI) are well simulated and meet the needs of flood forecasting. It is evident that the forecastability of two TCN models is higher than that of LSTM at almost each forecast period, especially for long periods close to 6 h. The performance

Table 1 | The best TCN model settings under different forecast periods

Forecast periods (h)	Kernel size	Residual block	Filters	Batch size	Epochs	Loss
1	8	2	128	128	20	0.0023
2	8	2	64	128	20	0.0035
3	6	2	128	64	50	0.0058
6	8	2	128	128	50	0.0126
9	8	2	64	128	50	0.0249
12	8	2	128	128	50	0.0401

Table 2 | Performance of the five models in the Jingle watershed on the calibration set and validation set

Forecast period (h)	Dataset	Model	NSE	RMSE	Bias (%)
1	Calibration	EIESM	0.8848	46.7094	14.2147
		ANN	0.9536	9.8926	1.3384
		LSTM	0.9836	8.8401	0.6991
		TCN	0.9858	8.4564	0.5341
		TCN (NDVI)	0.9897	7.9532	0.5119
	Validation	EIESM	0.8610	51.9966	14.4069
		ANN	0.9477	11.9942	1.4488
		LSTM	0.9704	9.1252	0.7197
		TCN	0.9825	8.7127	0.5458
		TCN (NDVI)	0.9838	8.1511	0.5176
6	Calibration	EIESM	0.7665	60.4484	26.0965
		ANN	0.8120	53.4280	19.1487
		LSTM	0.8859	45.7676	14.7178
		TCN	0.9061	40.4808	10.6355
		TCN (NDVI)	0.9109	38.8848	8.9400
	Validation	EIESM	0.7588	61.1700	27.1619
		ANN	0.8055	55.5501	20.3971
		LSTM	0.8721	47.7827	14.8497
		TCN	0.8963	43.2517	11.7935
		TCN (NDVI)	0.9031	40.9977	9.9135
12	Calibration	EIESM	0.7203	86.6261	43.0147
		ANN	0.6761	123.2829	52.4860
		LSTM	0.7354	83.3445	40.4083
		TCN	0.7790	75.9966	31.2881
		TCN (NDVI)	0.7983	73.3085	26.1076
	Validation	EIESM	0.7106	91.4497	45.7214
		ANN	0.6800	132.6323	58.1628
		LSTM	0.7185	88.1648	34.6354
		TCN	0.7574	80.0811	29.1310
		TCN (NDVI)	0.7729	77.2392	27.5773

of the three models declined as forecast periods increased. For forecast periods of less than 6 h, the NSE of TCNs is higher than that of LSTM, and RMSE and bias are lower. When the forecast periods exceed 6 h, the prediction accuracy of LSTM sharply decreases, whereas the NSE of TCNs remains above 0.7; RMSE and bias remain below 100 and 40%, respectively. Finally, for the two different input TCN models, the latter model with NDVI input is better simulated, with higher NSE, lower RMSE and bias in all forecast periods. The input including NDVI better reflects the true characteristics of the watershed, so that TCN can learn the relationship between data more fully. At the same time, deep learning requires a large amount of data as support, so adding the NDVI sequence is beneficial for TCN fitting. Among five models, the proposed TCN model shows the highest accuracy compared to normal machine learning and physical models.

To evaluate the ability of the TCN model to forecast the flood process, of which the first three are the large flood events in the verification set, and the last one is the smaller and most common flood during the verification set. We use the previously mentioned model containing the best TCN structure and training hyperparameters to simulate the flood process. Figure 4 shows the observed and estimated hydrographic map of four flood events during the forecast period of 1, 6, and 12 h.

Flood events 1 and 2 were considered abnormal events during the verification set. The rainfall that formed the flood was high in intensity and short in duration, and the peak shape was sharp. At the forecast period of 1 h, the predicted value of the EIESM model for the peak and low tide section is higher than the actual value. ANN fluctuates significantly, especially before the flood peak occurs. However, the forecast runoff curves of LSTM and TCNs both fit the observed runoff curve well. They have strong predictive capabilities of backwater stages in good agreement with the actual process. In contrast, the flood peaks forecasted by TCNs are more realistic than those of LSTM, indicating that TCNs are more sensitive to rainfall and runoff processes. Although there is no obvious difference between TCN and the proposed TCN (NDVI), the latter fits the true value more accurately in the preceding section. At the forecast period of 6 h, the performance of all models has a certain

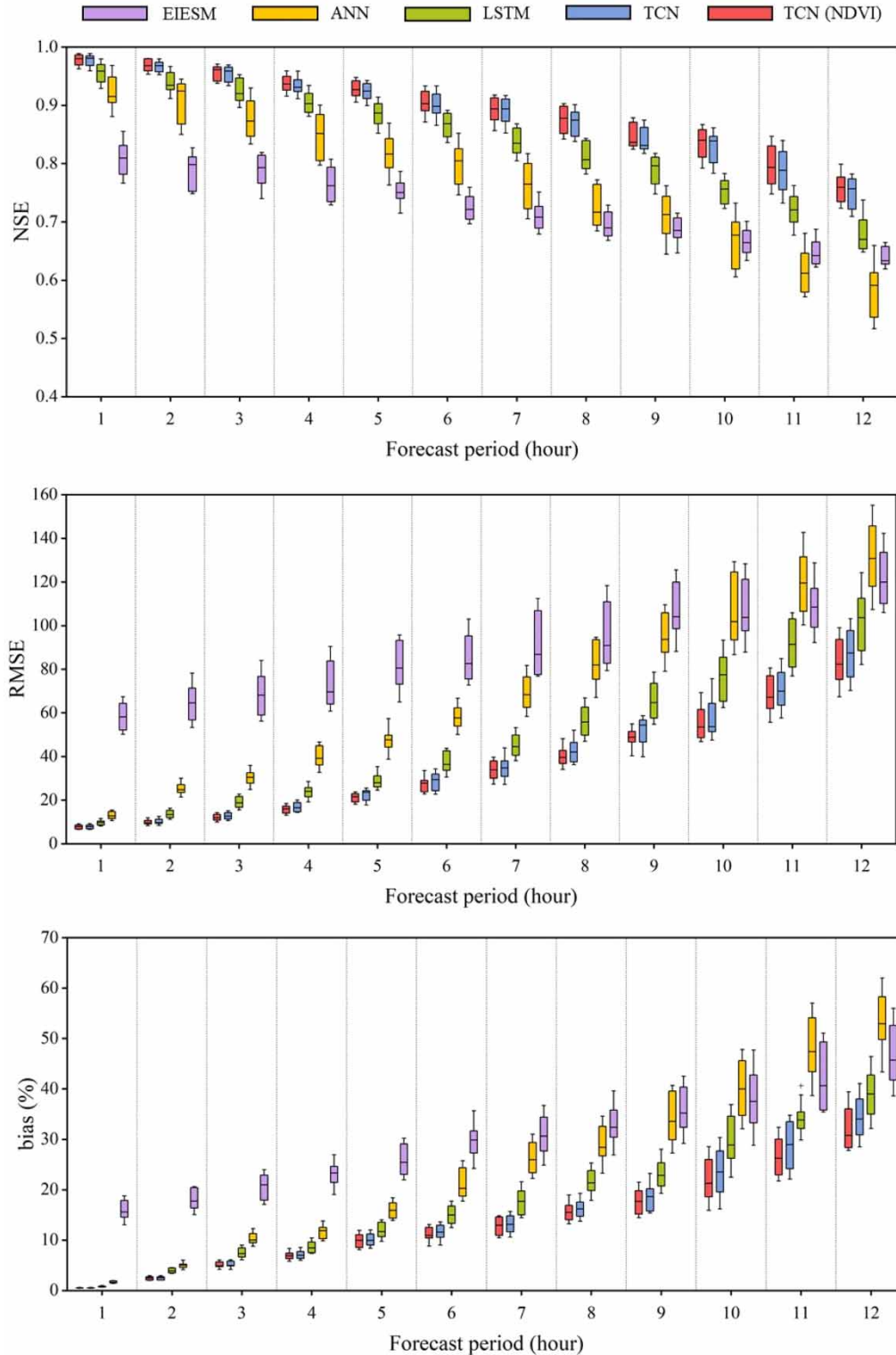


Figure 3 | Boxplots of different tests. Each box is calculated from four realizations of model runs. It shows the average NSE for the prediction with different forecast periods of 1, 6, and 12 h. It can be seen that the results of all models are closely related to the forecast periods, and the prediction accuracy decreases with the increase of the forecast period. Among five models, the proposed TCN model shows the highest accuracy compared to normal machine learning and physical models.

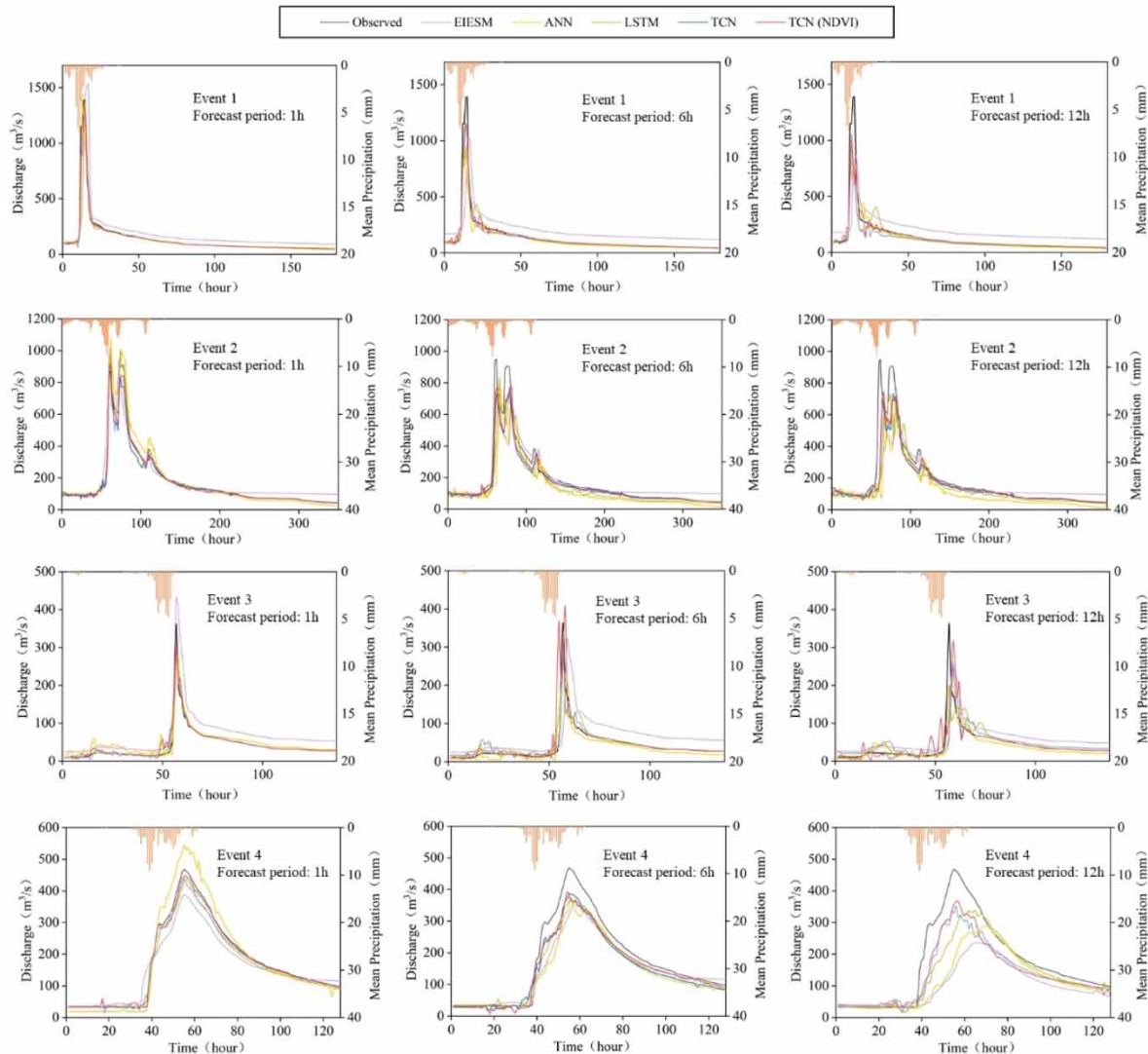


Figure 4 | Observed and estimated hydrographs of the five models at the validation stage (flood events 1, 2, 3, and 4) with different lead times of 1, 6, and 12 h. Flood events 1 and 2 were considered abnormal events during the verification set. And the last two floods have a lower peak flow than the first two events, which are normal flood events. The results show that TCN (NDVI) best predicted the hourly peak flow, whereas the other models were insufficient to predict the values and had lower forecast accuracy.

degree of degradation. Although EIESM expresses the correct flood process trend, the overall accuracy is insufficient. Moreover, the forecasting ability of ANN and LSTM deteriorates significantly, the phenomenon of underestimating flood peaks increase, and the simulated values fluctuate abnormally compared with observed values. ANN has large fluctuations and abnormal values, which may be due to the insufficient ability of ANN to learn long-term data. The flood peak flow forecasted by LSTM is later than the observed flood peak flow, which will seriously affect the flood warning. It is obvious that TCNs better simulate rainfall-runoff and forecast floods well and have higher accuracy than other models. Between the two TCN models, the model with NDVI input has a better grasp of where the flood peak appears and has less fluctuation. When the forecast period exceeds 6 h, due to the lack of hydrological data to form future runoff, the forecast flow curves of all models are much later than future observations. Even so, TCNs are minimally affected and the results are still practical. The results also prove that considering NDVI as an input can effectively improve forecast accuracy.

The last two floods have a lower peak flow than the first two events, which are normal flood events. Both figures show that the overall performance of EIESM is stable and insensitive to forecast period factors. Conversely, changes in the forecast period are likely to cause fluctuations in the neural network models. TCN (NDVI) best predicted the hourly peak flow, whereas the other models were insufficient to predict the values and had lower forecast accuracy.

Model application in a different watershed

To evaluate the practicality of the model structure, we applied the established TCN model to the Kuye watershed and selected the same length data as the previous experiment. The Kuye watershed is larger and narrower than the Jingle watershed and has longer travel time and different topography, soil type, and land use. TCN network modeling used hyperparameter combinations obtained from repeated experiments: the filter, kernel size, and residual blocks are selected as 128, 8, and 2, respectively, and the expansion factor is [1, 2, 4, 8, 16]. Table 3 shows the results based on different forecast periods for four models. In the verification set, NSE of the EIESM, ANN, LSTM, TCN, and TCN (NDVI) models are 0.8416, 0.9463, 0.9787, 0.9798, and 0.9819 at a forecast period of 1 h, respectively. With extended forecast periods, the simulation accuracy of all the models declined to varying degrees. Among them, EIESM and ANN have the most obvious downward trend. Although LSTM maintains a high level of prediction, it is always lower than TCNs. The performance of TCN without NDVI input still lags behind TCN (NDVI). The final TCN (NDVI) model consistently outperformed the other models. These results are the same as those in the Jingle watershed, confirming that the TCN (NDVI) model established in this study responds relatively smoothly to disturbed watershed attributes and can be used to make accurate predictions in multiple watersheds.

CONCLUSIONS

At present, more and more deep neural network methods are applied to rainfall–runoff prediction. On the one hand, with the advancement of technology, the means of obtaining data are more intelligent, and the types of data available are more diverse, such as soil evapotranspiration, wind speed, and pressure in the watershed. Some data that cannot be directly observed can be

Table 3 | Performance of the five models in the Kuye watershed on the calibration set and validation set

Forecast period (h)	Dataset	Model	NSE	RMSE	Bias (%)
1	Calibration	EIESM	0.8667	49.1452	14.9225
		ANN	0.9544	10.3626	1.4078
		LSTM	0.9802	9.2594	0.7345
		TCN	0.9848	8.8991	0.5612
		TCN (NDVI)	0.9874	8.3301	0.5364
	Validation	EIESM	0.8416	54.5497	15.1129
		ANN	0.9463	12.5921	1.5247
		LSTM	0.9766	9.5869	0.7542
		TCN	0.9798	9.1584	0.5716
		TCN (NDVI)	0.9819	8.5426	0.5447
6	Calibration	EIESM	0.7504	63.5939	27.4180
		ANN	0.7929	56.1515	20.0869
		LSTM	0.8665	48.1196	15.4171
		TCN	0.8876	42.4766	11.1822
		TCN (NDVI)	0.8915	40.8133	9.4066
	Validation	EIESM	0.7413	64.2193	27.4510
		ANN	0.7869	56.3258	20.3226
		LSTM	0.8512	48.1237	15.5833
		TCN	0.8771	43.1984	11.3480
		TCN (NDVI)	0.8847	40.9412	9.5563
12	Calibration	EIESM	0.7036	91.0993	45.1899
		ANN	0.6621	129.4762	54.9864
		LSTM	0.7194	87.4816	42.0057
		TCN	0.7590	79.7766	34.7225
		TCN (NDVI)	0.7811	76.8033	27.3611
	Validation	EIESM	0.6964	96.1865	48.0235
		ANN	0.6641	127.6674	52.6108
		LSTM	0.7035	92.4926	39.3347
		TCN	0.7611	83.8810	30.6066
		TCN (NDVI)	0.7649	81.1183	28.9558

obtained by the inversion of remote sensing images. Progress in data requires a framework that can consider multiple factors to interpret the hidden relationships between data. On the other hand, the rainfall–runoff process contains many complicated steps. The runoff mechanism in semi-humid, semi-arid areas is more complicated than in humid areas (Le *et al.* 2019). The physical model is not enough to reflect the complete mechanism. Therefore, the deep neural network has become an effective new way of simulating rainfall and runoff.

As a time-series prediction model based on convolution that has been proposed in the past 2 years, TCN has been successfully applied in some fields. This paper proposes a TCN learning model for rainfall–runoff prediction, which uses one-dimensional convolution operation to process input and output sequences. The TCN model uses the observed rainfall data, evaporation, and NDVI data from rainfall stations in the basin as input, and the outlet section flow as output.

In the process of modeling, some hyperparameters in the structure of TCN neural network need to be considered, such as kernel size, filter size, and residual blocks. Inappropriate hyperparameter combinations will cause the prediction results to deviate from the true value. In each case of the forecast period of 1–12 h, 273 different hyperparameter combinations were tested in the Jingle watershed, and finally, the set of results with the best prediction effect was selected. Several domain characteristics are necessary for a successful application. First, the change of the best combination under different forecast periods is not obvious. The filter and kernel sizes fluctuate between 6, 8 and 64, 128, respectively, and the residual blocks remain at 2. In addition, some of the hyperparameters used for training, such as the batch size, are stable at 128. At the same time, TCN needs more iterations to complete the fitting in response to a longer forecast period. The characteristics of these hyperparameter combinations are considered to be useful for rainfall–runoff simulation.

In this study, we use the TCN network to model two watersheds. The experimental results show that the proposed TCN method can better simulate the rainfall–runoff process, has a better prediction effect than other models, and has less deviation when dealing with long-foreseeing prediction tasks. Traditional physical models and early machine learning models, such as EIESM and ANN, generally have low simulation accuracy. LSTM meets the forecast requirements under the short forecast period, but loses more accuracy with the prolongation of the forecast period, and significantly underestimates flood peaks and abnormal fluctuations. It is worth noting that the TCN model, including NDVI input, has a certain ability to reflect the characteristics of the underlying surface, which leads to better performance in the simulation on the selected watershed. This provides a new idea for the subsequent deep learning for hydrological forecasting tasks: inputting different types of data on the basin that have the potential to affect the flow. At the same time, the simulation in the Kuye watershed showed the stability of the proposed TCN model, and the selected hyperparameter combinations can be applied to a new watershed.

Using TCN to model the rainfall–runoff process can capture the relationship between hydrological data. In TCN, because the filter exists in each layer of the network, convolution can complete the prediction task in parallel, while RNN, such as LSTM, can only process information in one direction, that is, it needs to wait for the forward pass of the previous time step to be completed. The forward pass of the next time step and the deep LSTM network require a larger amount of calculation, resulting in a long time-consuming prediction task. In addition, the adjustable kernel size and expansion coefficient support the receptive field, which enables TCN to handle tasks with varying degrees of complexity. In the experiment, we found that TCN has a faster convergence rate than LSTM and occupies less memory, which is indispensable for prediction. These results also demonstrate the strong potential of applying deep-learning methods to other hydrological problems, specifically other time-series tasks.

The hydrological process contains complex and diverse variables. More variables that affect the target value can be used as input to enhance the deep-learning model's cognition of physical processes in the future, such as soil moisture and wind speed. In addition, deep neural network methods like TCN often have many hyperparameters. At present, we use repeated experiments to determine the best combination of them, which has limitations. If it can be combined with more intelligent parameter optimization methods, it will greatly improve the efficiency of obtaining the desired results.

ACKNOWLEDGEMENTS

The authors acknowledge the financial support received from Projects of National Natural Science Foundation of China (No. 51979250), National Key Research Priorities Program of China (No. 2016YFC040240203), National Key Research Priorities Program of China (No. 2019YFC1510703), Key Projects of National Natural Science Foundation of China (No. 51739009), and Key Research and Promotion Projects (technological development) in Henan Province.

DATA AVAILABILITY STATEMENT

All relevant data are included in the paper or its Supplementary Information.

REFERENCES

- Abdellatif, M., Atherton, W. & Alkhaddar, R. 2013 A hybrid generalised linear and Levenberg-Marquardt artificial neural network approach for downscaling future rainfall in North Western England. *Hydrology Research* **44** (6), 1084–1101. doi:10.2166/nh.2013.045.
- Adamowski, J. & Sun, K. 2010 Development of a coupled wavelet transform and neural network method for flow forecasting of non-perennial rivers in semi-arid watersheds. *Journal of Hydrology* **390** (1–2), 85–91. doi:10.1016/j.jhydrol.2010.06.033.
- Ahmad, S. K. & Hussain, F. 2019 A generic data-driven technique for forecasting of reservoir inflow: application for hydropower maximization. *Environmental Modelling & Software* **119**, 147–165. doi:10.1016/j.envsoft.2019.06.008.
- Bafithile, T. M. & Li, Z. 2019 Applicability of epsilon-support vector machine and artificial neural network for flood forecasting in humid, semi-humid and semi-arid basins in China. *Water* **11** (1). doi:10.3390/w11010085.
- Bai, S., Kolter, J. Z. & Koltun, V. 2018 An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271.
- Bandara, K., Bergmeir, C. & Smyl, S. 2020 Forecasting across time series databases using recurrent neural networks on groups of similar series: a clustering approach. *Expert Systems with Applications* **140**. doi:10.1016/j.eswa.2019.112896.
- Beven, K., Kirkby, M., Schofield, N. & Tagg, A. 1984 Testing a physically-based flood forecasting model (TOPMODEL) for three UK catchments. *Journal of Hydrology* **69** (1–4), 119–143.
- Chang, F. J., Chang, L. C. & Huang, H. L. 2002 Real-time recurrent learning neural network for stream-flow forecasting. *Hydrological Processes* **16** (13), 2577–2588. doi:10.1002/hyp.1015.
- Chen, Y., Kang, Y., Chen, Y. & Wang, Z. 2020 Probabilistic forecasting with temporal convolutional neural network. *Neurocomputing* **399**, 491–501. doi:10.1016/j.neucom.2020.03.011.
- Crivellari, A. & Beinat, E. 2020 Trace2trace – a feasibility study on neural machine translation applied to human motion trajectories. *Sensors* **20** (12). doi:10.3390/s20123503.
- Deng, S., Zhang, N., Zhang, W., Chen, J., Pan, J. Z. & Chen, H. 2019 Knowledge-driven stock trend prediction and explanation via temporal convolutional network. *Companion of the World Wide Web Conference* 678–685. doi:10.1145/3308560.3317701..
- Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. & Trevor, D. 2015 Long-term recurrent convolutional networks for visual recognition and description. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2625–2634.
- Douglas-Mankin, K. R., Srinivasan, R. & Arnold, J. G. 2010 Soil and water assessment tool (SWAT) model: current developments and applications. *Transactions of the ASABE* **53** (5), 1423–1431.
- Gao, S., Huang, Y., Zhang, S., Han, J., Wang, G., Zhang, M. & Lin, Q. 2020 Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation. *Journal of Hydrology* **589**. doi:10.1016/j.jhydrol.2020.125188.
- He, K. M., Zhang, X. Y., Ren, S. Q. & Sun, J. 2016 Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 770–778. doi:10.1109/cvpr.2016.90.
- Hochreiter, S. & Schmidhuber, J. 1997 Long short-term memory. *Neural Computation* **9** (8), 1735–1780. doi:10.1162/neco.1997.9.8.1735.
- Hsu, K. I., Gupta, H. V. & Sorooshian, S. 1995 Artificial neural network modeling of the rainfall-runoff process. *Water Resources Research* **31** (10), 2517–2530.
- Hu, C., Guo, S., Peng, D. & Xiong, L. 2003 Comparative study on hydrological models in semi-dry and semi-humid region. *Engineering Journal of Wuhan University Engineering Edition* **36** (5), 38–42.
- Hu, C., Wu, Q., Li, H., Jian, S., Li, N. & Lou, Z. 2018 Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. *Water* **10** (11). doi:10.3390/w10111543.
- Kisi, O. 2011 Wavelet regression model as an alternative to neural networks for river stage forecasting. *Water Resources Management* **25** (2), 579–600. doi:10.1007/s11269-010-9715-8.
- Kratzert, F., Klotz, D., Brenner, C., Schulz, K. & Herrnegger, M. 2018 Rainfall-runoff modelling using long short-term memory (LSTM) networks. *Hydrology and Earth System Sciences* **22** (11), 6005–6022. doi:10.5194/hess-22-6005-2018.
- Kumar, D. N., Raju, K. S. & Sathish, T. 2004 River flow forecasting using recurrent neural networks. *Water Resources Management* **18** (2), 143–161.
- Lara-Benitez, P., Carranza-Garcia, M., Luna-Romera, J. M. & Riquelme, J. C. 2020 Temporal convolutional networks applied to energy-related time series forecasting. *Applied Sciences-Basel* **10** (7). doi:10.3390/app10072322.
- Le, X.-H., Hung Viet, H., Lee, G. & Jung, S. 2019 Application of long short-term memory (LSTM) neural network for flood forecasting. *Water* **11** (7). doi:10.3390/w11071387.
- Lin, K., Sheng, S., Zhou, Y., Liu, F., Li, Z., Chen, H., Xu, C., Chen, J. & Guo, S. 2020 The exploration of a temporal convolutional network combined with encoder-decoder framework for runoff forecasting. *Hydrology Research* **51** (5), 1136–1149. doi:10.2166/nh.2020.100.
- Liu, J., Zhu, H., Liu, Y., Wu, H., Lan, Y. & Zhang, X. 2019 Anomaly detection for time series using temporal convolutional networks and Gaussian mixture model. 2018 International Symposium on Power Electronics and Control Engineering. *Journal of Physics Conference Series*. doi:10.1088/1742-6596/1187/4/042111.

- Long, J., Shelhamer, E. & Darrell, T. 2015 [Fully convolutional networks for semantic segmentation](#). In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 3431–3440. doi:10.1109/cvpr.2015.7298965.
- Miao, Q., Pan, B., Wang, H., Hsu, K. & Sorooshian, S. 2019 [Improving monsoon precipitation prediction using combined convolutional and long short term memory neural network](#). *Water* **11** (5). doi:10.3390/w11050977.
- Nash, J. E. & Sutcliffe, J. V. 1970 River flow forecasting through conceptual models Part I – a discussion of principles. *Journal of Hydrology* **10** (3), 290.
- Qin, J., Liang, J., Chen, T., Lei, X. & Kang, A. 2018 [Simulating and predicting of hydrological time series based on TensorFlow deep learning](#). *Polish Journal of Environmental Studies* **28** (2), 795–802. doi:10.15244/pjoes/81557.
- Sahoo, B. B., Jha, R., Singh, A. & Kumar, D. 2019 [Long short-term memory \(LSTM\) recurrent neural network for low-flow hydrological time series forecasting](#). *Acta Geophysica* **67** (5), 1471–1481. doi:10.1007/s11600-019-00330-1.
- Shamseldin, A. Y. 1997 [Application of a neural network technique to rainfall-runoff modelling](#). *Journal of Hydrology* **199** (3–4), 272–294.
- Shiri, J., Marti, P., Nazemi, A. H., Sadraddini, A. A., Kisi, O., Landaras, G. & Fard, A. F. 2015 [Local vs. external training of neuro-fuzzy and neural networks models for estimating reference evapotranspiration assessed through k-fold testing](#). *Hydrology Research* **46** (1), 72–88. doi:10.2166/nh.2013.112.
- Tian, Y., Xu, Y.-P., Yang, Z., Wang, G. & Zhu, Q. 2018 [Integration of a parsimonious hydrological model with recurrent neural networks for improved streamflow forecasting](#). *Water* **10** (11). doi:10.3390/w10111655.
- Tokar, A. S. & Johnson, P. A. 1999 [Rainfall-runoff modeling using artificial neural networks](#). *Journal of Hydrologic Engineering* **4** (3), 232–239. doi:10.1061/(ASCE)1084-0699(1999)4:3(232).
- van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. & Kavukcuoglu, K. 2016 [WaveNet: a generative model for raw audio](#). arXiv: 1609.03499.
- Wan, R., Mei, S., Wang, J., Liu, M. & Yang, F. 2019 [Multivariate temporal convolutional network: a deep neural networks approach for multivariate time series forecasting](#). *Electronics* **8** (8). doi:10.3390/electronics8080876.
- Wang, G. Q., Zhang, J. Y., Jin, J. L., Pagano, T. C., Calow, R., Bao, Z. X., Liu, C. S., Liu, Y. L. & Yan, X. L. 2012 [Assessing water resources in China using PRECIS projections and a VIC model](#). *Hydrology and Earth System Sciences* **16** (1), 231–240. doi:10.5194/hess-16-231-2012.
- Wen, Y., Hu, C., Zhang, G. & Jian, S. 2020 [Response of the parameters of excess infiltration and excess storage model to land use cover change](#). *Journal of Hydrology and Hydromechanics* **68** (2), 99–110. doi:10.2478/johh-2020-0006.
- Wu, D., Jiang, Z., Xie, X., Wei, X., Yu, W. & Li, R. 2020 [LSTM learning with Bayesian and Gaussian processing for anomaly detection in industrial IoT](#). *IEEE Transactions on Industrial Informatics* **16** (8), 5244–5253. doi:10.1109/tii.2019.2952917.
- Yuan, X., Chen, C., Lei, X., Yuan, Y. & Adnan, R. M. 2018 [Monthly runoff forecasting based on LSTM-ALO model](#). *Stochastic Environmental Research and Risk Assessment* **32** (8), 2199–2212. doi:10.1007/s00477-018-1560-y.
- Yujie, L., Hongbin, D., Xingmei, W. & Shuang, H. 2019 [Time series prediction based on temporal convolutional network](#). In: *2019 IEEE/ACIS 18th International Conference on Computer and Information Science*, pp. 300–305.
- Yunpeng, L., Di, H., Junpeng, B. & Yong, Q. 2017 [Multi-step ahead time series forecasting for different data patterns based on LSTM recurrent neural network](#). In: *2017 14th Web Information Systems and Applications Conference (WISA)*. pp. 305–310. doi:10.1109/wisa.2017.25.
- Zhao, R. 1992 [The Xinanjiang model applied in China](#). *Journal of Hydrology* **135**, 371–381. doi:10.1016/0022-1694(92)90096-E.

First received 10 March 2021; accepted in revised form 23 June 2021. Available online 7 July 2021