AIMS Mathematics

AIMS Mathematics, 9(6): 14681–14696.
DOI: 10.3934/math.2024714
Received: 21 January 2024
Revised: 04 March 2024
Accepted: 08 March 2024
Published: 23 April 2024

http://www.aimspress.com/journal/Math

Research article

Flood prediction with optimized gated recurrent unit-temporal convolutional network and improved KDE error estimation

Chenmin Ni[1,2], Muhammad Fadhil Marsani[2,*], Fam Pei Shan[2] and Xiaopeng Zou[3]

[1] School of International Business, Zhejiang Yuexiu University, Shaoxing 312000, China
[2] School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia
[3] School of Economics, Zhejiang University, Hangzhou 310058, China

* **Correspondence:** Email: fadhilmarsani@usm.my; Tel: +6046533657.

Abstract: Flood time series forecasting stands a critical challenge in precise predictive models and reliable error estimation methods. A novel approach utilizing a hybrid deep learning model for both point and interval flood prediction is presented, enhanced by improved kernel density estimation (KDE) for prediction comparison and error simulation. Firstly, an optimized gated recurrent unit-time convolutional network (GRU-TCN) is constructed by tuning the internal structure of the TCN, the activation function, the L2 regularization, and the optimizer. Then, Pearson Correlation is used for feature selection, and the hyperparameters of the improved GRU-TCN are optimized by the subtraction-average-based optimizer (SABO). To further assess the prediction uncertainty, interval predictions are provided via Non-parametric KDE, with an optimized bandwidth setting for accurate error distribution simulation. Experimental comparisons are made on 5-year hydro-meteorological daily data from two stations along the Yangtze River. The proposed model surpasses long short-term memory network (LSTM), TCN, GRU, TCN-LSTM, and GRU-TCN, with a reduction of more than 13% in root mean square error (RMSE) and approximately 15% in mean absolute error (MAE), resulting in better interval estimation and error control. The improved kernel density estimation curves for the errors are closer to the mean value of the confidence intervals, better reflecting the trend of the error distribution. This research enhances the accuracy and reliability of flood predictions and improves the capacity of humans to cope with climate and environmental changes.

Keywords: flood prediction; temporal convolutional networks; gated recurrent units; subtraction-average-based optimizer; kernel density estimation
Mathematics Subject Classification: 62G05, 68Q32

1. Introduction

Floods are a major global natural disaster, causing significant property damage and safety hazards to humans. According to a report by the World Meteorological Organization, floods have caused 11,072 disaster reports, 2,064,929 deaths, and a total economic loss of US$ 3.6 trillion between 1970 and 2019 [1]. Although traditional algorithms based on Arima [2] and Bayesian models have achieved good prediction results [3], there are challenges in dealing with nonlinear relationships that reduce model performance. With the development of artificial intelligence, integrating mathematical statistics with artificial intelligence in deep learning has emerged as a powerful approach for nonlinear prediction, gaining prominence in flood forecasting research [4–6]. In deep learning models, a long short-term memory network (LSTM) is a special type of recurrent neural network (RNN) with gating mechanism and long short-term memory function, which solves the problem of gradient vanishing that traditional RNNs are prone to when dealing with long sequential data, and embodies a good performance of water level prediction[7,8]. However, LSTM lacks bi-directional learning capabilities and has many parameters that are prone to cause problems with hard training and overfitting [9]. gated recursive unit (GRU) [10] is a variant of LSTM, which has the advantage of a more straightforward structure and fewer parameters. In [11], the authors proposed a multi-directional GRU with a convolutional neural network (CNN) to improve the accuracy of load and energy forecasting. In the case of small samples, GRU is more accessible to train and tune to obtain good flood prediction ability [12]. In 2018, Bai et al. [13] showed that time convolutional networks (TCNs), which use a particular convolution, do better than other common networks like LSTM and GRU in many different tasks. Since then, TCNs have been used extensively in various research areas, such as gas drainage prediction [14], remaining useful life prediction [15]. In [16,17], the authors used TCN for flood and water level prediction, proved that the TCN-based model outperformed the LSTM-based model. Many studies used hybrid models or optimized parameters to improve the accuracy of water level prediction, Zhang et al. [18] constructed TCM-LSTM to predict groundwater levels, confirming that hybrid models have better predictive ability. In [18], three deep learning models, LSTM, GRU, and TCN, were used to predict water levels at five stations, and the Bayesian model average was used to synthesize the prediction results of the three deep learning models, then improved prediction accuracy. However, in the hybrid model, the parameter sensitivity is more extensive, and the rational selection of parameters is more complicated.

Most flood forecasting research focuses on point estimate forecasting, which makes it difficult to estimate the uncertainty of flood forecasting. Interval forecasting is a forecasting method that can provide an estimate of forecasting uncertainty. It helps to better understand the scope and possible risks of prediction and is currently a hot prediction research topic [19–22]. The methods for predicting intervals can be classified into parametric and non-parametric methods. Parametric methods, such as Bayesian interval estimation, use prior knowledge and sample data to estimate the distribution of parameters and calculate interval estimates based on the posterior distribution [23,24]. When dealing with highly fluctuating and random data that makes it difficult to estimate prior distributions, non-parametric methods such as resampling technique bootstrap and kernel density estimation (KDE)-based methods are commonly used for interval estimation [25,26]. In [25], an interval prediction for wind power by combining graph neural network and bootstrap technology was provided, by which the prediction uncertainty is analyzed. The bootstrap-based methods approximate the overall sampling distribution for parameter estimation, confidence interval estimation, or hypothesis testing by resampling the sample data rather than relying on the specific form of the distribution. KDE is based on the maximum likelihood principle and reveals data structure through smoothing. KDE is particularly suitable for scenarios with small samples and high sampling errors. Regarding computational efficiency, KDE usually outperforms bootstrap. In [26,27], the authors utilized KDE-Gaussian for interval prediction to further quantify the uncertainty of the prediction results. In [28], the authors performed interval estimation of the LSTM-GRU network using KDE and compared the prediction effects under four kernel functions to estimate the probability density of forecast errors. These methods help understand the uncertainty in predictions and are essential for decision-making and risk assessment. KDE is a non-parametric method to estimate the probability density function of a random variable. It smoothens the data points and is particularly useful when the underlying distribution is unknown. However, it is essential to note that choosing the suitable bandwidth and kernel function in KDE is crucial for accurate error estimation.

Based on the above analysis, an optimized GRU-TCN prediction framework is proposed, implements point and interval estimation to evaluate the prediction uncertainty, improves the bandwidth selection method of KDE to increase the fitness of the error probability density curve, and achieves quantitative evaluation of the uncertainty in prediction results. The main contributions of this research are as follows:

(1) Pearson correlation is used for feature selection, and the hyperparameters of the improved GRU-TCN are optimized by the subtraction-average-based optimizer (SABO).

(2) The internal structure of the TCN network is adjusted, the Selu activation function is used to alleviate the problem of gradient vanishing, the L2 regularization parameter is introduced to prevent over fitting, and the AdamW optimizer is adopted instead of the Adam optimizer to reduce over-regularization.

(3) To further evaluate the uncertainty of flood prediction, the interval prediction is implemented with Non-parametric KDE. The bandwidth setting of KDE is improved, and the prediction error distribution is analyzed for model comparison.

Using the hydro-meteorological data from two stations along the Yangtze River, the proposed model surpasses LSTM, TCN, GRU, TCN-LSTM, TCN-GRU, and GRU-TCN, with a reduction of more than 13% in root mean square error (RMSE) and approximately 15% in mean absolute error (MAE). The combination of interval and point estimation provides a more comprehensive analysis for flood prediction, and the estimation of the prediction error distribution based on the improved KDE benefits the model's accuracy and reliability. This approach introduces a novel theoretical framework for flood prediction and model comparison. Further, it enhances the applicability and effectiveness of hybrid models in complex environments, thus providing better support for the decision-making process in flood management.

2. Methods

2.1. GRU network

The GRU network simplifies the LSTM architecture by combining the input and forgets gates into a single update gate and adding a reset gate. This design addresses the issue of information redundancy that occurs during LSTM's training process ([10,18]). Figure 1a shows the basic structure of GRU. The calculation equations of the reset gate r_t, the update gate z_t, and the basic unit output h_t for prediction by the GRU network are as follows [18]:

$$\begin{aligned} r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] + b_r), \\ z_t &= \sigma(W_z \cdot [h_{t-1}, x_t] + b_z), \\ \tilde{h}_t &= \tanh(W_{\tilde{h}} \cdot [r_t * h_{t-1}, x_t] + b_{\tilde{h}}), \\ h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t, \end{aligned} \tag{1}$$

where W and b represents the weight matrix and deviation vector of each unit structure, respectively; $*$ stands for the dot product of a matrix; σ represents a sigmoid function; and tanh $(\cdot)$ represents a hyperbolic tangent function.x represents the feature vector in the input network at time t, $\tilde{h}_\mathrm{t}$ represents the update information of the current unit at time t, and $h_{\mathrm{t}-1}$ represents the network calculation output at previous time $\mathrm{t}-1$.

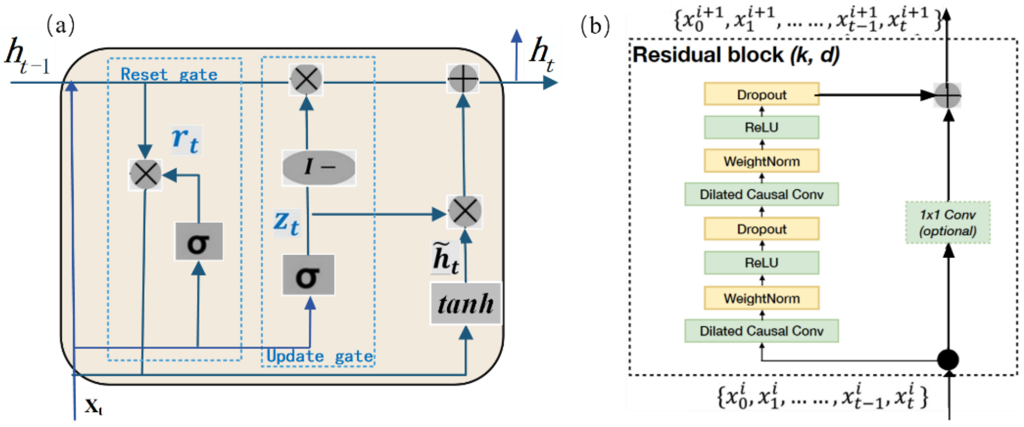

Figure 1. The basic structure of GRU and TCN network: (a) GRU unit; (b) TCN unit.

2.2. Temporal convolutional network

TCN is a CNN that utilizes convolution layers to process time series data. It employs a distinctive causal dilated convolution, consisting of one-dimensional dilated and causal convolution layers with identical input and output lengths. This guarantees the causal relationship of the input sequence and prevents future data leakage. Simultaneously, TCN can efficiently capture the local dependencies in sequence data and retain more historical information by utilizing stacked convolution layers and increasing the receptive field of the convolution kernel. Equation (2) shows the extended causal convolution calculation (see [13,18,29]):

$$G(F, X) = (F * X)_{x_t} = \sum_{k=1}^{K} f_k \cdot x_{t-(K-k)d}, \tag{2}$$

where $X = \{x_1, x_2, \dots, x_T\}$ represents the input time series, $F = \{f_1, f_2, \dots, f_K\}$ represents the size of the convolution kernel, f_K represents the number of filters in the convolution operation, and d represents the expansion rate.

In addressing the gradient vanishing problem inherent in convolutional degradation, TCN has been innovatively enhanced with residual modules. These modules facilitate a seamless data flow across layers, maintaining integrity without bypassing intermediate stages. This methodology significantly reduces information loss during the feature extraction phase in TCN, culminating in the aggregation of causally convoluted feature sets for output generation. Comprising two dilated causal convolutions, batch normalization, dropout techniques, and Relu activation functionalities, the residual module exemplifies a sophisticated architectural integration. To preserve dimensional consistency between input and output, a 1×1 convolution is adeptly employed, as illustrated in Figure 1b. For more details about TCN, please refer to [13].

2.3. *Subtraction-average-based optimizer (SABO)*

SABO, introduced in 2023, is grounded in the principles of mathematical variable characteristics, encompassing concepts like average values, disparities in search locations, and the formulation of objective functions [30]. In SABO, the displacement of any search agent X_i in the search space is calculated by the arithmetic mean of the v-subtraction of each search agent $X_j, j = 1,2, \dots, N$, from the search agent X_i. The updated position for every search agent is calculated by Eq (3) [30]:

$$X_i^{\text{new}} = X_i + \vec{r}_i * \frac{1}{N}\sum_{j=1}^{N} \left(X_i - vX_j\right), i = 1,2, \dots, N, \tag{3}$$

where X_i^{new} is the new proposed position for the i-Th search agent X_i, N is the total number of the search agents, and $\vec{r}_i$ is a vector of the dimension m, in which components have a normal distribution with the values from the interval [0,1].

SABO uses a dynamic approach to calculate the mean and standard deviation of the data. It utilizes the arithmetic mean position of all search agents for updating each agent's position, instead of exclusively depending on the positions of the best or worst agents. Concurrently, the model's parameters are adjusted in response to this statistical data. This characteristic enables the SABO model to manage noise and outliers adaptively and swiftly adjust to new conditions.

2.4. Research framework and optimized GRU-TCN network

Figure 2 shows the research flowchart, and the improvement details of the optimized GRU-TCN network are given in Sections 2.4.1 and 2.4.2.

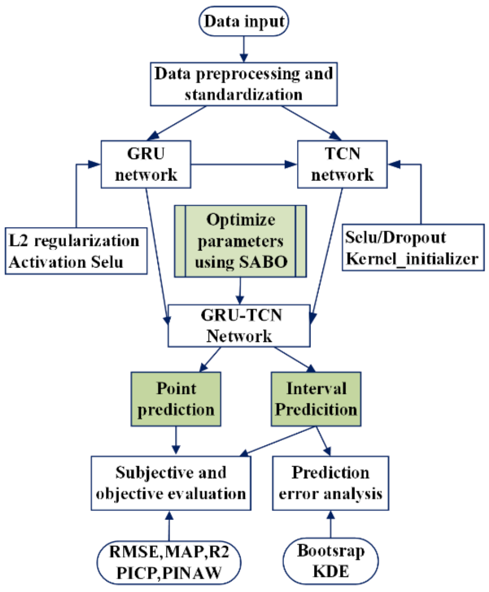
Data input
Data preprocessing and standardization
GRU network
TCN network
L2 regularization Activation Selu
Optimize parameters using SABO
Selu/Dropout Kernel_initializer
GRU-TCN Network
Point prediction
Interval Predicition
Subjective and objective evaluation
Prediction error analysis
RMSE,MAP,R2 PICP,PINAW
Bootsrap KDE

Figure 2. The research framework.

Based on Figure 2, the research workflow follows a structured process. First, data preprocessing and standardization are conducted as per Section 3.1. Then, the optimized GRU-TCN network is constructed in Section 2.4.1. After that, the network undergoes hyper-parameter optimization, as detailed in Section 2.4.2. Finally, Section 3.3 covers point and interval estimations and the evaluation of specific experiments. The optimized bandwidth for KDE is determined according to Section 2.4.3. Finally, in Section 3.4, kernel density estimation is used to compare the prediction error rates, enabling a more comprehensive evaluation of the model's predictive performance.

2.4.1. Establishment and optimization of the GRU-TCN network

According to the experimental tests, an optimized hybrid neural network is constructed. The data is first inputted into a single layer of a GRU neural network to learn signal features. To prevent overfitting, L2 regularization is added to the GRU. The output features of the GRU are then transmitted to a TCN network, where the internal structure is adjusted. The activation function "Relu" is replaced with "Selu". The self-normalization property of "Selu" helps to stabilize the neural network training, and its output range is close to zero mean and unit variance, this reduces the risk of gradient disappearance or explosion and achieves a better fitting effect.

The normal weight matrix initialization is replaced with orthogonal initialization, which initializes the weight matrix as an orthogonal matrix. The orthogonal matrix's transpose matrix equals its inverse matrix, which can help maintain the numerical stability of the weight matrix during the training process.

In the selection of optimizer for hybrid network, AdamW optimizer combining Adam optimization algorithm and weight attenuation is adopted. Unlike traditional Adam optimizers, where L2 regularization is typically achieved by adding a weight-squared term to the loss function, this AdamW optimizer addresses weight decay independently from the optimization process. This approach is crucial in adaptive learning rate algorithms, as it prevents the compromise of L2 regularization effectiveness due to learning rate adjustments. By treating weight decay separately and not as a part of the loss function, the AdamW optimizer ensures that the process is unaffected by learning rate modifications, thereby enhancing its effectiveness.

For other parameters of TCN, the architecture is designed with 2 stacks of residual blocks, incorporating a sequence of dilations set as [1,2,4,8,16,32,64], the dropout rate is set to 0.15, and "orthogonal" is used as the core initializer to stabilize training and improve efficiency.

2.4.2. Optimizing hyperparameters of GRU-TCN using SABO

The hyperparameters of the GRU-TCN model, including the learning rate, number of filters for TCN, number of hidden layers, and batch sizes for GRU, are optimized using SABO. The optimization process involved a population size of 10 and 60 iterations, with a search dimension of 4. The four parameters' lower and upper search limits are set to [0.0005,5,10,15] and [0.001,12,60,40]. Figure 3 displays the process of optimizing the data parameters for the Hankou station.

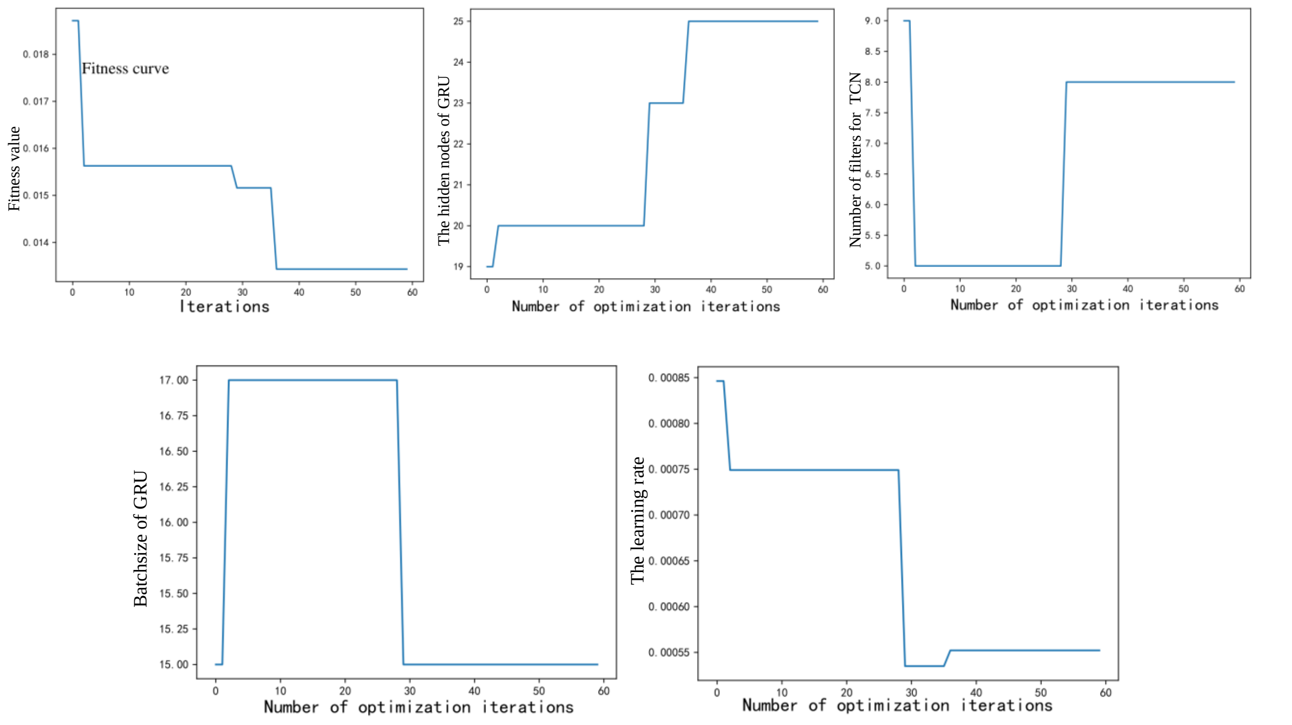

Figure 3. Optimization of GRU-TCN parameters with SABO.

Figure 3 shows that after 60 iterations, at a fitness value of 0.0134, the learning rate, number of filters in TCN, number of hidden layers in GRU, and optimal batch size parameters are [0.00055,8,25,15]. Similarly, the optimization parameters for the Luoshan station have been optimized.

2.4.3. KDE-based interval estimation and bandwidth optimization

Let $X_1, X_2, \dots, X_n$ be an independent and identically distributed sample from a univariate variable X, the kernel density estimate of the density function of the distribution that X follows is given by the Eq (4) [31]:

$$f(\mathrm{X}) = \frac{1}{n}\sum_{j=1}^{n} \frac{1}{h} K\left(\frac{X-X_i}{h}\right), \tag{4}$$

where n is the number of the sample, $K(.)$ is the kernel function, $h > 0$ is a smoothing parameter, called bandwidth.

There are several kernel functions to choose from for specific use, such as Gauss, Laplace, and Cauchy. Researchers can also design any suitable kernel functions, but they must satisfy the nature of probability density. The Gaussian kernel function is selected in this study.

In KDE, bandwidth determines the smoothness and shape of the estimated probability density function. Adjusting the bandwidth allows for a better fit to the data and a more accurate distribution characterization. It is important to note that the bandwidth parameter should be carefully chosen to avoid under or over-smoothing the data. In traditional KDE, a fixed bandwidth is typically used, but this may result in errors due to varying estimation accuracies in different density regions. To optimize bandwidth selection, the golden section search algorithm [32] is adopted to adaptively search for the optimal bandwidth. The optimal bandwidth is dynamically adjusted by seeking the minimum value of the loss function, as shown in Eq (5), to obtain the best result

$$\text{loss} = y^2 - 2y \times y_{\text{hist}} + \frac{2}{\sqrt{2\pi}} \times \frac{1}{h} \times y_{\text{hist}}, \tag{5}$$

where loss is the loss function,y is the kernel density estimate. y_{hist} is the data histogram. h is the locally optimal bandwidth.

To conduct interval predictions using KDE, the first step involves selecting a suitable kernel function and bandwidth for data smoothing. Then, calculate the contribution by applying the kernel function and selected bandwidth to each data point. Next, compute the cumulative probability density function to determine the confidence interval. The bounds of the intervals at a given confidence level can be found by performing density estimation on the dataset. This method is well-suited for data with irregular or non-standard distributions because it does not require strict assumptions about the distribution.

3. Data and results

3.1. Study area and data preparation

The study area is in Wuhan, China, as depicted in Figure 4. Wuhan, with a population of over 13 million, serves as China's economic and geographic center. It is prone to flooding in the Yangtze River basin due to its geographic positioning. Located at the convergence of the Yangtze and Hanging Rivers, the city regularly confronts substantial flood challenges in the midsection of the Yangtze River. Its coordinates range roughly between 29°58′N to 31°22′N latitude and 113°41′E to 115°05′E longitude.

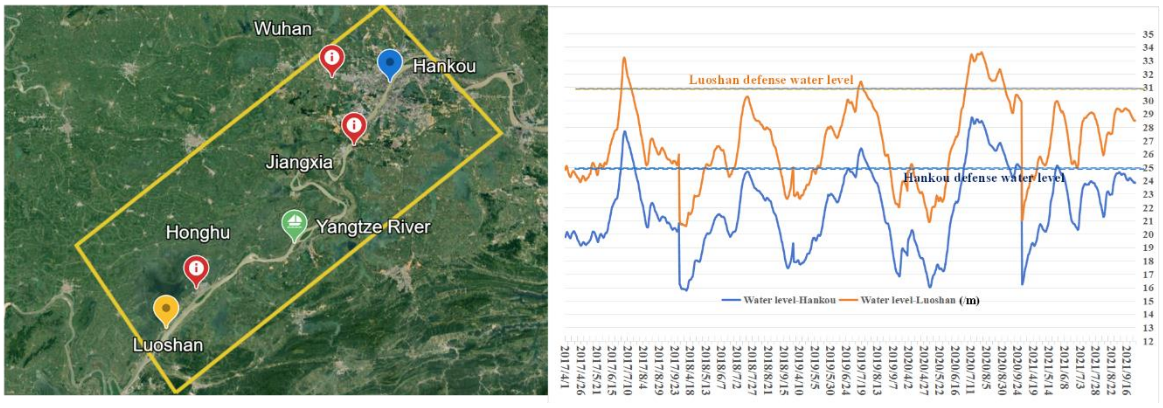

Figure 4. The research area and data visualization: (a) target area; (b) true water level.

Yangtze River flooding is primarily caused by heavy basin rainfall, upper reaches snowmelt, constrained river channels, and sediment accumulation. These elements collectively result in elevated water levels and subsequent flooding. The defense water levels of Hankou and Luoshan stations are 25 m and 31 m, respectively. According to Figure 3b, the risk of a relatively high water level has been concentrated in July for the past five years, and the water level exceeding the fortification line has been concentrated in June-September. This study is based on daily data from April to September yearly to improve training efficiency and relevance.

The water level and hourly data from adjacent meteorological stations for 2017–2021 were collected from the thousand miles water and rainfall information query website and the China Meteorological Network. Outliers were eliminated using the 3 sigma principle, and missing values were filled by averaging with the sequence proximity points. The dataset for this study was constructed using the corresponding hydrological station data (http://113.57.190.228:8001/web/Report/RiverReport) and meteorological data (https://data.cma.cn/) at 6:00 PM each day. Rainfall was recorded as the daily cumulative value. Predictive feature inputs were selected using Pearson's correlation analysis, with variables meeting the correlation conditions at a test level of p=0.05. The selected input feature variables at the two stations are water volume, rainfall, temperature, wind speed, and humidity. Normalization uses Eq (6) after dividing the data by 0.8:0.2 for training and testing:

$$x_i^* = \frac{x_i - \bar{x}}{\sigma}, \tag{6}$$

where x_i^* is the input data after standardization, $\bar{x}$ is the mean value, and σ is the standard deviation.

3.2. Evaluation indicators

The four indexes shown in Table 1 (I) and Eq (7) are selected as objective evaluation indexes for point prediction evaluation, and the three indexes shown in Table 1 (II) and Eq (8) are chosen as the evaluation indexes for interval evaluation.

Table 1. Evaluation indicator description.

Num	Indexes	Interpretations	Variable description
(I)	*RMSE*	Root mean square error	x_i and $\hat{x}_i$ denote the i-th input actual water level value and the predicted value, n is the number of samples, $\bar{x} = (x_1 + x_2 + \cdots + x_n)/n$. Small *RMSE*, *MAPE*, *MAE* and R^2 close to 1 indicate the superiority of the prediction model.
	MAPE	Mean absolute percentage error	
	MAE	Mean absolute error	
	R²	Adjusted R-squared (goodness of fit)	
(II)	*PI*	Prediction interval	U_i represents the upper bound, and L_i denotes the lower bounds,V is the range of target values (for normalization). μ *is* the confidence level. Larger *PICP* and smaller *PINAW* and *CWC* represent better interval estimation.
	PINAW	PI normalized averaged width	
	PICP	PI coverage probability	
	CMC	Coverage width criterion	

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}, MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i}\right|, MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i|, R^2 = 1 - \frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \tag{7}$$

$$PICP = \left(\frac{1}{n}\sum_{i=1}^{n}\bar{\bar{C_\iota}}\right) \times 100\%,\ PINAW = \frac{\sum_{i=1}^{n}(U_i - L_i)}{n} \cdot V, CWC = PINAW\left(1 + \gamma e^{-\eta(PICP - \mu)}\right). \tag{8}$$

If $x_i \in (L_i, U_i), \bar{\bar{C}} = 1$, otherwise, $\bar{\bar{C_\iota}} = 0$. If $PICP < \mu, \gamma = 1$, otherwise, $\gamma = 0$.

The penalty parameter η determines the degree of penalty for failing to meet the *PICP* confidence level. As η decreases, the penalty decreases as well. Let $\eta = 10$ in the following experiments.

3.3. Experimental comparisons

3.3.1. Prediction step selection and parameter settings

To determine the appropriate prediction step size, the partial auto correlation function (PACF) is analyzed for both hydrological and meteorological data. Figure 5 illustrates the PACF plots for water level and rainfall, showing partial autocorrelation coefficients with 95% confidence intervals for lag orders from 1 to 100. Based on the experiments and data from Figure 5, the number of lag points is identified as 4 for meteorological data and 2 for hydrological data, both outside the 95% confidence interval. Therefore, the prediction step chosen for subsequent experiments is 3, averaging these two values.

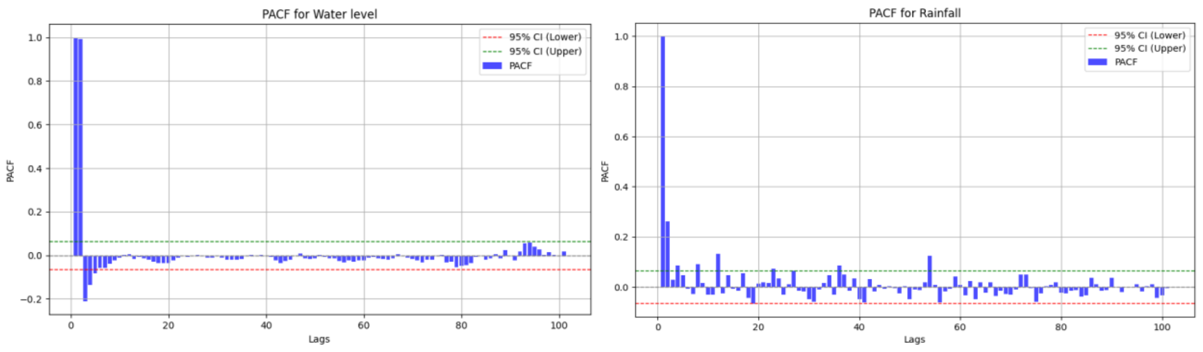

Figure 5. PACF hysteresis diagram.

The comparison algorithms for the following experiments are set up: LSTM and GRU are two-layer networks with parameters set as follows: epoch=100, batch size=32, and learning rate=0.001. The number of the first and second hidden layers are 16 and 32, respectively. TCN parameters are set as follows: The number of filters used in the convolutional layers is 10, the kernel size used in each convolutional layer is 6, the number of stacks of residual blocks is 2, and the dropout rate is 0. 15. Both TCN-LSTM and GRU-TCN use single-layer networks, with LSTM and GRU parameters set to match those of the first layer of the GRU bilayer network.

3.3.2. Point prediction and interval estimation

Tables 2 and 3 compare metrics between point and interval estimation, where the interval estimator is based on the 95% confidence interval of the KDE. In the following comparisons, LSTM is used in [33], GRU is used in [12], TCN is used in [16], TCN-LSTM is used in [17], TCN-GRU is used in [34], and SOGT(SABO-Optimized-GRU-TCN) is the proposed model.

As shown in Tables 2 and 3, the SOGT algorithm achieved the best point prediction results with the smallest RMSE, MAPE, and MAE and the largest R^2 compared to the six comparison algorithms. On the Hankou dataset, the SOGT algorithm reduced the RMSE by 13.01%, the MAPE and MAE by about 18.44%, 16.81%, respectively, compared to the GRU-TCN, and the R^2 is improved by 0.36%. PINAW is the smallest among the six algorithms, and PICP and CWC of TCN-LSTM also achieve good results.

Table 2. Experimental predictive results of Hankou station from 2017 to 2021.

Model	RMSE (m)	MAPE (%)	MAE (m)	R^2	PINAW	PICP	CWC
LSTM	0.2892	1.0261	0.221	0.98	0.0958	0.9333	1.1827
GRU	0.2869	1.0443	0.2335	0.9804	0.1222	0.9333	1.2091
TCN	0.3053	1.0227	0.23	0.9778	0.1364	0.9111	1.351
TCN-LSTM	0.2604	0.9033	0.2028	0.9838	0.114	0.9444	1.1422
TCN-GRU	0.4944	1.8927	0.4154	0.9417	0.2134	0.9556	0.2134
GRU-TCN	0.2475	0.889	0.1945	0.9854	0.1109	0.9333	1.1978
SOGT	0.2153	0.7251	0.1618	0.9889	0.0954	0.9333	1.1823

Table 3. Compare metrics for Luoshan station from 2017 to 2021.

Model	RMSE (m)	MAPE (%)	MAE (m)	R^2	PINAW	PICP	CWC
LSTM	0.2786	0.8334	0.2204	0.9807	0.1052	0.9222	1.2542
GRU	0.2707	0.7874	0.2133	0.9818	0.1115	0.9278	1.229
TCN	0.3194	0.9134	0.2512	0.9747	0.1419	0.9056	1.3908
TCN-LSTM	0.2834	0.8188	0.2242	0.9801	0.1304	0.9278	1.2479
TCN-GRU	0.423	1.3304	0.3496	0.9556	0.1799	0.9611	0.1799
GRU-TCN	0.2835	0.8851	0.2387	0.9801	0.1268	0.9500	0.1268
SOGT	0.2283	0.686	0.1878	0.9871	0.091	0.9389	1.1481

In the experiment at Luoshan station, the SOGT algorithm reduces the RMSE by 15.66%, MAPE and MAE by about 12.88%, 11.95%, respectively, and the R^2 is improved by 0.54% relative to GRU. PINAW is the smallest among the six algorithms, and PICP and CWC of GRU-TCN also achieve good results. Despite the modest CWC values observed for TCN-GRU and GRU-TCN models, they exhibited higher PINAW values, indicating less precision in their interval predictions. Moreover, their peak prediction curves did not achieve a good fit, suggesting a need for further refinement in capturing peak behaviors. The effect of point estimation, especially in predicting the flooding point, is that the SOGT algorithm has better tracking fit and higher accuracy (as shown in Figures 6a and 7a), so the SOGT algorithm proposed has better prediction stability. Excluding the time for parameter optimization, the runtime of different models revealed that a single model operates more swiftly, when the complexity of a hybrid model increases, leading to a longer runtime. The proposed SOGT's runtime is slightly shorter than that of GRU-TCN, demonstrating commendable operational efficiency.

Figures 6 and 7 present comparison plots of point estimation for Hankou and Luoshan stations, as well as visualizations of point and interval estimation evaluation metrics, and confidence intervals at different confidence levels of SOGT.

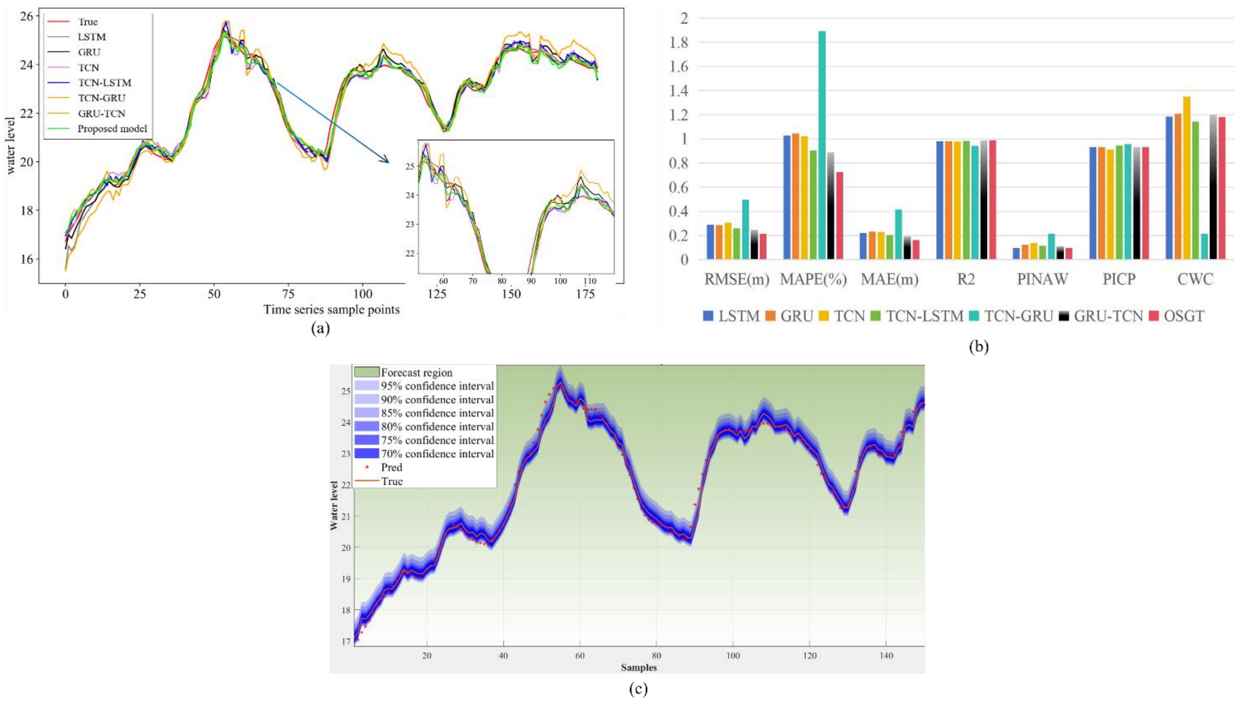

Figure 6. Prediction comparison of Hankou station: (a) comparison of point estimates for Hanko; (b) displaying indicators for different prediction models; (c) interval estimation of SOGT at different confidence levels.

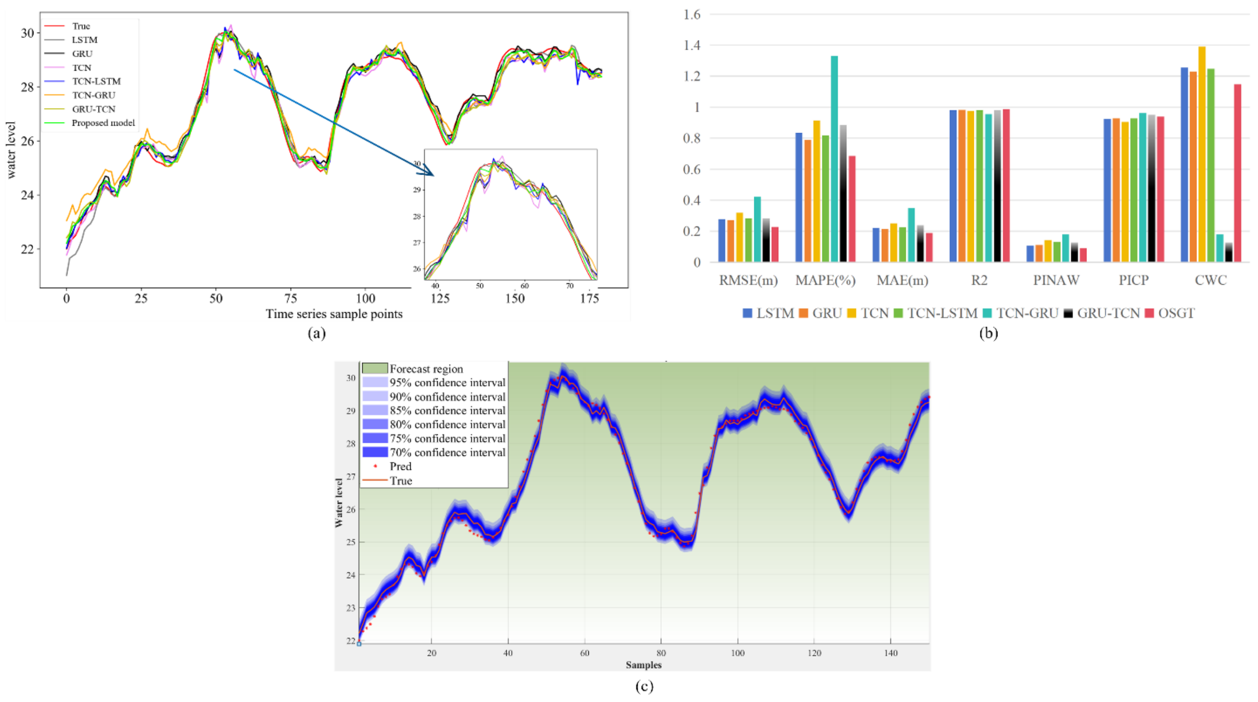

Figure 7. Prediction comparison of Luoshan station: (a) comparison of point estimates for Luoshan; (b) displaying indicators for different prediction models; (c) interval estimation of SOGT at different confidence levels.

Figures 6 and 7 show that the SOGT algorithm tracks the actual points more accurately, particularly, in the flooding point. The interval estimation curves demonstrate that the prediction intervals' coverage of the true values varies with different confidence levels. It is important to note that as the confidence level increases, coverage decreases, and real points are covered more extensively as the confidence level decreases. To ensure a balance between the confidence level and the interval width, a standard reference for interval estimation is 95%. Under the 95% confidence level, Figures 6c and 7c display the evaluation indexes for point estimation and interval estimation. Combined with Tables 2 and 3, it is evident that the algorithm proposed performs better in both point and interval predictions

4. Discussion

4.1. Error kernel density estimation based on improved bandwidth

An analysis of the distribution of prediction errors offers a more detailed comparison of model performance and insights into error characteristics, which can guide model refinement. Focusing on Hankou station, Figure 8 shows the KDE error estimation curves of seven models before and after the KDE improvement at a 95% confidence level, along with the mean curve of the confidence interval.

In Figure 8, the fitted curves based on the improved KDE bandwidth by the Golden Section Search algorithm are closer to the mean value of the confidence intervals and better reflect the trend of the error confidence intervals. KDE error analysis can reveal biases or systematic errors in a model. Ideally, a model's errors would cluster symmetrically around zero. Figure 8 indicates that the LSTM's prediction errors are left-skewed with a mean less than the median, suggesting a tendency towards underestimation. On the contrary, the error distributions of GRU, TCN, TCN-LSTM, and TCN-GRU are right-skewed, with the mean exceeding the median, indicating overprediction. The error distribution of GRU-TCN is approximately symmetric, but there is a slight change in the right tail. Although the proposed improved SOGT model and GRU-TCN have similar shapes in their error distributions, the prediction error distribution of the SOGT model has more balanced tails and symmetry. Moreover, the peak of the kernel density curve at zero is particularly sharp, reaching about 2.4, which means that the prediction errors of the SOGT model are more concentrated around zero, implying a higher prediction accuracy. The enhanced SOGT model introduced in this study exhibits more balanced tails and improved symmetry, reflecting a more uniform error distribution.

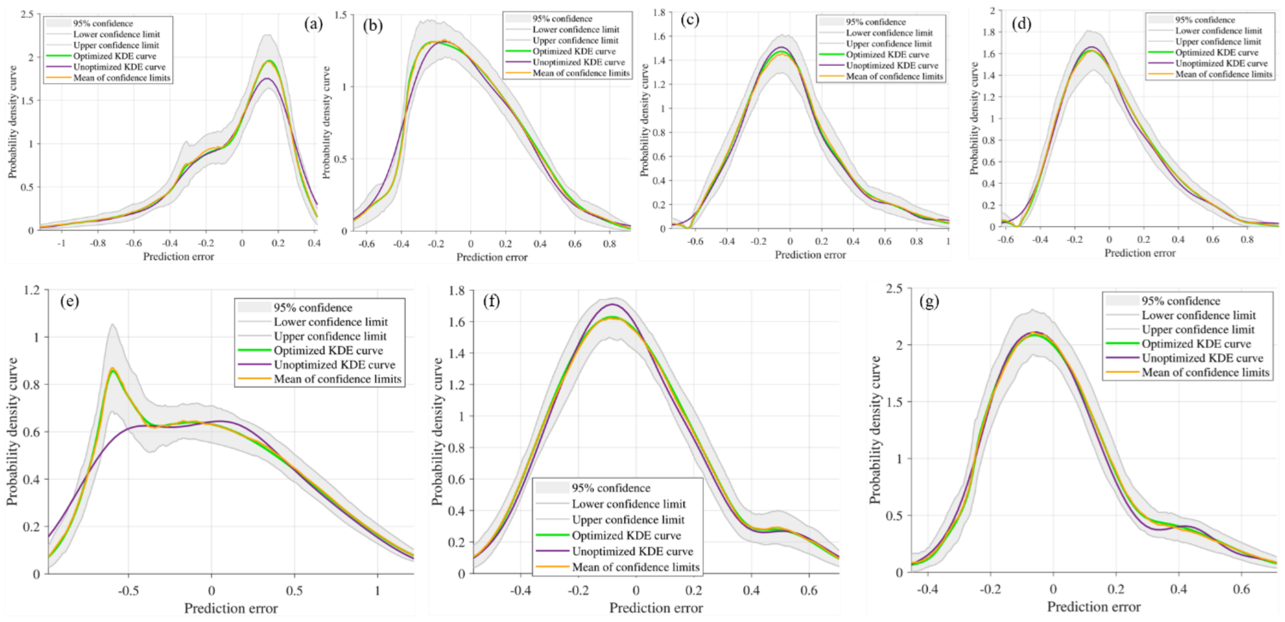

Figure 8. Comparison of KDE curves for prediction error of Hankou station: (a) LSTM; (b) GRU; (c) TCN; (d) TCN-LSTM; (e) TCN-GRU; (f) GRU-TCN; (g) SOGT.

4.2. Forecast error and evaluation analysis

Deep learning prediction algorithms *can have a* black box effect, making it difficult to understand the interrelationships within the network. Therefore, different datasets often require different network structures and parameters. Predictive errors can be reduced by adjusting the network structure, combining networks, and adjusting parameters. Additionally, appropriate variable selection can help improve prediction accuracy and reduce error sensitivity due to the large number of feature variables. In model evaluation, many algorithms predominantly focus on point estimation metrics. However, this study reveals a divergence between point estimation and interval prediction indexes. A fairer assessment of prediction models emerges when both prediction modes are considered together. Furthermore, analyzing the distribution of prediction errors enriches understanding of a model's susceptibility to errors, offering more profound insights into its predictive reliability.

5. Conclusions

This study developed a daily time series dataset from hydrological and meteorological stations and advanced point and interval water level predictions for the Yangtze River stations, showcasing enhancements in model structure, parameter optimization, and evaluation techniques. The innovation centers on an optimized hybrid model for precise point and interval predictions, with improved KDE bandwidth for error analysis. These innovations in model evaluation are generalizable and can be evaluated for a wide range of forecasting models.

Despite its achievements, the study recognizes certain limitations, such as optimizing parameters may require a lot of computation and time, especially in large datasets, and determining the depth of hidden layers and the interpretability of the model structure is also more difficult. Future researchers can address these constraints by incorporating an attention mechanism and multimodal decomposition or finding other parameter selection methods, thereby refining the integrated model's predictive capabilities. Exploring how to improve performance by estimating the probability distribution of prediction errors is also a worthwhile research direction.

This research provides decision-makers with powerful tools for evaluating models, facilitating the application of predictive models in critical areas such as flood forecasting. Additionally, it offers an innovative research tool for constructing and evaluating deep learning models, which can be used not only for predicting flood time series but also for predicting and estimating errors in other multivariate time series.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

Acknowledgment to "Ministry of Higher Education Malaysia for Fundamental Research Grant Scheme with Project Code: FRGS/1/2022/STG06/USM/03/1".

Conflict of interest

The authors declare no conflicts of interest.

References

1. J. Douris, G. Kim, *The Atlas of mortality and economic losses from weather, climate and water extremes (1970–2019)*, World Meteorological Organization, 2021.
2. J. B. Liu, X. Y. Yuan, Prediction of the air quality index of Hefei based on an improved ARIMA model, *AIMS Math.*, **8** (2023), 18717–18733. https://doi.org/10.3934/math.2023953
3. B. Yan, R. Mu, J. Guo, Y. Liu, J. Tang, H. Wang, Flood risk analysis of reservoirs based on full-series ARIMA model under climate change, *J. Hydrol.*, **610** (2022), 127979. https://doi.org/10.1016/j.jhydrol.2022.127979
4. B. Jiang, S. Chen, B. Wang, B. Luo, MGLNN: semi-supervised learning via multiple graph cooperative learning neural networks, *Neural Networks*, **153** (2022), 204–214. https://doi.org/10.1016/j.neunet.2022.05.024
5. D. K. Hakim, R. Gernowo, A. W. Nirwansyah, Flood prediction with time series data mining: systematic review, *Nat. Hazards Res.*, in press, 2023. https://doi.org/10.1016/j.nhres.2023.10.001

6. A. M. Roy, J. Bhaduri, DenseSPH-YOLOv5: an automated damage detection model based on DenseNet and Swin-Transformer prediction head-enabled YOLOv5 with attention mechanism, *Adv. Eng. Inf.*, **56** (2023), 102007. https://doi.org/10.1016/j.aei.2023.102007
7. L. Zhang, H. Qin, J. Mao, X. Cao, G. Fu, High temporal resolution urban flood prediction using attention-based LSTM models, *J. Hydrol.*, **620** (2023), 129499. https://doi.org/10.1016/j.jhydrol.2023.129499
8. Z. Vizi, B. Batki, L. Rátki, S. Szalánczi, I. Fehérváry, P. Kozák, et al., Water level prediction using long short-term memory neural network model for a lowland river: a case study on the Tisza River, Central Europe, *Environ. Sci. Eur.*, **35** (2023), 92. https://doi.org/10.1186/s12302-023-00796-3
9. C. Ni, P. S. Fam, M. F. Marsani, A data-driven method and hybrid deep learning model for flood risk prediction, *Int. J. Intell. Syst.*, **2024** (2024), 3562709. https://doi.org/10.1155/2024/3562709
10. J. Chung, Ç. Gülçehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, *ArXiv*, 2014. https://doi.org/10.48550/arXiv.1412.3555
11. F. Abid, M. Alam, F. S. Alamri, I. Siddique, Multi-directional gated recurrent unit and convolutional neural network for load and energy forecasting: a novel hybridization, *AIMS Math.*, **8** (2023), 19993–20017. https://doi.org/10.3934/math.20231019
12. C. Ji, T. Peng, C. Zhang, L. Hua, W. Sun, An integrated framework of GRU based on improved whale optimization algorithm for flood prediction, *Res. Square*, 2021. https://doi.org/10.21203/rs.3.rs-947198/v1
13. S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, *ArXiv*, 2018. https://doi.org/10.48550/arXiv.1803.01271
14. H. Xue, X. Gui, G. Wang, X. Yang, H. Gong, Prediction of gas drainage changes from nitrogen replacement: a study of a TCN deep learning model with integrated attention mechanism, *Fuel*, **357** (2024), 129797. https://doi.org/10.1016/j.fuel.2023.129797
15. R. Gong, J. Li, C. Wang, Remaining useful life prediction based on multisensor fusion and attention TCN-BiGRU model, *IEEE Sensors J.*, **22** (2022), 21101–21110. https://doi.org/10.1109/jsen.2022.3208753
16. Y. Xu, C. Hu, Q. Wu, Z. Li, S. Jian, Y. Chen, Application of temporal convolutional network for flood forecasting, *Hydrol. Res.*, **52** (2021), 1455–1468. https://doi.org/10.2166/nh.2021.021
17. X. Zhang, F. Dong, G. Chen, Z. Dai, Advance prediction of coastal groundwater levels with temporal convolutional and long short-term memory networks, *Hydrol. Earth Syst. Sci.*, **27** (2023), 83–96. https://doi.org/10.5194/hess-27-83-2023
18. G. Li, Z. Liu, J. Zhang, H. Han, Z. Shu, Bayesian model averaging by combining deep learning models to improve lake water level prediction, *Sci. Total Environ.*, **906** (2024), 167718. https://doi.org/10.1016/j.scitotenv.2023.167718
19. M. A. Khanesar, D. T. Branson, Prediction interval identification using interval type-2 fuzzy logic systems: lake water level prediction using remote sensing data, *IEEE Sensors J.*, **21** (2021), 13815–13827. https://doi.org/10.1109/jsen.2021.3067841
20. J. Wang, Z. Li, Wind speed interval prediction based on multidimensional time series of convolutional neural networks, *Eng. Appl. Artif. Intell.*, **121** (2023), 105987. https://doi.org/10.1016/j.engappai.2023.105987
21. C. Pan, J. Tan, D. Feng, Prediction intervals estimation of solar generation based on gated recurrent unit and kernel density estimation, *Neurocomputing*, **453** (2021), 552–562. https://doi.org/10.1016/j.neucom.2020.10.027

22. B. Badyalina, N. A. Mokhtar, N. A. M. Jan, M. F. Marsani, M. F. Ramli, M. Majid, et al., Hydroclimatic data prediction using a new ensemble group method of data handling coupled with artificial bee colony algorithm, *Sains Malays.*, **51** (2022), 2655–2668. https://doi.org/10.17576/jsm-2022-5108-24
23. M. Zhou, Y. Zhang, S. Wu, L. Kong, Z. Wang, Interval prediction of remaining life of a bearing based on CNN, *J. Mech. Electr. Eng.*, **40** (2023), 1225–1230. https://doi.org/10.3969/j.issn.1001-4551.2023.08.011
24. X. Zhangsun, C. D, Research on equipment reliability of nuclear power plant by interval estimation of exponential distribution life test, *Nuclear Saf.*, **22** (2023), 90–94. https://doi.org/10.16432/j.cnki.1672-5360.2023.05.010
25. W. Liao, S. Wang, B. Bak-Jensen, J. R. Pillai, Z. Yang, K. Liu, Ultra-short-term interval prediction of wind power based on graph neural network and improved bootstrap technique, *J. Mod. Power Syst. Clean Energy*, **11** (2023), 1100–1114. https://doi.org/10.35833/mpce.2022.000632
26. H. Xu, Y. Chang, Y. Zhao, F. Wang, A novel hybrid wind speed interval prediction model based on mode decomposition and gated recursive neural network, *Environ. Sci. Pollut. Res.*, **29** (2022), 87097–87113. https://doi.org/10.1007/s11356-022-21904-5
27. L. Wu, Q. Tai, Y. Bian, Y. Li, Point and interval forecasting of ultra-short-term carbon price in China, *Carbon Manag.*, **14** (2023), 2275576. https://doi.org/10.1080/17583004.2023.2275576
28. M. Wang, F. Ying, Point and interval prediction for significant wave height based on LSTM-GRU and KDE, *Ocean Eng.*, **289** (2023), 116247. https://doi.org/10.1016/j.oceaneng.2023.116247
29. D. Li, F. Jiang, M. Chen, T. Qian, Multi-step-ahead wind speed forecasting based on a hybrid decomposition method and temporal convolutional networks, *Energy*, **238** (2022), 121981. https://doi.org/10.1016/j.energy.2021.121981
30. P. Trojovský, M. Dehghani, Subtraction-average-based optimizer: a new swarm-inspired metaheuristic algorithm for solving optimization problems, *Biomimetics*, **8** (2023), 149. https://doi.org/10.3390/biomimetics8020149
31. N. Jiang, X. Yu, M. Alam, A hybrid carbon price prediction model based-combinational estimation strategies of quantile regression and long short-term memory, *J. Clean. Prod.*, **429** (2023), 139508. https://doi.org/10.1016/j.jclepro.2023.139508
32. V. Paquianadin, K. N. Sam, G. Koperundevi, M. M. R. Singaravel, Current sensor-based single MPPT controller using sequential golden section search algorithm for hybrid solar PV generator-TEG in isolated DC microgrid, *Solar Energy*, **266** (2023), 112147. https://doi.org/10.1016/j.solener.2023.112147
33. J. F. Ruma, M. S. G. Adnan, A. Dewan, R. M. Rahman, Particle swarm optimization based LSTM networks for water level forecasting: a case study on Bangladesh river network, *Result Eng.*, **17** (2023), 100951. https://doi.org/10.1016/j.rineng.2023.100951
34. L. Li, Y. Li, R. Mao, L. Li, W. Hua, J. Zhang, Remaining useful life prediction for lithium-ion batteries with a hybrid model based on TCN-GRU-DNN and dual attention mechanism, *IEEE Trans. Transp. Electrific.*, **9** (2023), 4726–4740. https://doi.org/10.1109/tte.2023.3247614