Heterogeneous Dynamic Graph Convolutional Networks for Enhanced Spatiotemporal Flood Forecasting by Remote Sensing

Jiange Jiang, *Student Member, IEEE*, Chen Chen, *Senior Member, IEEE*, Yang Zhou,
Stefano Berretti, *Senior Member, IEEE*, Lei Liu, *Member, IEEE*, Qingqi Pei, *Senior Member, IEEE*,
Jianming Zhou, and Shaohua Wan, *Senior Member, IEEE*

***Abstract*—Accurate and timely flood forecasting, facilitated by remote sensing technology, is crucial to mitigate the damage and loss of life caused by floods. However, despite years of research, accurate flood prediction still faces numerous challenges, including complex spatiotemporal features and varied flood patterns influenced by multiple variables. Moreover, long-term flood forecasting is always tricky due to the constantly changing conditions of the surrounding environment. In this study, we propose a heterogeneous dynamic temporal graph convolutional network (HD-TGCN) for flood forecasting. Specifically, we designed a dynamic temporal graph convolution module (D-TGCM) to generate a dynamic adjacency matrix by incorporating a multihead self-attention mechanism, enabling our model to capture the dynamic spatiotemporal features of flood data by utilizing temporal graph convolution operations on the dynamic matrix. Furthermore, to reflect the impact of multiple meteorological and hydrological features on the heterogeneity of flood data, we propose a novel approach that utilizes multiple parallel D-TGCMs for processing heterogeneous graph data and implements a fusion mechanism to capture varied flood patterns influenced by multiple variables. Experiments conducted on a real dataset in Wuyuan County, Jiangxi Province, demonstrate that the HD-TGCN outperforms the state-of-the-art flood prediction models in mean absolute error, Nash–Sutcliffe efficiency, and root-mean-square error, with improvements of 80.32%, 0.15%, and 73.99%, respectively, providing a more accurate flood forecasting method that will play a critical role in future flood disaster prevention and control.**

***Index Terms*—Deep learning, dynamic graph convolution, flood forecasting, multivariable prediction, spatiotemporal graph data.**

Manuscript received 29 August 2023; revised 22 November 2023 and 20 December 2023; accepted 22 December 2023. Date of publication 3 January 2024; date of current version 18 January 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1807500, in part by the National Natural Science Foundation of China under Grant 62072360, Grant 62001357, Grant 62172438, and Grant 61901367, in part by the Key Research and Development Plan of Shaanxi province under Grant 2021ZDLGY02-09, Grant 2023-GHZD-44, and Grant 2023-ZDLGY-54, in part by the Natural Science Foundation of Guangdong Province of China under Grant 2022A1515010988, in part by the Key Project on Artificial Intelligence of Xi'an Science and Technology Plan under Grant 23ZDCYJSGG0021-2022, Grant 23ZDCYYYCJ0008, and Grant 23ZDCYJSGG0002-2023, in part by Xi'an Science and Technology Plan under Grant 20RGZN0005, and in part by the Proof-of-Concept Fund from Hangzhou Research Institute of Xidian University under Grant GNYZ2023QC0201. *(Corresponding authors: Chen Chen; Jianming Zhou.)*

Jiange Jiang, Lei Liu, and Qingqi Pei are with the School of Telecommunications Engineering, Xidian University, Xi'an 710071, China (e-mail: jiangejiang@stu.xidian.edu.cn; leiliu@mail.xidian.edu.cn; qqpei@mail.xidian.edu.cn).

Chen Chen is with the School of Telecommunications Engineering, Xidian University, Xi'an 710071, China, also with the Xidian Guangzhou Institute of Technology, Guangzhou 510555, China, and also with the Xidian Hangzhou Institute of Technology, Hangzhou 311231, China (e-mail: cc2000@mail.xidian.edu.cn).

Yang Zhou is with the Ministry of Water Resources of China, Beijing 101400, China (e-mail: zhy@mwr.gov.cn).

Stefano Berretti is with the Department of Information Engineering, University of Florence, 50127 Florence, Italy (e-mail: stefano.berretti@unifi.it).

Jianming Zhou is with the China Unicom Shenzhen Branch, Shenzhen 510627, China (e-mail: zhoujm27@chinaunicom.cn).

Shaohua Wan is with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518110, China (e-mail: shaohua.wan@uestc.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3349162

Nomenclature

$G, \mathcal{G}_t$	Graph matrix and the graph at time t.
$\mathcal{V}, v_i$	Nodes of graph, $\lvert\mathcal{V}\rvert = N$ and the ith node.
$E, \mathcal{E}_t$	Edges of graph and the edges at time t.
U	Eigenvector matrix.
Λ	Eigenvalue matrix.
D	Degree matrix.
$L, L_{\text{sym}}, \tilde{L}$	Laplacian matrix, regularized Laplacian matrix, and rescaled Laplacian matrix.
I_N	Identity matrix of size N.
Q, K, V	Query matrix, key matrix, and value matrix of the attention mechanism.
C	Number of dynamic graphs.
N_c	Number of nodes in the cth dynamic graphs.
$\mathbf{W}_t^c \in \mathbb{R}^{N_c \times N_c}$	Weighted adjacency matrix.
M	Number of features.
$\mathcal{X}_t \in \mathbb{R}^{C \times N_c \times M}$	Heterogeneous spatiotemporal graph signal.
T_h	Historical sequence length.
T_p	Predictive sequence length.
F	Number of objective nodes.
$\mathcal{Y} \in \mathbb{R}^{F \times T_p}$	Predicted spatiotemporal graph signal.
h	Number of heads in MHSA.
f, f_c	Predictive function and graph convolution filter.
u_t, r_t, c_t, h_t	Update state, reset state, cell state, and hidden state at time t.
σ^2	Variance.
ϵ	Threshold.

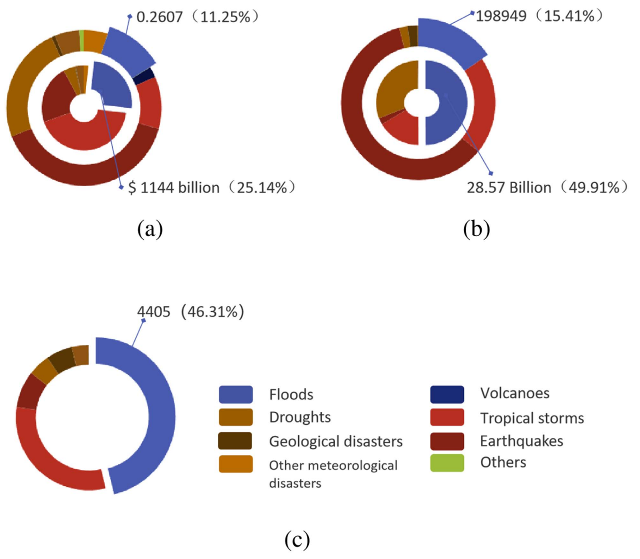
I. Introduction

FLOODS cause immense economic and social impact annually on a global scale, highlighting the critical need for timely and accurate flood forecasting to aid in flood control and disaster reduction efforts (see Fig. 1). Remote sensing technology, encompassing high-resolution satellite images and data from various sensors, can play a pivotal role in enhancing flood forecasting accuracy by providing comprehensive and precise surface information. This includes details such as water body distribution, terrain characteristics, and soil moisture, all of which are crucial for understanding the intricate spatiotemporal features involved in the formation and development of floods. Flood forecasting not only provides indispensable decision-making information for flood control departments but also assists in the regulation of reservoirs and managing river basins, mitigating losses caused by floods. Consequently, an expanding contingent of researchers is dedicating themselves to the study of flood forecasting. Early flood forecasting research can generally be categorized as process- and data-driven methods.

Process-driven methods establish flood forecasting models with physical significance by encompassing topographical, geological, climatic, and vegetational factors, along with natural physical traits, hydrodynamics, hydrologic processes, and flow generation [1]. Given that the model involves a myriad of disturbance factors, advanced expertise and experience are necessary to calibrate parameters or hypothesize unknown parameters, resulting in a considerable decrease in forecast efficiency and accuracy. Fortunately, with the advancement of the Internet of Things and 5G/6G technologies, the extensive accumulation and retention of hydrological data have catalyzed the rise of data-driven approaches.

Different from process-driven methods, data-driven methods do not require a priori consideration of the physical mechanisms of hydrological processes. Instead, they employ mathematical analyses of temporal data series and rely on differential or difference equations to excavate the pertinent relationships between input and output variables. These methods obviate the challenges associated with model assumptions or predetermined perturbations, thereby exerting a substantial influence on model accuracy and flexibility. For instance, machine-learning-based models, such as autoregressive integrated moving average (ARIMA) [2], support vector regression (SVR) [3], [4], and multilayer perceptron [5], have remained popular in contemporary hydrological research. However, these models are often encumbered by the constraints of stationary assumptions in predicting highly nonlinear nonstationary flood time-series data. Furthermore, traditional machine learning models necessitate manual feature extraction and preprocessing, which can limit their applicability in complex real-world scenarios [6].

Recently, emerging deep learning models have facilitated automated feature extraction and learning from data, curtailing manual interventions and amplifying processing efficacy, such as recurrent neural network (RNN) [7], long short-term memory (LSTM) [8], [9], gated recurrent unit (GRU) [10], bidirectional recurrent neural network (BiGRU) [11], etc. Leveraging the gate units of RNNs, these models can effectively extract temporal features from flood series data. However, spatial features, such as the dispersion of hydrological and meteorological data, fluvial morphology, and edaphic composition, bear significant influences on the genesis and progression of inundations [12]. Remote sensing technology can provide more comprehensive and accurate surface information by acquiring high-resolution satellite images and data from other sensors [13], [14], [15]. This information includes water body distribution, terrain characteristics, soil moisture, and more. Such data are crucial for understanding the spatiotemporal features involved in the formation and development of floods. As shown in Fig. 2, the same amount of rainfall will result in different runoff for stations located in different geographical locations. To effectively exploit spatial features in the time-series forecasting (i.e., spatiotemporal sequence forecasting), a few academics have harnessed convolutional neural networks (CNNs) to derive the adjacency relationships within the spatial domain. For instance, the ConvLSTM model employed by Chen et al. [16] adeptly uses LSTM to extract temporal features, while simultaneously drawing on the CNN to extract the spatial characteristics of grid flood data. Despite their notable accomplishments, these methodologies continue to face significant hurdles in achieving accurate forecasts of flooding.



(a) (b) (c)

Fig. 1. Proportion of flood disasters in losses incurred by human society and economy in the past 30 years due to natural disasters. (a) Proportion of direct economic loss (to GDP) due to natural disasters. (b) Number of people affected/number of deaths and missing. (c) Overall frequency.

First, flood data have irregularity and nonlinearity concerning their spatiotemporal distribution, coupled with discernible autocorrelative and nonsimilar attributes. These features of flood data pose a formidable challenge to the grid-based analysis approach (CNNs). Graph-based methodologies furnish a more proficient recourse to address the intricate data characteristics and facilitate the discernment of spatiotemporal correlations. Nevertheless, the existing graph convolutional network (GCN)-based techniques commonly integrate manually defined adjacency matrices to characterize spatial associations, necessitating detailed priors for optimal model exhibition. This stems

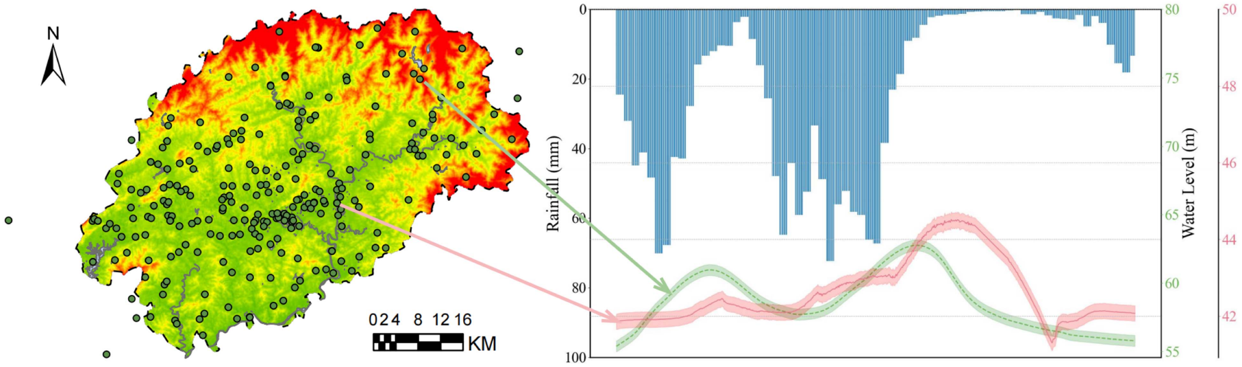


Fig. 2. Effects of spatial features on runoff under same rainfall conditions in Wuyuan County.

from the inherent capacity of graph construction to dictate node relations, hence influencing the feature extraction capabilities of GCNs. A maladaptation in graph structure may result in the distortion or loss of information, thereby exerting a deleterious influence on the prognostic efficiency of model. Moreover, relying on a static graph structure in graph convolution presumes a homogeneity in the spatiotemporal correlations within the data, which seems implausible given the mercurial nature of spatiotemporal data. For instance, alterations to natural factors, such as precipitation, geology, and topography, may exert an impact on river water level, flow velocity, and other relevant attributes, eliciting variations in the connectivity among hydrological stations.

Second, flood spatiotemporal forecasting entails modeling diverse sorts of data using remote sensing technology, such as meteorological, hydrological, topographic, and anthropogenic patterns, among others. These data typically exhibit disparate attributes and structures that defy portrayal through identical graphical frameworks. Moreover, such a corpus often manifests multitier correlations, spanning across diverse spatial and temporal spheres, thereby mandating modeling at various scales. Predictive frameworks that utilize heterogeneous graphs can adeptly account for disparate data relationships and features based on heterogeneous nodes and edges, thereby conferring superior adaptability and scalability, culminating in enhanced precision of flood prediction. Consequently, another pivotal challenge in the realm of flood spatiotemporal prediction lies in constructing models for spatiotemporal data based on heterogeneous graph to comprehensively consider multiple data types and provide richer data information for accurate flood prediction.

To overcome these challenges, we propose a novel deep learning model, namely, heterogeneous dynamic temporal graph convolutional network (HD-TGCN), which is designed to overcome the limitations of the existing methods by seamlessly integrating remote sensing data into the forecasting framework. The HD-TGCN not only leverages the adaptability of graph structures but also accommodates heterogeneous data types encompassing meteorological, hydrological, topographic, and anthropogenic patterns, among others. This enables the model to capture the multifaceted correlations inherent in flood dynamics across diverse spatial and temporal scales. The main contributions of this article are summarized as follows.

1) We design a dynamic temporal graph convolutional module (D-TGCM) that creatively introduces self-attention mechanisms to generate a self-adaptive adjacency matrix for extracting spatiotemporal dynamics in flood data.
2) We propose an HD-TGCN, which utilizes multiple parallel GCNs to process heterogeneous graphs and uses fusion mechanisms to capture complex relationships and dynamic changes between heterogeneous graphs.
3) We evaluate our proposed model on flood datasets and outperform the state-of-the-art flood prediction models in the mean absolute error (MAE), Nash–Sutcliffe efficiency (NSE), and root-mean-square error (RMSE) with improvements of 80.32%, 0.15%, and 73.99%, respectively.

The rest of this article is organized as follows. In Section II, we summarize the related work on flood prediction and GCNs. Section III introduces the preliminary concepts. In Section IV, we describe the technical details of our proposed HD-TGCN model. Subsequently, in Section V, we evaluate the performance of our proposed model on real-world datasets. Finally, Section VI concludes this article.

II. Related Work

In this section, we will summarize the literature works and provide the motivation for our research.

A. *Machine Learning for Flood Forecasting*

Flood forecasting constitutes a fundamental and paramount problem within the realm of smart hydrology, which has garnered significant attention from scholars over the past several decades [17]. In recent years, the integration of remote sensing data has emerged as a promising avenue to enhance the accuracy and reliability of flood forecasting models. While earlier studies utilized prototypical machine learning approaches, such as history average (HA), ARIMA, SVR, and vector autoregressive (VAR), to account for linear interdependencies among multiple time series [4], [18], the incorporation of remote sensing data introduces a new dimension of information that can capture the complex spatiotemporal dynamics inherent in flood events. However, these conventional time-series models pose rigorous assumptions regarding data stationarity, linearity, and normality and are susceptible to data noise, rendering their capacity to predict highly nonlinear flood data. More recently, given the remarkable efficiency of deep learning in automated feature extraction or representation learning, extensive research has been devoted to applying deep-learning-based methodologies for flood prediction. The authors of [8], [12], and [19] applied the LSTM network model for flood forecasting. Chen et al. [11] explored the BiGRU-based encoder–decoder framework for multistep-ahead flood forecasting. In addition, the fusion of remote sensing data with machine learning methods, such as stacked autoencoders and RNN, has been proposed to enhance flood inundation forecasts [20]. CNNs have also been applied to extract the spatial characteristics of hydrological data. Chen et al. [16] converted the river network to a regular 2-D grid and applied a CNN to model the spatial dependencies of hydrological data. Khosravi et al. [21] prepared a flood susceptibility map of Iran using a deep CNN algorithm.

B. Graph Convolution for Prediction

While CNN-based models can extract the spatiotemporal characteristics of hydrological data, they are restricted to processing Euclidean data. GCNs were first introduced in [22], which bridge the spectral graph theory and deep neural networks. The diffusion convolutional recurrent neural network [23] and the spatial–temporal synchronous graph convolutional network [24] are constructed based on the GCN to extract the spatial feature of traffic networks, which improves the representation of spatial correlations among traffic flows [25]. Yan et al. [26] proposed the spatiotemporal graph convolutional network (STGCN), which leverages the GCN to model and recognize dynamic human skeletal movements and learn spatial and temporal patterns in skeletal dynamic data. Seo et al. [27] proposed the graph convolutional recurrent network to predict time-series data on graph structures, which has been validated in video prediction and natural language processing domains. While the existing graph convolutional models often rely on expert-defined static graphs based on real geographic sensor relations, the dynamic nature of remote sensing data necessitates a more adaptable approach.

Our method is different from all those methods. We model the sensor network as a dynamic graph based on the input data, which is more consistent with the features of flood prediction than grid and static graphs. Moreover, we design a parallel GCN predictive framework to handle heterogeneous graphs, as the formation of floods is influenced by various exogenous factors, such as meteorological conditions including precipitation. The integration of remote sensing data within both the machine learning and graph convolution frameworks presents a significant advancement in flood forecasting. However, computational costs effectively remain a challenge, yet the benefits outweigh these tradeoffs in flood prediction and management.

III. Preliminaries

In this section, we present a comprehensive analysis of the key concepts and methodologies that form the basis of our research. We begin by introducing the concept of GCNs, which have emerged as a powerful tool for analyzing graph-structured data. Then, we present the key principles of multihead attention (MHA), a fundamental technique that has shown remarkable success in modeling sequential data [28], [29]. To facilitate comprehension, we present all the symbols and their respective meanings in the Nomenclature.

A. Graph Convolutional Network

GCNs are a type of neural network designed to work with graph data. A graph is typically represented as $G = (V, E)$, where V represents the set of nodes, $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, N is the number of nodes, and E represents the set of edges. Each node represents an entity, and each edge represents the relationship between the entities. The main idea behind the GCN is to learn the node embedding by aggregating and transforming the features of the node and its neighbors through a series of convolution operations.

1) Spectral Graph Convolution: According to the convolution theorem, given a signal x as input and the other signal y as filter, graph convolution $*_G$ could be written as

$$x *_{\mathrm{G}} y = U\left(\left(U^T x\right) \odot \left(U^T \mathbf{y}\right)\right) \tag{1}$$

where the convolution filter in the spectral domain is $U^T y$. Let $U^T y = [\theta_0, \ldots, \theta_{n-1}]^T$ and $g_\theta = \mathrm{diag}[\theta_0, \ldots, \theta_{n-1}]$; (1) is expressed as

$$x *_{\mathrm{G}} y = U g_\theta U^T x \tag{2}$$

where $U^T x$ is the graph Fourier transform, $g_\theta U^T x$ is a convolution in the spectral domain, and $U g_\theta U^T x$ is the inverse graph Fourier transform.

$U^T = [u_1, \ldots, u_n]^T$ is the complete set of orthonormal eigenvectors $\{u_l\}_{l=1}^n$ of L, ordered by its nonnegative eigenvalues $\{\lambda_l\}_{l=1}^n$, i.e.,

$$L = U \Lambda U^T \tag{3}$$

$$\Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}. \tag{4}$$

Therefore

$$\begin{aligned} g_\theta &= \mathrm{diag}\left[\theta_0, \ldots, \theta_{n-1}\right] \\ &= g_\theta(\Lambda) = \theta_0 \Lambda^0 + \theta_1 \Lambda^1 + \cdots + \theta_n \Lambda^n. \end{aligned} \tag{5}$$

Equation (2) is expressed as

$$x *_{\mathrm{G}} y = U g_\theta(\Lambda) U^T x. \tag{6}$$

However, there are some drawbacks to performing spectral graph convolution according to (6):

1) requiring eigendecomposition of Laplacian matrix; eigenvectors are explicitly used in convolution;

2) high computational cost, multiplication with the eigenvector matrix U is $O(n^2)$;
3) not localized in the vertex domain.

To address these issues, Hammond et al. [30] proposed that a truncated expansion using Chebyshev polynomials $T_k(x)$ up to the Kth order can be used to obtain a good approximation of $g_\theta(\Lambda)$

$$\begin{aligned} x *_{\mathrm{G}} y &\approx U \sum_{k=0}^{K-1} \beta_k \Lambda^k U^T x \\ &= \sum_{k=0}^{K-1} \beta_k U \Lambda^k U^T x \\ &= \sum_{k=0}^{K-1} \beta_k L^k x. \end{aligned} \tag{7}$$

Note that the number of free parameters reduces from n to K, computational cost is reduced from $O(n^2)$ to $O(|E|)$, and the convolution is K-localized in the vertex domain.

2) Graph Convolutional Networks: In the GCN, the convolution operator is defined based on the Laplacian matrix of the graph. The Laplacian matrix is a matrix derived from the adjacency matrix. The Laplacian matrix is defined as

$$L = D - A \tag{8}$$

where A is the adjacency matrix of the graph, which is used to represent the connection between nodes. D is the degree matrix, $D_{ii} = \sum_j A_{ij}$

$$L_{\mathrm{sym}} = D^{-1/2} L D^{-1/2}. \tag{9}$$

L_{sym} has n nonnegative eigenvalues corresponding to n mutually orthogonal eigenvectors. To avoid gradient explosion, it is required that $x \in [-1, 1]$ in the Chebyshev polynomials $T_k(x)$, i.e., the eigenvalues of the Laplacian matrix $\in [-1, 1]$. The eigenvalues of $L_{\mathrm{sym}} \in [0, 2]$; therefore, L_{sym} needs to be rescaled to

$$\tilde{L} = \frac{2}{\lambda_{\max}} L_{\mathrm{sym}} - I_N \tag{10}$$

where $\lambda_{\max}$ is the maximum eigenvalue of L_{sym}. Limiting the number of Chebyshev polynomials to $K = 2$, a linear function can be obtained. In this linear formulation of a GCN, we further approximate $\lambda_{\max} \approx 2$. Under these approximations, (7) simplifies to

$$\begin{aligned} x *_{\mathrm{G}} y &\approx \beta_0 x + \beta_1 \left(L_{\mathrm{sym}} - I_N\right) x \\ &= \beta_0 x + \beta_1 \left(D^{-1/2}(D - A) D^{-1/2} - I_N\right) x \\ &= \beta_0 x - \beta_1 \left(D^{-1/2} A D^{-1/2}\right) x. \end{aligned} \tag{11}$$

Let $\beta_0 = -\beta_1 = \beta$; then, (11) is further simplified as

$$\begin{aligned} x *_{\mathrm{G}} y &= \beta \left(I_N + D^{-1/2} A D^{-1/2}\right) x \\ &= \beta \left(\widetilde{D}^{-1/2} \widetilde{A} \widetilde{D}^{-1/2}\right) x \end{aligned} \tag{12}$$

where $\widetilde{A} = A + I_N$ and $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$.

B. MHA Mechanism

The MHA mechanism serves as a fundamental operation in the model, which is widely utilized in deep learning models, particularly in tasks related to natural language processing and sequence modeling [31], to enhance the modeling capability of sequential data by capturing dependencies. An input sequence is represented as a matrix $X \in \mathbb{R}^{n \times d}$, where n denotes the sequence length, and d represents the dimension of each element in the sequence.

Given the input sequence, the MHA mechanism involves transforming the input matrix X into multiple sets of queries (Q), keys (K), and values (V). These transformations are expressed as linear projections

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V \tag{13}$$

where the weight matrices W^Q, W^K, and W^V are learnable. Next, attention weights are independently computed for each attention head. For each head i, the attention weights A_i are calculated by taking the dot product between Q and K, scaled by the square root of the key dimension

$$A_i = \mathrm{softmax}\left(\frac{QW_i^Q \cdot KW_i^K}{\sqrt{d_k}}\right) \tag{14}$$

where d_k represents the dimension of the keys. The attention weights A_i are then utilized to weight the corresponding values V_i, resulting in contextual representations for each attention head

$$\mathrm{head}_i = A_i V_i. \tag{15}$$

Finally, the contextual representations from all the attention heads are concatenated and linearly combined to obtain the final contextual representation

$$\begin{aligned} &\mathrm{MultiHead}(Q, K, V) \\ &\quad = \mathrm{Concatenate}\ (\mathrm{head}_1, \mathrm{head}_2, \ldots, \mathrm{head}_h)\, W^O \end{aligned} \tag{16}$$

where h represents the number of attention heads, and W^O denotes the learnable weight matrix for linear combination.

By incorporating multiple attention heads, the MHA mechanism enables the model to capture diverse aspects and relationships within the input sequence. The parallel processing of attention heads allows effective attention to different parts of the sequence, facilitating the extraction of relevant information and enhancing the capabilities of model for tasks such as time-series prediction.

IV. Our Approach: HD-TGCN

A. Problem Definition

Definition 1 (Heterogeneous dynamic graph): The network of stations within a watershed is defined as a heterogeneous dynamic graph $\mathcal{G}_t = \{\mathcal{G}_t^1, \mathcal{G}_t^2, \ldots, \mathcal{G}_t^C\}$, consisting of C dynamic graphs, and the network structure evolves over time. $\mathcal{G}_t^c = (\mathcal{V}_t^c, \mathcal{E}_t^c, \mathbf{W}_t^c)$ represents the cth dynamic graph, where $\mathcal{V}_t^c$

is a set of $|\mathcal{V}_t^c| = N_c$ nodes representing monitor stations, including those equipped with remote sensing devices. $\mathcal{E}_t^c$ represents the set of edges in the graph, while $\mathbf{W}_t^c \in \mathbb{R}^{N_c \times N_c}$ is a weighted adjacency matrix used to denote the connectivity relationships among the N_c nodes in the cth dynamic graph.

Definition 2 (Heterogeneous spatiotemporal matrix): At each time step t, the observations on the graph $\mathcal{G}_t$ are denoted by a heterogeneous spatiotemporal matrix $\mathcal{X}_t = (\mathbf{X}_t^1, \mathbf{X}_t^2, \ldots, \mathbf{X}_t^C)$. $\mathbf{X}_t^c = (\mathbf{x}_t^{c,1}, \mathbf{x}_t^{c,2}, \ldots, \mathbf{x}_t^{c,N_c}) \in \mathbb{R}^{N_c \times M}$ represents the spatiotemporal matrix on the cth dynamic graph, which includes the data collected by monitor stations equipped with various sensors, including remote sensing devices, where $\mathbf{x}_t^{c,n}$ denotes the feature vector on the nth node, and M is the number of features.

Provided the graph $\mathcal{G}_t$ and its historical T_h step heterogeneous spatiotemporal matrices $\{\mathcal{X}_{t-T_h+1}, \mathcal{X}_{t-T_h+2}, \ldots, \mathcal{X}_t\}$, flood forecasting aims to learn a mapping function $f(\cdot)$ that transforms the historical spatiotemporal matrices, including data from remote sensing devices, into future spatiotemporal matrices $\mathcal{Y} = \{\mathcal{Y}^1, \mathcal{Y}^2, \ldots, \mathcal{Y}^F\} \in \mathbb{R}^{F \times T_p}$ of all the objective nodes over the next T_p step, where $\mathcal{Y}^f = (\mathcal{Y}_{t+1}^f, \mathcal{Y}_{t+2}^f, \ldots, \mathcal{Y}_{t+T_p}^f)$ denotes the future hydrological values of node f from t. The mapping relation is represented as follows:

$$\mathcal{Y}_{t+1:t+T_p} = f(\mathcal{X}_{t-T_h+1:t}; \mathcal{G}_t). \tag{17}$$

B. Dynamic Temporal Graph Convolution Module

In many existing architectures, the graph convolution has been served as a key module in modeling non-Euclidean data. The weighted adjacency matrix, as a fundamental component of graph convolution, is typically manually defined based on expert knowledge to quantify the relationships between nodes. However, this practice not only significantly compromises the efficiency of GCNs but also leads to decreased accuracy due to the temporal dynamics inherent in graph structures. To efficiently and accurately extract spatiotemporal features of flood sequences, we adopt an attention mechanism to automate the construction of dynamic adjacency matrices in the D-TGCM. By leveraging the inherent dynamics of the graph, we aim to fully exploit the spatiotemporal characteristics. The dynamic adjacency matrix is divided into two distinct components: the static subgraph, denoted as $\mathcal{G}_s$, and the dynamic subgraph, denoted as $\mathcal{G}_t$. The construction of $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s, \mathbf{W}_s)$ is based on the physical locations of the stations within the basin using a thresholded Gaussian kernel [32]

$$w_s^{(i,j)} = \begin{cases} \exp\left(-\frac{d(v^i,v^j)^2}{\sigma^2}\right) & ,i \neq j \text{ and } \exp\left(-\frac{d(v^i,v^j)^2}{\sigma^2}\right) \geq \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where v^i is the ith node, $d(v^i, v^j)$ represents the distance from station v^i to station v^j according to haversine formula [33], while $w_s^{(i,j)}$ denotes the edge weight between station v^i and station v^j. σ^2 and ϵ determine the distribution and sparsity of matrix, respectively:

$$\text{HAVERSINE}\left(\frac{d}{R}\right) = \text{haversin}\left(\varphi_2 - \varphi_1\right) + \cos\left(\varphi_1\right)\cos\left(\varphi_2\right) \cdot \text{haversin}(\Delta\lambda) \tag{19}$$

$$\text{haversin}\left(\theta\right) = \sin^2(\theta/2) = (1 - \cos(\theta))/2 \tag{20}$$

where R represents the radius of the earth, which can be taken as an average of 6371.393 km, φ_1 and φ_2 represent the latitudes of two points, and $\Delta\lambda$ represents the difference in longitudes between two points.

The weight matrix W_t^c of the cth dynamic graph $\mathcal{G}_c$ is derived through the utilization of the multihead self-attention (MHSA) mechanism, where the MHSA mechanism enables the model to learn intricate dependencies between each individual feature and all other features within the graph. It offers a larger receptive field compared to the conventional MHA mechanism

$$\widehat{\mathbf{W}_t^c} = \text{MultiHead}(\mathbf{X}_{t-T_h+1:t}^c, \mathbf{X}_{t-T_h+1:t}^c, \mathbf{X}_{t-T_h+1:t}^c) \tag{21}$$

where $\widehat{\mathbf{W}_t^c} \in \mathbb{R}^{T_h \times N_c}$, and the computation complexity of MHSA is $O(T_h^2 Mh)$. After the diagonalization operation

$$w_t^c(i,j,k) = \begin{cases} \widehat{w_t^c}(i,j), & j = k \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

The matrix $\widehat{\mathbf{W}_t^c}$ is transformed into $\mathbf{W}_t^c \in \mathbb{R}^{T_h \times N_c \times N_c}$. This process expands the second dimension of $\widehat{\mathbf{W}_t^c}$, creating a new matrix that allows for a more detailed representation of the relationships between elements in the spatiotemporal matrix. Based on this, we can obtain the dynamic Laplacian matrix $\mathbf{L}_t^c$ of the cth dynamic graph at time t

$$\mathbf{L}_t^c = \frac{1}{N_c}\mathbf{W}_t^c + \mathbf{W}_s^c + \mathbf{I}_{N_c} \tag{23}$$

where $\mathbf{W}_s^c \in \mathbb{R}_{N_c \times N_c}$ is the static weight matrix constructed from the nodes in the cth dynamic graph, and $\mathbf{I}_{N_c}$ is the identity matrix of dimension N_c. Specifically, we have normalized the graph Laplacian $\widehat{\mathbf{L}_t^c}$ as

$$\widehat{\mathbf{L}_t^c} = \left(\widetilde{\mathbf{D}_t^c}^{-1/2}\widetilde{\mathbf{L}_t^c}\widetilde{\mathbf{D}_t^c}^{-1/2}\right) \tag{24}$$

where $\widetilde{\mathbf{D}_t^c}$ is the diagonal degree matrix of $\widehat{\mathbf{L}_t^c}$ with $\widetilde{\mathbf{D}^c}_{t(ii)} = \sum_j \widetilde{\mathbf{L}^c}_{t(ij)}$.

Intuitively, the elements $\widehat{\mathbf{L}_t^c}(ij)$ of the matrix $\widehat{\mathbf{L}_t^c}$ can simultaneously characterize the spatiotemporal correlation between node i and node j in the cth dynamic graph. Notably, this correlation exhibits a dynamic nature, thus aligning appropriately with the inherent spatiotemporal heterogeneity encountered in the domain of flood prediction. Such heterogeneity arises due to the ever-changing characteristics of the riverbed substrate, necessitating an adaptive representation of the evolving spatiotemporal relationships. Algorithm 1 demonstrates the procedure to create a dynamic Laplacian matrix for the D-TGCM.

Then, we use the D-TGCM based on the two-layer temporal graph convolutional network (T-GCN) model [34] to learn

Algorithm 1: CREATELAPLACIAN: Create a Dynamic Laplacian Matrix for the D-TGCM.

```
Input: The latitude and longitude coordinates for $N_c$ nodes
    in the $c$th dynamic graph $(\mathbf{lat}, \mathbf{lng})$; Historical flood data
    sequence $\{\mathbf{X}^c_{t-T_h+1}, \mathbf{X}^c_{t-T_h+2}, \ldots, \mathbf{X}^c_t\}$; Historical
    sequence length $T_h$; Variance $\sigma^2$; Threshold $\epsilon$; The
    number of heads in MHSA $h$.
Output: Dynamic Laplacian Matrix: $\mathbf{L}^c_t$.
// Create static matrix
for int $i$, $j$=1 to $N_c$ do
    $d(v^i, v^j) = \text{HAVERSINE}(\mathbf{lat}, \mathbf{lng})$;
    if $i \neq j$ and $\exp(-\frac{d(v^i,v^j)^2}{\sigma^2}) \geq \epsilon$ then
        $w_s^{(i,j)} = \exp(-\frac{d(v^i,v^j)^2}{\sigma^2})$;
    else
        $w_s^{(i,j)} = 0$;
    end if
end for
// Create dynamic matrix
$Q, K, V = \mathbf{X}^c_{t-T_h+1:t}$;
$head_i = \text{softmax}(\frac{QW_i^Q \cdot KW_i^K}{\sqrt{d_k}})V_i$;
$\widehat{\mathbf{W}^c_t} =$ Concatenate $(head_1, head_2, \ldots, head_h)W^O$;
for int $i$, $j$, $k$=1 to $N_c$ do
    if $j$=$k$ then
        $w^c_t(i, j, k) = \widehat{w^c_t}(i, j)$;
    else
        $w^c_t(i, j, k) = 0$;
    end if
end for
$\mathbf{L}^c_t = \frac{1}{N_c}\mathbf{W}^c_t + \mathbf{W}^c_s + \mathbf{I}_{N_c}$.
```

spatiotemporal features from sensors data

$$f_c\left(\mathcal{X}^c, \mathbf{L}^c\right) = \text{softmax}\left(\widehat{\mathbf{L}^c}\,\text{ReLU}\left(\widehat{\mathbf{L}^c}\mathcal{X}^c W^{(0)}\right) W^{(1)}\right) \tag{25}$$

$$u_t = \sigma\left(W_u\left[f_c\left(\mathcal{X}^c_t, \mathbf{L}^c_t\right), h_{t-1}\right] + b_u\right) \tag{26}$$

$$r_t = \sigma\left(W_r\left[f_c\left(\mathcal{X}^c_t, \mathbf{L}^c_t\right), h_{t-1}\right] + b_r\right) \tag{27}$$

$$c_t = \tanh\left(W_c\left[f_c\left(\mathcal{X}^c_t, \mathbf{L}^c_t\right), \left(r_t * h_{t-1}\right)\right] + b_c\right) \tag{28}$$

$$h_t = u_t * h_{t-1} + (1 - u_t) * c_t. \tag{29}$$

Each execution of the graph convolution operation requires matrix multiplication, which results in a complexity of $O(N^2)$, where N is the number of nodes. Fig. 3 illustrates the architecture of our D-TGCM, which can extract the dynamic spatiotemporal features. At each time step, the module demonstrates an autoregressive behavior, wherein the previously generated data are leveraged as an additional input to produce the next sequence of data.

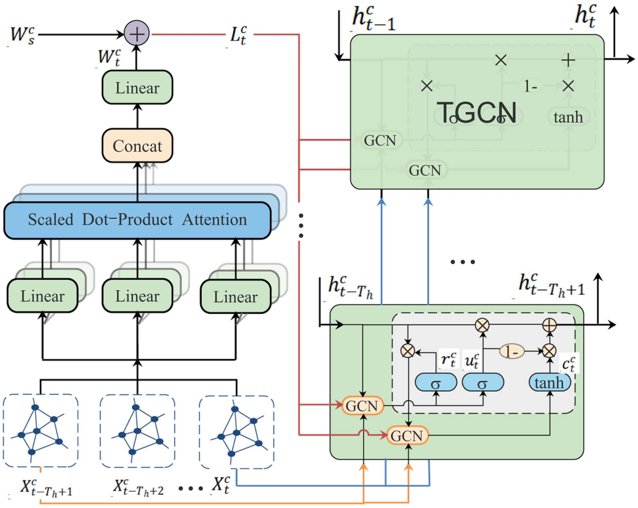


Fig. 3. Architecture of the D-TGCM. The T-GCN utilizes a dynamic Laplacian matrix generated by an MHA mechanism to capture the spatiotemporal dynamics of input features. The module is autoregressive and generates data in a sequential manner, where each step takes into account the previously generated data.

C. Heterogeneous Dynamic Temporal Graph Convolutional Network

To accurately predict the occurrence and development trends of floods, it is necessary to combine multiple influencing factors such as hydrology and meteorology, using a multivariate prediction method [35]. The multivariate prediction method takes multiple related variables as model inputs and considers the interrelationships between variables to establish a prediction model for the target variable. This method can greatly reduce prediction errors caused by omitting key factors and better reflect the complexity and diversity of the real world. In flood prediction, multivariate prediction can comprehensively consider the influence of various factors such as rainfall, evapotranspiration, and runoff, including their interactions and time-delay characteristics, to more accurately predict important parameters such as flood occurrence time, frequency, and intensity.

However, different sensors typically produce graphs with different structures, and traditional T-GCN models cannot effectively integrate and process data among all the sensors, resulting in inaccurate predictions. In addition, different sensors may have complex spatial and temporal relationships, and if physically close but different sensors are assigned to the same subgraph, it can diminish the accuracy and reliability of the prediction. To address these issues, we propose the HD-TGCN model based on the D-TGCM, which can perform topology learning and sequence prediction on heterogeneous graphs. The overall framework of the HD-TGCN model is shown in Fig. 4.

The sensor network within the watershed is defined as C dynamic heterogeneous undirected graphs $\mathcal{G} = \{\mathcal{G}^1, \ldots, \mathcal{G}^C\}$ based on the classification of different sensor types. Each dynamic undirected graph $\mathcal{G}^c$ is associated with a heterogeneous spatiotemporal matrix $\mathcal{X}^c$ corresponding to the collected data by the sensor network. Subsequently, each group of dynamic

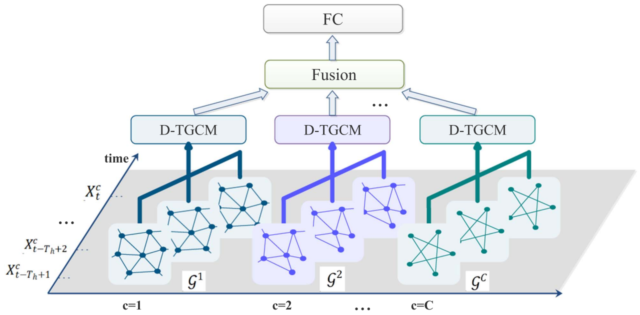


Fig. 4. Framework of the HD-TGCN.

undirected graphs $\mathcal{G}^c$ and heterogeneous spatiotemporal matrices $\mathcal{X}^c$ are fed into the D-TGCM to extract dynamic spatiotemporal features. As shown in Fig. 3, the D-TGCM is a model integrating spatiotemporal graph convolution and GRU designed to handle dynamic spatiotemporal data and extract spatiotemporal features. After extracting spatiotemporal features for each sensor network, the features are concatenated and input into a fully connected (FC) layer for comprehensive feature learning and processing, enabling the global extraction of dynamic spatiotemporal features for the entire sensor network. The computational complexity of our method is $O(T_h^2 Mh + N^2)$. With this approach, the HD-TGCN effectively addresses complex sensor network data, learns spatiotemporal relationships among sequence data, and achieves accurate flood prediction. Moreover, the model exhibits robust generalization ability and effectively handles heterogeneous sensor network data. Algorithm 2 characterizes the training steps for our model.

V. Experiments

In this section, we first present an overview of the datasets, evaluation metrics, baseline methods, and implementation details. Then, we show the experimental results to demonstrate the proficiency of our proposed model.

A. Datasets

The *Wuyuan* dataset includes hydrological and meteorological data collected by the Hydrological Monitoring System of Small and Medium Rivers in Jiangxi Province, as well as remote sensing data. The data are available in CSV format from Oracle Database. Wuyuan is located between 117°22′ to 118°11′ E and 29°01′ to 29°35′ N, with a total land area of 2947.064 km^2, mainly consisting of low hills and mountains. Mountains and hills account for more than 83% of the total area. Due to the slope of the mountain rivers and the rapid rate of confluence, Wuyuan is prone to flash floods. The monsoon period is from April to July, with a monthly rainfall of 200 to 300 mm, accounting for 47.9% of the annual rainfall. The water level reaches its peak in June and July each year. Therefore, the data from the months of June and July from 2020 to 2022 were selected in our experiments. The data were preprocessed according to [16], aggregated with a time interval of 5 min, which means that 12 sequence data represent 1 h. There are 18 hydrological stations and 164 rainfall stations distributed throughout the basin, and these two types of sensor networks are defined as two heterogeneous undirected graphs.

Algorithm 2: Training Algorithm of the HD-TGCN Framework.

```
Input: Flood data $\mathcal{X} = [\mathcal{X}_1, \ldots, \mathcal{X}_{N_t}]$; Heterogeneous
   dynamic graph $\mathcal{G} = \{\mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^C\}$; Predictive
   sequence length $T_p$;
Output: Parameters of HD-TGCN Prediction Model $\omega$.
  // Construct samples
  $\varnothing \to \Gamma$
  for int $t$=1 to $N_t$ do
      for int $c$=1 to $C$ do
          $\hat{\mathcal{X}}^c \leftarrow [\mathbf{X}_{t-T_h+1}^c, \mathbf{X}_{t-T_h+2}^c, \ldots, \mathbf{X}_t^c]$;
      end for
      $\hat{\mathcal{Y}}^{N_F} \leftarrow [\mathbf{X}_{t+1}^{N_F}, \mathbf{X}_{t+2}^{N_F}, \ldots, \mathbf{X}_{t+T_p}^{N_F}]$;
      // $N_F$ is the set of target nodes for prediction;
      $\Gamma \leftarrow (\hat{\mathcal{X}}^c, \hat{\mathcal{Y}}^{N_F}, \mathcal{G}^c)$;
end for
  // Train model parameters
while Model not converged do
      for int $c$=1 to $C$ do
          $\mathcal{L}^c$ = CREATELAPLACIAN($\hat{\mathcal{X}}^c$);
          for int $t$=1 to $T_h$ do
              $\mathcal{Y}_t^c, h_t^c$ = D-TGCM($\hat{\mathcal{X}}^c, \mathcal{L}^c$);
          end for
          $\mathcal{Y}^{N_F} =$
          $softmax($ Concatenate $(\mathcal{Y}^1, \mathcal{Y}^2, \ldots, \mathcal{Y}^C))$;
      end for
       $\mathrm{L}_{Train} = loss(\hat{\mathcal{Y}}^{N_F}, \mathcal{Y}^{N_F})$;
      $\omega = \omega - \alpha \nabla_\omega \mathrm{L}_{\text{train}}$ ;
end while
```

B. Evaluation Metrics and Baseline Methods

To evaluate the performance of our model, we utilize the MAE and RMSE as evaluation metrics. Furthermore, we use NSE, which is an assessment metric typically utilized to verify hydrological model simulation results and ranges from $-\infty$ to 1. A higher NSE value implies better model performance.

1) MAE

$$\mathrm{MAE} = \frac{1}{T}\sum_{t=1}^{T} \left| \mathcal{Y}_{\text{real}}^t - \mathcal{Y}_{\text{pred}}^t \right| . \tag{30}$$

2) RMSE

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum\nolimits_{t=1}^{T} \left(\mathcal{Y}_{\text{real}}^t - \mathcal{Y}_{\text{pred}}^t \right)^2} . \tag{31}$$

3) NSE

$$\mathrm{NSE} = 1 - \frac{\sum_{t=1}^{T} \left(\mathcal{Y}_{\text{real}}^t - \mathcal{Y}_{\text{pred}}^t \right)^2}{\sum_{t=1}^{T} \left(\mathcal{Y}_{\text{real}}^t - \overline{\mathcal{Y}}_{\text{real}} \right)^2} . \tag{32}$$

TABLE I
PARAMETERS OF MODEL

Parameters	**Value**
Learning rate	1e-3
Epoch	100
Batch size	32
Hidden unit	64
Optimizer	Adam
C	2
N_1	18
N_2	164
F	17
N_t	52704
σ^2	0.1
ϵ	0.1

Here, $\mathcal{Y}_{\text{pred}}^t$ and $\mathcal{Y}_{\text{real}}^t$ mean the forecasted runoff and its ground truth, respectively. $\overline{\mathcal{Y}_{\text{real}}}$ represents the mean value of ground truth during the forecast period T.

We compare our model with the following baseline methods. The benchmark models are used to predict water level for the next 15, 30, and 60 min.

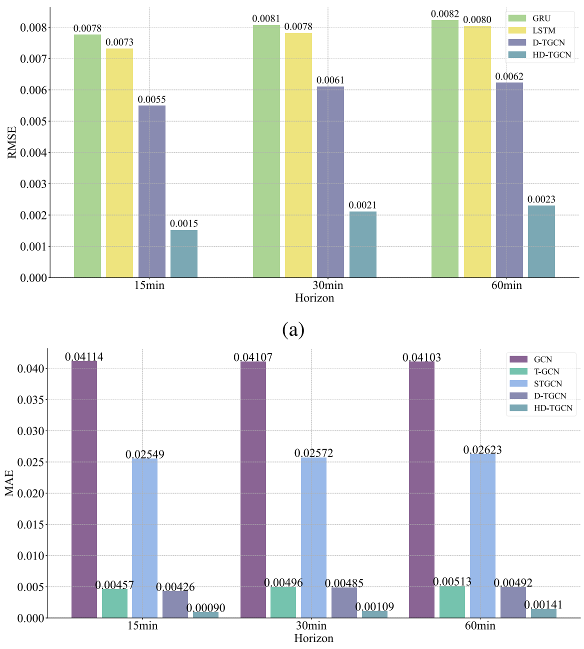
1) *ARIMA [2]:* This model is a parametric model that uses the autocorrelation and trend of the time series to predict future values.
2) *VAR [36]:* This is a multivariate forecasting technique that captures the linear relationship between variables and uses their lagged values as prediction.
3) *GRU [37]:* This is a special RNN model.
4) *LSTM [38]:* This network is a well-known time-series prediction model based on deep learning.
5) *GCN [39]:* This is a localized first-order approximation of spectral graph convolutions.
6) *T-GCN [34]:* The T-GCN model is in combination with the GCN and the GRU.
7) *STGCN [40]:* This network uses ChebNet in the spatial dimension and 2-D convolutional networks in the temporal dimension to model the spatial–temporal correlations in graph data.

C. Implementation Details

Our proposed model is implemented on a NVIDIA RTX 3090 using Torch 1.12.1. We split the dataset chronologically in the ratio of 8:2 to generate a nonoverlapped train and test set and also split 30% samples from the train set as the validation set to avoid overfitting. We train for 100 epochs using Adam [41] with a learning rate of 0.001. Each baseline algorithm is executed five times to enhance robustness. The model is trained with batch size 32, and the hidden unit of each layer is 64. The hyperparameters of the model are listed in Table I.

D. Results

Table II presents the experimental results of our proposed model and other baseline models on the *Wuyuan* dataset under different forecasting horizons (i.e., 15, 30, and 60 min). The bold values in the table indicate the best performance, while the underlined values indicate the second-best performance. From Table II, we can see that our proposed model outperforms all other baseline models in all the evaluation metrics for all the forecasting horizons.



(a)

(b)

Fig. 5. (a) Comparison of spatiotemporal- and temporal-based models for flood prediction. (b) Comparison of spatiotemporal- and spatial-based models for flood prediction

Furthermore, among all the deep-learning-based time-series prediction models, including GRU, LSTM, T-GCN, and our proposed HD-TGCN model, they all perform better than traditional machine learning models, such as HA, ARIMA, and VAR. For instance, when forecasting over a 15-min horizon, our model outperformed the ARIMA model, presenting a 99.96% decrease in the RMSE and a 99.97% decrease in the MAE, along with a 3.2% improvement in NSE. Similarly, our model showed a notable decrease of 99.99% and 99.97% in the RMSE and the MAE, respectively, and a 14.48% improvement in NSE compared to the VAR model. This is mainly because such machine learning models assume that the time-series data are stationary when mining the time-series features. However, flood sequence data are often complex, nonstationary, and highly nonlinear, which makes it difficult for traditional machine learning models to accurately capture its inherent patterns. In contrast, deep-learning-based models, such as GRU, LSTM, and our proposed model, can better capture and extract the complex features of nonstationary time-series data, which improve the performance in forecasting flood.

To validate the importance of temporal and spatial features for flood prediction, we conducted two comparisons as follows.

1) *Comparison between the spatiotemporal models, i.e., HD-TGCN and D-TGCM, and the temporal-based models, i.e., GRU and LSTM:* As shown in Fig. 5(a), the performance

TABLE II
PREDICTION RESULTS OF OUR MODELS AND OTHER BASELINE METHODS ON THE *WUYUAN* DATASETS

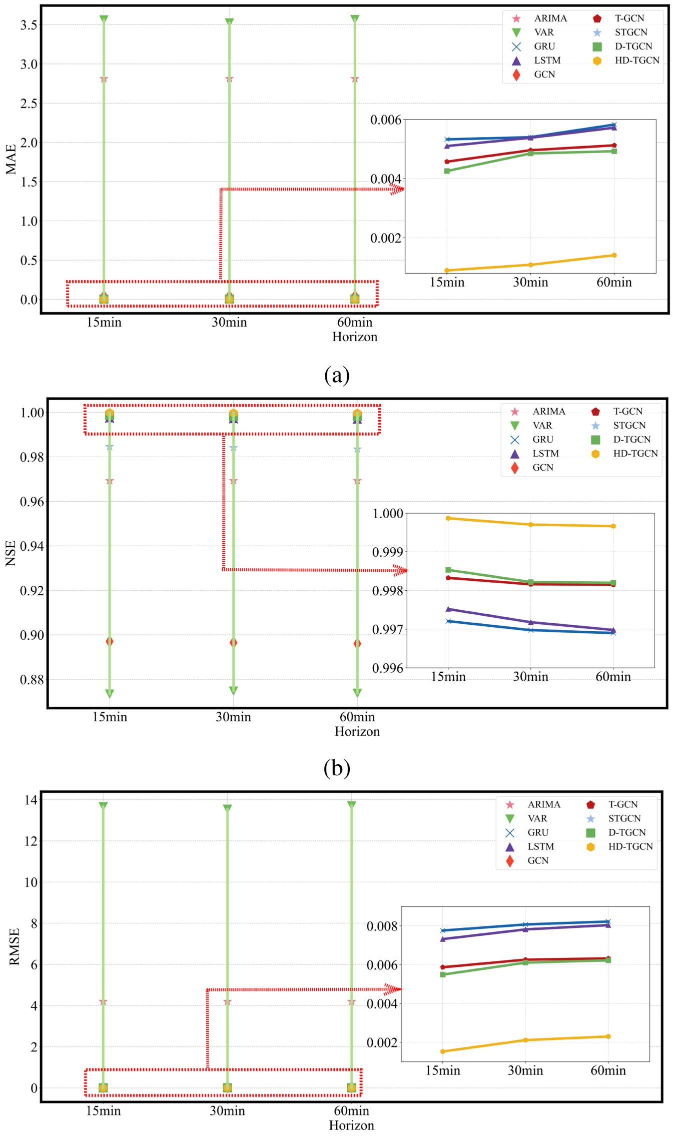
Baselines Horizon	 Metrics	ARIMA	VAR	GRU	LSTM	GCN	T-GCN	STGCN	D-TGCN*	HD-TGCN*
15min	MAE	2.81168	3.56174	0.00533 ± 3.8E-4	0.00510 ± 4.7E-4	0.04114 ± 1.2E-4	0.00457 ± 1.2E-4	0.02549 ± 5.1E-4	0.00426 ± 1.2E-4	**0.00090 ± 1.5E-5**
	NSE	0.96929	0.87341	0.99721 ± 2.8E-4	0.99752 ± 2.4E-4	0.89707 ± 1.0E-3	0.99833 ± 8.4E-4	0.98457 ± 2.2E-4	0.99853 ± 7.4E-4	**0.99987 ± 4.7E-5**
	RMSE	4.19854	13.65987	0.00776 ± 3.9E-4	0.00732 ± 3.3E-4	0.04718 ± 2.5E-4	0.00587 ± 1.4E-4	0.03047 ± 1.5E-4	0.00549 ± 1.4E-4	**0.00153 ± 9.1E-5**
30min	MAE	2.81168	3.52334	0.0054 ± 7.7E-4	0.00538 ± 5.5E-4	0.04107 ± 1.2E-4	0.00496 ± 6.1E-4	0.02572 ± 3.2E-4	0.00485 ± 1.2E-4	**0.00109 ± 6.2E-5**
	NSE	0.96929	0.87475	0.99698 ± 4.7E-4	0.99718 ± 6.4E-3	0.89655 ± 6.4E-4	0.99816 ± 3.9E-4	0.98411 ± 1.7E-4	0.99822 ± 7.8E-4	**0.99970 ± 7.7E-5**
	RMSE	4.19854	13.53892	0.00808 ± 6.4E-4	0.00782 ± 7.5E-4	0.04730 ± 1.8E-4	0.00626 ± 6.2E-4	0.04154 ± 7.4E-4	0.00611 ± 1.4E-4	**0.00211 ± 2.9E-4**
60min	MAE	2.81168	3.56566	0.00583 ± 2.8E-4	0.00572 ± 8.5E-4	0.04103 ± 5.2E-5	0.00513 ± 3.3E-4	0.02623 ± 2.4E-4	0.00493 ± 7.3E-4	**0.00141 ± 1.3E-4**
	NSE	0.96929	0.87378	0.99690 ± 1.4E-4	0.99698 ± 5.5E-4	0.89605 ± 4.3E-4	0.99815 ± 1.9E-4	0.98351 ± 6.0E-4	0.99820 ± 4.7E-4	**0.99966 ± 7.7E-6**
	RMSE	4.19854	13.70358	0.00823 ± 1.7E-4	0.00804 ± 7.3E-4	0.04737 ± 1.0E-4	0.00633 ± 3.8E-4	0.04160 ± 1.9E-4	0.00623 ± 7.3E-4	**0.00230 ± 1.0E-4**

The best results are in bold and underline denotes the second-best results. * Denotes our models.

of the spatiotemporal models is superior to that of the two temporal-based models GRU and LSTM. This is because these two temporal-based models do not model the spatial features of flood sequence data. For instance, at a 30-min forecast horizon, our proposed model reduces the RMSE by 73.97% and 73.83% compared to LSTM and GRU models, respectively, demonstrating the impact of spatial features on flood prediction.

2) *Comparison between the spatiotemporal models, i.e., HD-TGCN, D-TGCM, T-GCN, and STGCN, and the spatial-based model, i.e., GCN:* As illustrated in Fig. 5(b), the performance of the three spatiotemporal models is superior to that of the GCN model for all the forecasting horizons. For example, at a 15-min forecast horizon, our proposed model reduces the MAE and the RMSE by 97.81% and 96.77% compared to the GCN models, respectively. This is because the GCN model can only consider the spatial relationships between neighboring nodes in graph data and cannot model the temporal interrelationships. Based on the above analysis, the models that capture spatiotemporal features among nodes can more accurately predict flood. Moreover, the HD-TGCN model based on heterogeneous dynamic adjacency matrices has better spatiotemporal modeling ability compared to other baseline models.

As shown in Fig. 6, the performance of all the models decreases as the forecasting horizon increases. This is primarily because long-term predictions involve increased uncertainty as compared to short-term predictions, and errors accumulate with increasing forecasting horizon. Nevertheless, our proposed HD-TGCN model outperforms all other models across all the forecasting horizons. This is because the HD-TGCN model comprehensively considers various hydrological and meteorological features and uses dynamic adjacency matrices to fuse real-time changing factors, enhancing the flexibility of model. Specifically, our model outperforms the second-best model T-GCN by a considerable margin across all the forecasting horizons. This indicates that our model is not only adept at short-term predictions but also performs well in long-term predictions.



(a)

(b)

(c)

Fig. 6. Performance changes of different methods as the forecasting horizon increases.

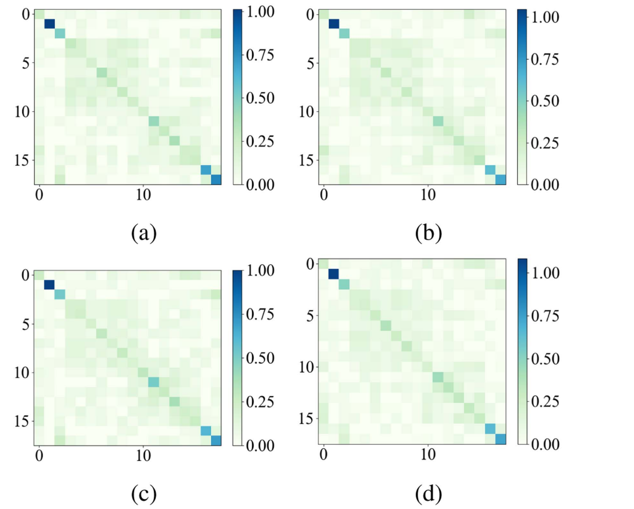


Fig. 7. (a)–(d) Visualization of dynamic adjacency matrices in four different stages.

In order to further explore the impact of the dynamic adjacency matrix, we visualize the adjacency matrix for different stages of the flood. As shown in Fig. 7, the heatmaps represent the adjacency matrices of four different stages. It can be observed that the connectivity between stations varies across different stages, which further confirms that the underlying surface continuously changes during the flood occurrence, leading to variations in flood frequency and peak flow.

The HD-TGCN is a novel network model that extends the capabilities of the traditional GCN to handle dynamic and heterogeneous graph data. To evaluate the performance of the HD-TGCN, we conducted a series of ablation experiments to analyze the impact of different components and settings of the model. In our ablation experiments, we considered four different models.

1) *Dynamic GCN (D-TGCN):* This is a GCN model extended to handle dynamic graph data by considering time-varying adjacency matrices.
2) *Heterogeneous GCN (H-TGCN):* This is a GCN model extended to handle heterogeneous graph data by considering different types of nodes and edges.
3) *HD-TGCN:* This is the proposed model that combines the advantages of the dynamic GCN and the heterogeneous GCN.

To evaluate the performance of each model, we use RMSE, NSE, and MAE as the evaluation indicator. As shown in Figs. 8 and 9, our experiments showed that the HD-TGCN outperformed all other models across all the evaluation metrics. Specifically, the HD-TGCN achieved significantly prediction accuracy compared to the dynamic GCN and the heterogeneous GCN. This suggests that the HD-TGCN effectively captures the dynamic and heterogeneous nature of graph data, leading to better predictive performance.

Furthermore, the HD-TGCN provided more interpretable results compared to other models, allowing for a better understanding of the topological structure and dynamics of the graph data. This is particularly important in real-world applications, where insights into the behavior of complex systems are crucial for making informed decisions. In conclusion, our ablation experiments demonstrate that the HD-TGCN is a superior model for handling dynamic and heterogeneous graph data compared to traditional GCN models extended for dynamic or heterogeneous settings alone. The HD-TGCN combines the advantages of both the approaches, resulting in improved predictive performance and better interpretability.

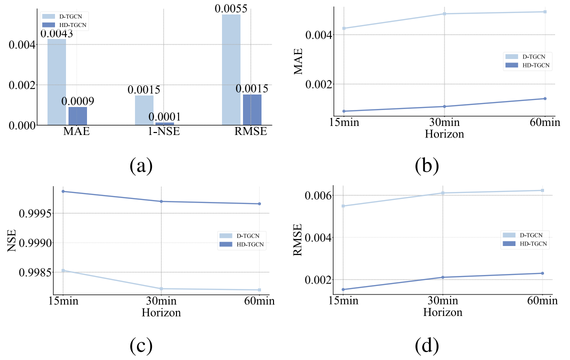


Fig. 8. (a)–(d) Component analysis in the heterogeneous graph.

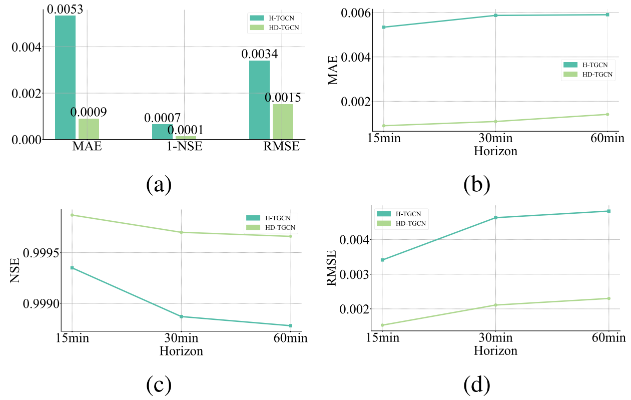


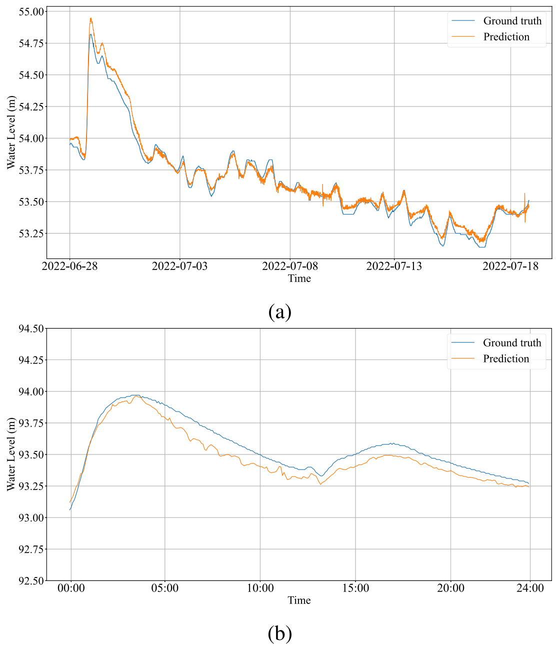
Fig. 9. (a)–(d) Component analysis in the dynamic graph.

TABLE III
IMPACT OF LEARNING RATE ON HD-TGCN PERFORMANCE

Learning rate	MAE	NSE	RMSE
0.01	0.00139	0.99945	0.00201
0.001	0.00090	0.99987	0.00153
0.0001	0.00116	0.99962	0.00174

To analyze hyperparameter sensitivity, we conducted additional experiments to analyze the impact of different hyperparameters in the HD-TGCN, such as the number of hidden units and learning rate. The learning rate is an important factor that affects the speed of model training and convergence, while the number of hidden units is directly related to the model's ability to capture spatiotemporal dependencies.

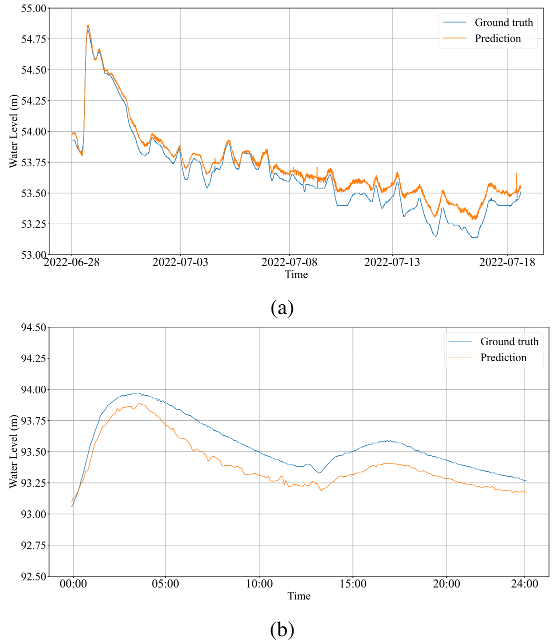
We first analyzed the impact of learning rate on model performance. We chose three different learning rates of 0.01, 0.001, and 0.0001 for our experiments. The experimental results are shown in Table III.



(a)

(b)

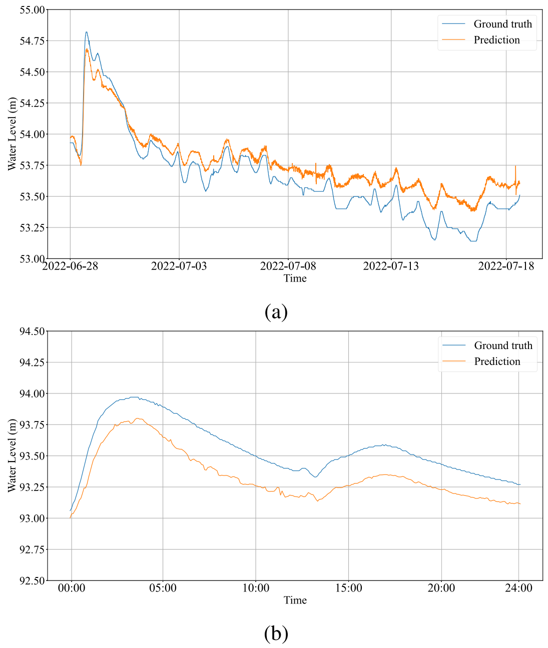
Fig. 10. (a) and (b) Visualization results for prediction horizon of 15 min.



(a)

(b)

Fig. 11. (a) and (b) Visualization results for prediction horizon of 30 min.



(a)

(b)

Fig. 12. (a) and (b) Visualization results for prediction horizon of 60 min.

TABLE IV
IMPACT OF HIDDEN UNITS ON HD-TGCN PERFORMANCE

Hidden units	MAE	NSE	RMSE
32	0.00218	0.99926	0.00251
64	0.00090	0.99987	0.00153
128	0.00155	0.99943	0.00198

Our experiments also showed that the learning rate has a significant impact on the model's convergence behavior. We observed that a lower learning rate of 0.0001 resulted in slower convergence but better generalization performance, whereas a higher learning rate of 0.01 led to faster convergence but worse generalization performance. Optimal performance was achieved with a learning rate of 0.001, which provided a good balance between convergence speed and generalization performance.

Next, we analyzed the effect of the number of hidden units on the performance of the model. We chose three different numbers of hidden units, 32, 64, and 128, for our experiments. The experimental results are shown in Table IV.

By adjusting the number of hidden units, we observed a direct correlation with the model's predictive capability. The experiment revealed that increasing the number of hidden units from 32 to 64 improved the model's RMSE, NSE, and MAE. However, further increasing the number of hidden units to 128 did not yield any significant improvement in performance, indicating a saturation effect.

These ablation experiments demonstrate that the HD-TGCN is sensitive to hyperparameter variations and that careful tuning of these parameters is essential for achieving optimal performance. Different hyperparameter settings can significantly affect the training speed, stability, and generalization performance of the model. Therefore, in practical applications, it is necessary to adjust these hyperparameters according to specific tasks and datasets to achieve optimal performance. In addition, for future research, it is possible to further optimize the hyperparameter selection strategy of the HD-TGCN to improve the performance and robustness of the model.

To better comprehend the HD-TGCN model, we visualized its prediction results for two stations in the Wuyuan dataset. Figs. 10–12 display the visualization results for the two stations for prediction horizons of 15, 30, and 60 min, respectively. The HD-TGCN model demonstrates exceptional performance in flood forecasting, irrespective of the prediction horizon. This model effectively captures the spatiotemporal characteristics of floods and accurately predicts variations in water levels. In addition, the HD-TGCN model can identify the initiation and cessation of flood events, generating predictions that closely resemble real water level. These capabilities are crucial for precise flood prediction and addressing other hydrological phenomena.

VI. Conclusion

In this article, we introduce the HD-TGCN, an effective deep-learning-based neural network designed for flood forecasting. The HD-TGCN employs innovative approaches to model flood data dynamics along with both the temporal and spatial dimensions, while factoring in the heterogeneity of flood data. Specifically, we present a D-TGCM that incorporates an MHSA mechanism to generate a dynamic adjacency matrix. This enables the model to capture dynamic spatiotemporal features of flood data by utilizing temporal graph convolution operations based on the dynamic matrix. In addition, given the complexity of flood formation arising from multiple interfering factors, our novel approach incorporates multiple parallel D-TGCMs to process heterogeneous graph data and fusion mechanisms to capture complex flooding patterns effectively. Moreover, the inclusion of remote sensing data enriches the feature representation, empowering the model to discern complex relationships that traditional methods struggle to capture. In summary, our proposed HD-TGCN model bridges the gap between spatiotemporal prediction and heterogeneous data integration by incorporating remote sensing information. Experimental results demonstrate that the HD-TGCN significantly outperforms state-of-the-art flood prediction models in terms of accuracy. Thus, our proposed framework makes a significant contribution to flood forecasting and holds enormous potential for mitigating the damage and loss of life caused by future large-scale flood events. Our future work will investigate the possibility of introducing the HD-TGCN model into the transportation system for traffic prediction and jam forecasting, with more lightweight models using edge intelligence [42], [43].

References

[1] W. Yang, Y. Zhao, D. Wang, H. Wu, A. Lin, and L. He, "Using principal components analysis and IDW interpolation to determine spatial and temporal changes of surface water quality of Xin'anjiang River in Huangshan, China," *Int. J. Environ. Res. Public Health*, vol. 17, no. 8, 2020, Art. no. 2942.

[2] M. Valipour, M. E. Banihabib, and S. M. R. Behbahani, "Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir," *J. Hydrol.*, vol. 476, pp. 433–441, 2013.

[3] S.-T. Chen and P.-S. Yu, "Pruning of support vector networks on flood forecasting," *J. Hydrol.*, vol. 347, no. 1-2, pp. 67–78, 2007.

[4] Z. Liang, Y. Li, Y. Hu, B. Li, and J. Wang, "A data-driven SVR model for long-term runoff prediction and uncertainty analysis based on the Bayesian framework," *Theor. Appl. Climatol.*, vol. 133, pp. 137–149, 2018.

[5] A. Jayawardena, D. Fernando, and M. Zhou, "Comparison of multilayer perceptron and radial basis function networks as tools for flood forecasting," *Int. Assoc. Hydrol. Sci. Publ.—Ser. Proc. Rep.*, vol. 239, pp. 173–182, 1997.

[6] A. Mosavi, P. Ozturk, and K.-W. Chau, "Flood prediction using machine learning models: Literature review," *Water*, vol. 10, no. 11, 2018, Art. no. 1536.

[7] Y. Zhou, S. Guo, C.-Y. Xu, F.-J. Chang, and J. Yin, "Improving the reliability of probabilistic multi-step-ahead flood forecasting by fusing unscented Kalman filter with recurrent neural network," *Water*, vol. 12, no. 2, 2020, Art. no. 578.

[8] X.-H. Le, H. V. Ho, G. Lee, and S. Jung, "Application of long short-term memory (LSTM) neural network for flood forecasting," *Water*, vol. 11, no. 7, 2019, Art. no. 1387.

[9] T. Song, W. Ding, J. Wu, H. Liu, H. Zhou, and J. Chu, "Flash flood forecasting based on long short-term memory networks," *Water*, vol. 12, no. 1, 2019, Art. no. 109.

[10] S. Sahraei, M. Asadzadeh, and F. Unduche, "Signature-based multi-modelling and multi-objective calibration of hydrologic models: Application in flood forecasting for Canadian prairies," *J. Hydrol.*, vol. 588, 2020, Art. no. 125095.

[11] C. Chen, J. Jiang, Y. Zhou, N. Lv, X. Liang, and S. Wan, "An edge intelligence empowered flooding process prediction using Internet of Things in smart city," *J. Parallel Distrib. Comput.*, vol. 165, pp. 66–78, 2022.

[12] R. Hu, F. Fang, C. Pain, and I. Navon, "Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method," *J. Hydrol.*, vol. 575, pp. 911–920, 2019.

[13] H. S. Munawar, A. W. Hammad, and S. T. Waller, "Remote sensing methods for flood prediction: A review," *Sensors*, vol. 22, no. 3, 2022, Art. no. 960.

[14] W. Li et al., "An effective multi-model fusion method for SAR and optical remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 5881–5892, 2023.

[15] N. Lv et al., "A hybrid-attention semantic segmentation network for remote sensing interpretation in land-use surveillance," *Int. J. Mach. Learn. Cybern.*, vol. 14, no. 2, pp. 395–406, 2023.

[16] C. Chen, J. Jiang, Z. Liao, Y. Zhou, H. Wang, and Q. Pei, "A short-term flood prediction based on spatial deep learning network: A case study for Xi County, China," *J. Hydrol.*, vol. 607, 2022, Art. no. 127535.

[17] W. Li, J. Wu, H. Chen, Y. Wang, Y. Jia, and G. Gui, "UNet combined with attention mechanism method for extracting flood submerged range," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 6588–6597, 2022.

[18] M. M. H. Khan, N. S. Muhammad, and A. El-Shafie, "Wavelet based hybrid ANN-ARIMA models for meteorological drought forecasting," *J. Hydrol.*, vol. 590, 2020, Art. no. 125380.

[19] Y. Wang, Y. Huang, M. Xiao, S. Zhou, B. Xiong, and Z. Jin, "Medium-long-term prediction of water level based on an improved spatio-temporal attention mechanism for long short-term memory networks," *J. Hydrol.*, vol. 618, 2023, Art. no. 129163.

[20] I.-F. Kao, J.-Y. Liou, M.-H. Lee, and F.-J. Chang, "Fusing stacked autoencoder and long short-term memory for regional multistep-ahead flood inundation forecasts," *J. Hydrol.*, vol. 598, 2021, Art. no. 126371.

[21] K. Khosravi et al., "Convolutional neural network approach for spatial prediction of flood hazard at national scale of Iran," *J. Hydrol.*, vol. 591, 2020, Art. no. 125552.

[22] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*.

[23] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv: 1707.01926*.

[24] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 914–921.

[25] W. Chen, L. Chen, Y. Xie, W. Cao, Y. Gao, and X. Feng, "Multi-range attentive bicomponent graph convolutional network for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 3529–3536.

[26] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. 32nd AAAI Conf. Artif. Intell./13th Innov. Appl. Artif. Intell. Conf./8th AAAI Symp. Educ. Adv. Artif. Intell.*, 2018, pp. 7444–7452.

[27] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proc. 25th Int. Conf. Neural Inf. Process.*, 2018, pp. 362–373.

[28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[29] W. Li, H. Chen, Q. Liu, H. Liu, Y. Wang, and G. Gui, "Attention mechanism and depthwise separable convolution aided 3DCNN for hyperspectral remote sensing image classification," *Remote Sens.*, vol. 14, no. 9, 2022, Art. no. 2215.

[30] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, 2011.

[31] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[32] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[33] C. C. Robusto, "The cosine-haversine formula," *Amer. Math. Monthly*, vol. 64, no. 1, pp. 38–40, 1957.

[34] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[35] K. Park, Y. Jung, Y. Seong, and S. Lee, "Development of deep learning models to improve the accuracy of water levels time series prediction through multivariate hydrological data," *Water*, vol. 14, no. 3, 2022, Art. no. 469.

[36] M. Anuruddhika, L. Premarathna, K. Perera, W. Hansameenu, and V. Weerasinghe, "Vector autoregressive (VAR) model for forecasting water level in Attanagalu Oya," in *Proc. Int. Conf. Appl. Pure Sci.*, 2020, p. 54.

[37] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[38] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5-6, pp. 602–610, 2005.

[39] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[40] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[42] C. Chen, W. Chenyu, C. Li, X. Ming, and P. Qingqi, "A V2V emergent message dissemination scheme for 6G-oriented vehicular networks," *Chin. J. Electron.*, vol. 32, no. 6, pp. 1179–1191, 2023.

[43] H. Li, C. Chen, H. Shan, P. Li, Y. C. Chang, and H. Song, "Deep deterministic policy gradient-based algorithm for computation offloading in IoV," *IEEE Trans. Intell. Transp. Syst.*, early access, doi: 10.1109/TITS.2023.3325267.

Jiange Jiang (Student Member, IEEE) received the B.Eng. degree in electronic and information engineering from Nanchang Hangkong University, Nanchang, China, in 2019. She is currently working toward the Ph.D. degree in information and communication engineering with Xidian University, Xi'an, China.

Her research interests include edge computing, spatial–temporal series prediction, and deep learning.

Chen Chen (Senior Member, IEEE) received the B.Eng., M.Sc., and Ph.D. degrees in telecommunication from Xidian University, Xi'an, China, in 2000, 2006, and 2008, respectively.

He is currently a Professor with the School of Telecommunications Engineering, Xidian University, where he is also the Director of the Xi'an Key Laboratory of Mobile Edge Computing and Security and the Intelligent Transportation Research Laboratory. He was a visiting Professor with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, USA, and with the Department of Computer Science, University of California, Irvine, CA, USA. He is a General Chair, Program Committee Chair, Workshop Chair, or Technical Program Committee Member of a number of conferences. He has authored or coauthored two books and more than 130 scientific papers in international journals and conference proceedings. He has contributed to the development of five copyrighted software systems and invented more than 100 patents.

Dr. Chen is a Distinguished Member of the China Computer Federation. He is a Senior Member of the China Institute of Communications.

Yang Zhou received the B.Eng. degree in communication engineering from Xidian University, Xi'an, China, in 2000.

Since 2000, he has been with the Ministry of Water Resources of China, Beijing, China. His research interests include wireless communication and computer engineering.

Stefano Berretti (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in electronic engineering and informatics and telecommunication engineering from University of Florence, Florence, Italy, in 1997 and 2001, respectively.

He is an Associate Professor with the Media Integration and Communication Center, Florence, Italy, and with the Department of Information Engineering, University of Florence, Florence. He was a Visiting Professor with the University of Lille, Lille, France, and the University of Alberta, Edmonton, AB, Canada. He provided several contributions on 3-D object retrieval and partitioning, face biometrics, facial expression and emotion recognition, human action recognition, and 3-D surface description for relief pattern classification. He was a Chair of some workshops and was the Program Chair of several workshops and conferences. His research interests include computer vision, pattern recognition, and multimedia.

Dr. Berretti was the Information Director of *ACM Transactions on Multimedia Computing, Communications, and Applications* (ACM TOMM). He is an Associate Editor for *IET Computer Vision*, ACM TOMM, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (IEEE TCSVT), and *Sensors*. He is also the Associate Editor in Chief for IEEE TCSVT. He is a Member of the Italian Association for Computer Vision, Pattern Recognition and Machine Learning, the Computer Vision Foundation, and the Association for Computing Machinery.

Lei Liu (Member, IEEE) received the B.Eng. degree from Zhengzhou University, Zhengzhou, China, in 2010, and the M.Sc. and Ph.D. degrees from Xidian University, Xi'an, China, in 2013 and 2019, respectively, all in communication engineering.

He is currently a Lecturer with the Department of Electrical Engineering and Computer Science, Xidian University. From 2013 to 2015, he was with a technology company. From 2018 to 2019, he was a visiting Ph.D. student with the University of Oslo, Oslo, Norway. His research interests include vehicular ad hoc networks, intelligent transportation, mobile edge computing, and the Internet of Things.

Qingqi Pei (Senior Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees in computer science and cryptography from Xidian University, Xi'an, China, in 1998, 2005, and 2008, respectively.

He is currently a Professor and Member with the State Key Laboratory of Integrated Services Networks, Xidian University, where he is also the Director of the Shaanxi Key Laboratory of Blockchain and Secure Computing. His research interests include digital contents protection and wireless networks and security.

Dr. Pei is a Professional Member of the Association for Computing Machinery. He is a Senior Member of the Chinese Institute of Electronics and the China Computer Federation.

Jianming Zhou received the bachelor's degree in mathematics and management from the University of Electronic Science and Technology of China, Chengdu, China, in 2005, and the Master of Engineering degree in big data technology and engineering from Tongji University, Shanghai, China, in 2019.

He is currently a Researcher of China Unicom Shenzhen Branch, Shenzhen, China. His research interests include big data, cloud computing, Internet of Things, industrial Internet, 5G, and communication technology.

Shaohua Wan (Senior Member, IEEE) received the joint Ph.D. degree in computer software and theory from the School of Computer, Wuhan University, Wuhan, China, and the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA, in 2010.

In 2015, he was a Postdoctoral Researcher with the State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan. From 2016 to 2017, he was a visiting Professor with the Department of Electrical and Computer Engineering, Technical University of Munich, Munich, Germany. He is currently an Associate Professor with the School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan. He is an author of more than 90 peer-reviewed research papers and books. His main research interests include deep learning for Internet of Things and edge computing.