GC-SALM: Multi-Task Runoff Prediction Using Spatial-Temporal Attention Graph Convolution Networks

Jin Lu[*†], Zaipeng Xie[✉*†], Jiayu Chen[†], Maohua Li[†], Chenghong Xu[†], and Hongli Cao[‡]
[*]Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing, China
[†]College of Computer and Information, Hohai University, Nanjing, China
[‡]Key Laboratory of Underwater Acoustic Signal Processing of Ministry of Education, Southeast University, Nanjing, China
Email: {jinlu, zaipengxie, jiayuchen, maohuali, chxu}@hhu.edu.cn, honglicao@seu.edu.cn

***Abstract*—Runoff prediction is essential for flood forecasting, irrigation planning, and sustainable water resource management. However, accurate predictions can be challenging due to the involvement of multiple variables. This paper presents a novel Graph Convolution-based Spatial-temporal Attention LSTM Multi-Task learning (GC-SALM) model for accurate runoff predictions. Our approach combines a multilayer neural network and an attention mechanism for enhanced generalization performance. The GC-SALM model employs spatial attention and graph convolutional networks to discern local and global spatial patterns, while temporal attention and LSTM are utilized to capture temporal characteristics within extended sequences. Experimental results reveal that the proposed model outperforms six state-of-the-art methods in runoff prediction and flow calibration, emphasizing its potential for real-world hydrological applications.**

***Index Terms*—Runoff prediction, Multi-Task Learning, Spatial and temporal modeling, Graph convolutional networks**

I. Introduction

Runoff prediction is a critical task [1] in hydrology and water resources management, and it plays a vital role in understanding the effects of climate change and urbanization on water resources. Accurate runoff predictions [2] can help estimate the amount of water available for human uses in a given basin and facilitate making decisions about water allocation, flood management, and the design and operation of water resource infrastructures in a sustainable way. Neural network-based models [2], [3] have gained popularity for predicting runoff using historical data. These models can be trained on historical data of precipitation, temperature, and other meteorological variables, as well as information about the land surface, such as soil type and vegetation cover, to make predictions about future runoff. For example, convolutional neural networks (CNNs) [4] [5] and recurrent neural networks (RNNs) [6] can effectively predict spatio-temporal patterns in complex systems. However, hydrological data are spatio-temporal series, dependent on the volatility and uncertainty of meteorological and hydrological characteristics in the time dimension and the correlation in space dimension. They may have limitations in capturing temporal and spatial dependencies, with RNNs lacking the ability to capture long-term temporal dependencies and CNNs being unable to capture non-local spatial dependencies.

Recently, researchers have been using graph convolutional neural networks (GCNs) to model the spatial and temporal dependencies in time series data in order to make more accurate predictions by directly acting on a graph and aggregating its structural information. For example, traffic flow prediction can be modeled as a graph problem [7], and GCNs can be used to capture the spatial correlation among roads. In the field of hydrology, GCNs have shown potential in predicting runoff. Sit et al. [8] proposed a model using Graph Convolutional GRUs to predict the next 36 hours of stream flow at a specific sensor location using information from the upstream river network. Feng et al. [9] developed a new method that combines GCN and LSTM to extract spatial and temporal information from raw flood data for hydrological prediction. However, conventional GCNs encounter challenges in capturing global correlations when predicting runoff, resulting in disconnection between hydrological stations located at distant locations.

To overcome this issue, multi-task learning (MTL) [10] has been introduced as a technique that simultaneously models multiple prediction tasks by utilizing the correlation between them. The information shared between related tasks can improve overall predictive accuracy by training models jointly. Sadler et al. [10] explored the benefits of modeling two interdependent variables, daily average streamflow and daily average stream water temperature, using multi-task deep learning. Zhang et al. [11] proposed a Multi-task two-stream spatiotemporal convolutional neural network for forecasting convective storms, using radar and satellite data to learn task correlations automatically. Their results showed that prediction accuracy may be improved by jointly learning through shared representations. Li et al. [12] proposed a multi-task GNN to synchronously predict spatial-temporal traffic data at different regions and their transitions by constructing multi-task graph representations to extract multiple spatial correlations. However, due to the complexity of the models

The corresponding author is Zaipeng Xie. This work is supported by the Belt and Road Special Foundation of the State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering under Grant 2021490811.

used in multi-task learning, designing a multi-task learning model requires careful consideration of the relationships and dependencies between tasks and the choice of appropriate loss functions to balance the learning across tasks.

This work proposes a novel Graph Convolution-based Spatial-temporal Attention LSTM Multi-Task learning (GC-SALM) model that combines spatial attention and GCN to identify local and global spatial patterns. Temporal attention and LSTM are employed to capture temporal characteristics of extended sequences. Our GC-SALM method, which utilizes a multi-layer fully connected neural network and attention mechanism, can improve prediction accuracy by identifying similarities among related tasks in our proposed runoff prediction models. The main contributions of this work are:

- We propose a novel graph neural network with spatial attention to better aggregate hydrological information from neighboring sites. LSTM and temporal attention modules are used to extract temporal dependencies dynamically.
- We propose a novel multi-task learning model that employs a multilayer fully connected neural network and an attention mechanism to find similarities between related tasks and improve the prediction accuracy of spatial and temporal input runoff data.
- We empirically show that the proposed model can effectively perform the runoff prediction with high accuracy when compared with several state-of-the-art models using the CAMELS datasets.

The remainder of the paper is organized as follows. Section II presents the general framework and details of the GC-SALM model. The experimental evaluation of the GC-SALM is presented in Section III, and Section IV concludes our work.

II. Methodology

A. Hydrology Spatial Graph

We consider each hydrological station as a vertex and build a graph to establish the relationship among stations. The hydrology Graph can be represented by $G = (V, E, A)$, where $V = \{v_1, v_2, \ldots v_N\}$ denotes hydrological stations that are collecting the hydrological data; E denotes the set of edges between stations in the Hydrology Graph; A is an adjacency matrix, and $A[i, j]$ describes the edge weight between station v_i and v_j. Therefore, the hydrological runoff prediction [13] can be described as a graph problem.

B. Problem Statement

At the moment t, the input feature matrix of the network can be represented as $X_t \in R^{N\times C}$, where c represents the number of features. Our goal is to learn a mapping function f to predict the hydrological runoff of future T moments given the hydrological features and hydrological graph G of historical n moment:

$$Y^{(t+1):(t+T)} = f\left(X^{(t-n):t}, G\right) \tag{1}$$

where N is the length of the historical time series and Y is the time series needed to be predicted.

C. Proposed GC-SALM Method

We propose the Graph Convolution-based Spatial-Temporal Attention LSTM Multi-Task Learning (GC-SALM) model, designed for runoff prediction. The GC-SALM model encompasses a spatial attention graph convolutional module (SA-GCN), a temporal attention LSTM module (TA-LSTM), and a multi-task learning module, as illustrated in Fig. 1. Our graph convolutional module aims to capture spatial relationships within hydrological data, while the attention mechanism assists in identifying crucial factors that influence runoff. This synergy results in enhanced performance and generalization capabilities. Moreover, the GC-SALM model can be trained to execute multiple tasks, enabling it to utilize information from related variables for more accurate predictions.

To effectively model the spatial correlation of runoff, it is essential to establish a topological structure for hydrological stations. Given that river flow states are dynamic and evolve over time, representing these fluctuations through graph nodes while maintaining a fixed graph structure is crucial. Developing a hydrological topology necessitates a thorough evaluation of both hydrological and statistical knowledge. From a hydrological standpoint, closely situated stations are more likely to influence each other's hydrological conditions. Statistically, the correlation between stations can be quantified using historical data acquired from various hydrological stations. We employ Pearson's correlation coefficient to determine the inter-station correlation. Our hydrological spatial graph generation module computes these correlations based on flow rates per unit of time for the previous three years. Subsequently, the adjacency matrix A is constructed as shown in

$$A = \begin{bmatrix} 1 & \cdots & p_{1,j} & \cdots & p_{1,N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{i,1} & \cdots & p_{i,j} & \cdots & p_{i,N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{N,1} & \cdots & p_{N,j} & \cdots & 1 \end{bmatrix} \tag{2}$$

where $p_{i,j}$ is the Pearson correlation result between station i and station j. Utilizing matrix A, we can obtain the adjacent edges and nodes. This allows us to generate the hydrological topology graph, which provides a visual representation of the connections between nodes in the network.

D. Spatial Attention Graph Convolutional Network Module

We know that the influences from other stations can vary, depending on the specific runoff conditions. For instance, the impact of one station on others might be more pronounced when the basin hosting the station undergoes heavy precipitation. Therefore, we employ the spatial attention mechanism [13] to extract spatial correlations, $\tilde{S}$, of flow series between stations, as given by

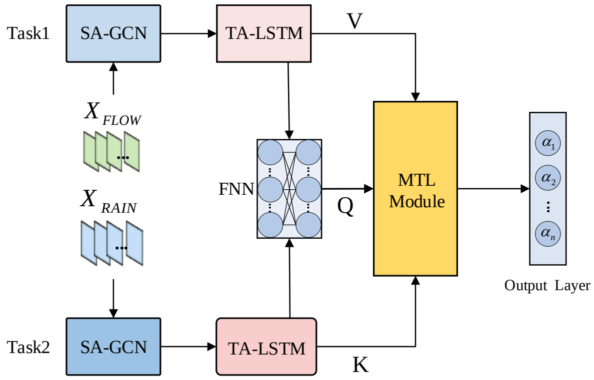


Fig. 1: Overview of the proposed GC-SALM architecture, where FNN stands for Feedforward Neural Network.

$$\tilde{S} = \sigma(W_s \cdot \sigma((XU_1)U_2(U_3X)^T + b_s)) \tag{3}$$

where $X \in \mathbb{R}^{N\times C\times T}$ is the input of the spatial attention block. N denotes the number of nodes in the hydrological network, C denotes the number of input data channels, and T denotes the length of time slices. The trainable parameters include $U_1 \in \mathbb{R}^T$, $U_2 \in \mathbb{R}^{C\times T}$, $U_3 \in \mathbb{R}^C$, W_s, and $b_s \in \mathbb{R}^{N\times N}$, while $\sigma(\cdot)$ denotes the sigmoid activation function. The spatial attention matrix, $\tilde{S} \in \mathbb{R}^{N\times N}$, is computed based on the current layer input. When performing graph convolution on the hydrographic graph, $\tilde{S}$ is calculated alongside the adjacency matrix A to adjust the graph nodes' weights. The value of $S_{i,j}$ in $\tilde{S}$ indicates the correlation degree between node i and j.

The hydrological data exhibit non-linear and complex characteristics, making it suitable for representation as graph-structured data. All hydrological station points within the river network can be treated as nodes, with each node's features considered as signals on the graph. To thoroughly extract the topological features of the river network, the input, adjusted by the attention mechanism, is fed into the spatio-temporal convolution module for a graph convolution operation. Graph convolution is a critical process for extracting a node's features based on its structural information. The spatio-temporal convolution module proposed in this study encompasses a graph convolution in the spatial dimension, which captures spatial dependencies from the neighboring nodes.

In classical convolution operations, the Graph Convolutional Network (GCN) model can be constructed by stacking multiple convolutional layers given by

$$H^{(l+1)} = \sigma\left(LH^{(l)}W^{(l)}\right) \tag{4}$$

where L is the normalized form of the Laplacian matrix, $H^{(l)}$ is the output of l layer, $W^{(l)}$ contains the parameters of that layer. Nonetheless, the convolution kernel may not be highly effective in processing complex graph models. When the graph scale is large, directly performing eigenvalue decomposition on the Laplacian matrix becomes computationally expensive. Computational complexity can be significantly reduced by utilizing a graph convolution kernel approximated with Chebyshev polynomials [14]. The equation for graph convolution using Chebyshev polynomials can be expressed as:

$$g_{\theta *G}x = g_\theta(\tilde{L})x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x \tag{5}$$

where $*_G$ denotes a graph convolution operation, θ_k are the filter coefficients, $T_k(\tilde{L})$ are the Chebyshev polynomials of degree k evaluated at the normalized graph Laplacian $\tilde{L}$, which is a rescaled version of the Laplacian matrix L, x is the input signal. Using the approximate expansion of the Chebyshev polynomial to solve this formulation involves extracting information about the neighboring nodes centered around each node in the graph via the convolution kernel. The embedded spatial attention matrix facilitates the enhancement of graph convolution for the extraction and enhancement of spatiotemporal dynamic features of the river network. Therefore, we associate $T_k(\tilde{L})$ with the spatial attention matrix $\tilde{S}_m \in \mathbb{R}^{N\times N}$ to obtain the combined polynomial $T_k(\tilde{L}) \odot \tilde{S}_m$, where $\odot$ is the Hadamard product. The input is $X_g = (x_1, x_2, \cdots, x_T) \in \mathbb{R}^{N\times C_g\times T}$ and let $t \in [1, T]$, where C_g denotes the number of feature channels of each node. Thus, Equation 5 can be rewritten as the following equation at time t:

$$g_{\theta *G}x_t = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L} \odot \tilde{S}_m)x_t \tag{6}$$

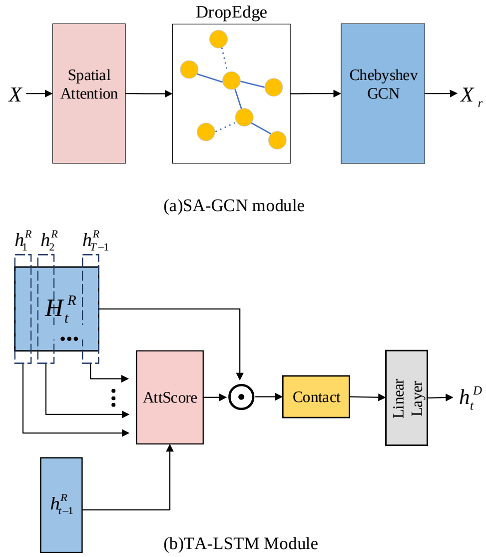
where $\tilde{S}_m$ is the matrix of spatial attention computed by X_g. The spatial attention matrix can be adjusted dynamically using the convolution formula provided. This formula enables the correlation between the nodes to be modified in a flexible manner. The output of the spatial attention graph convolutional block, denoted by X_r, employs the ReLU activation function:

$$X_r = ReLU(g_{\theta *G}x_t) \tag{7}$$

In summary, the spatial attention and convolution modules comprise the spatial attention graph convolutional block, which is followed by a fully connected layer to maintain output consistency. The block output is then passed to the temporal attention LSTM module for further processing. This methodology can effectively capture temporal information and improve the performance of the model.

E. Temporal Attention LSTM Module

The spatial attention graph convolutional module places a high priority on capturing the topological structure of hydrological networks. To process the sequence information and capture time dependencies, we utilize an LSTM and generate a hidden representation for each time step. However, given the known issue of long-term information loss [6] in traditional LSTM models, this poses a significant challenge in the context of modeling flooding processes, which exhibit



(a)SA-GCN module

(b)TA-LSTM Module

Fig. 2: Illustration of (a) the SA-GCN and (b) the TA-LSTM modules used in the training process. To enhance the robustness of the model, the DropEdge mechanism is employed, which randomly removes edges of a fixed proportion p in the original hydrology graph during each training iteration.

seasonality and periodicity and require careful consideration of long-term dependencies. In response to this challenge, we introduce temporal attention as a means of improving the model's capacity to assign weight to critical timing features and enhance prediction accuracy. Specifically, we input $X_r = (x_1^r, x_2^r, \cdots, x_T^r)$ to the temporal attention LSTM module and represent the hidden layer output in the LSTM network as a stacking matrix, $H_t^R = (h_1^R, h_2^R, \cdots, h_{T-1}^R)$. To calculate the attention weights α_t at each time step, we employ a scoring function in our attention mechanism, denoted by

$$\alpha_t = AttenScore\left(H_t^R, h_{t-1}^R\right) \tag{8}$$

where H_t^R is the stacking matrix of hidden layer output in the LSTM network, and h_{t-1}^R is the last hidden layer output. A scoring function is used to determine the similarity between these two representations, which may be in the form of additive functions, dot product functions, cosine functions, or other similarity measures commonly used in the attention mechanism. In this study, we employ dot product functions. The final output of the temporal attention module is a linear combination of the weighted vector c_t and the last hidden layer output h_{t-1}^R, which is obtained through a linear mapping operation given by

$$\begin{cases} h_t^R = LSTM(h_{t-1}^R) \\ c_t = H_t \cdot \alpha_t \\ h_t^D = W[c_t; h_{t-1}^R] + b \end{cases} \tag{9}$$

To obtain the complete output of the output gate within the range, we concatenate the entire time slice to form the output matrix, denoted by $H_l = (h_1^D, h_2^D, \cdots, h_T^D) \in \mathbb{R}^{N \times C^h \times T}$. The final output of the module is then obtained through the application of the ReLU activation function. This methodology enables the model to effectively capture and process temporal information, enhancing its predictive capacity and overall performance in a variety of applications.

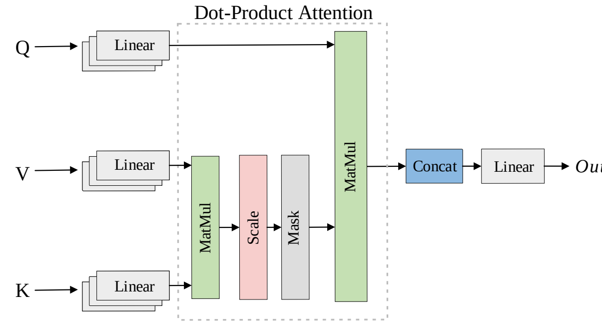


Fig. 3: Multi-Task Learning Module

F. Multi-Task Learning Module

Hydrological modeling has traditionally focused on developing models to predict a single variable. This study proposes a novel approach that utilizes multi-task learning (MTL) to learn two related hydrologic variables simultaneously. Our approach acknowledges the interconnected nature of hydrological variables to improve prediction accuracy. Specifically, we focus on streamflow and rainfall as the two variables of interest. One of the key contributions of our work is incorporating attention mechanisms into the MTL framework. The attention mechanisms [15] allows the model to learn task-specific inputs by capturing commonalities between related variables, as defined by

$$Att(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_{att}}}\right) \cdot V \tag{10}$$

where the scaling of attention mechanisms on values V is based on the relationships between keys K and queries Q. Here, V represents the output of a specific variable, such as flow, while K represents the output of an interrelated variable, such as rainfall. Similarly, Q represents the output of the task-shared module, which combines the outputs of a single variable and passes them into the FNN. Equation (10) determines the size of the attention weights based on the task and shared representations. These weights enable the model to focus only on the part of the task-specific input relevant to runoff prediction.

III. Experiment

A. Experimental Setup

1) Datasets Description: We utilize the CAMELS dataset [16], which provides catchment attributes and meteorological data for 671 catchments across the contiguous United States and includes precipitation and streamflow observations. Our experiments use Catchment units 01 (New England) and 04

TABLE I: PERFORMANCE OF DIFFERENT MODELS AT 01 BASIN AND 04 BASIN

T	Metric	01							04						
		HA	ARIMA	T-GCN	ST-GCN	H-GCN	LSTM	GC-SALM	HA	ARIMA	T-GCN	ST-GCN	H-GCN	LSTM	GC-SALM
3	MAE	285.546	206.807	183.601	190.251	220.329	173.921	**171.174**	148.155	111.101	93.123	104.321	117.265	97.361	**88.394**
	RMSE	348.081	384.313	336.448	352.454	372.118	343.355	**328.213**	182.54	342.518	158.908	168.683	187.245	165.211	**146.567**
	R^2	0.558	0.905	0.98	0.979	0.928	0.981	**0.982**	0.81	0.056	0.95	0.914	0.693	0.942	**0.954**
6	MAE	285.546	206.807	224.612	217.196	285.744	221.366	**206.425**	148.155	111.101	116.481	119.594	177.552	121.358	**108.792**
	RMSE	348.081	384.313	351.447	355.315	413.654	358.444	**343.251**	182.54	342.518	188.112	195.109	247.349	189.839	**179.768**
	R^2	0.558	0.656	0.749	0.859	0.317	0.862	**0.885**	0.81	0.056	0.922	0.894	0.539	0.924	**0.931**

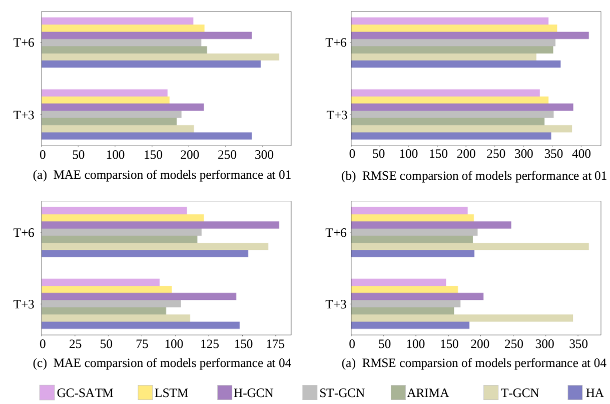
(Great Lakes), containing 26 and 23 hydrological stations, respectively. CAMELS is managed by the National Center for Atmospheric Research (NCAR) and has a relatively low anthropogenic influence. In our experiments, we partitioned the data into training (79%), validation (1%), and testing sets (20%). To ensure consistency in the orders of magnitude between different features and preserve the original trend of the time series data, we standardized the data using the Z-score method.

2) Experimental Settings: The experiments were conducted on a Linux server with an Intel(R) Xeon(R) CPU E5-2640 v4, 32GB RAM, and an Nvidia GeForce GTX 2080Ti. We implemented the GC-SALM algorithm using the PyTorch framework and evaluated its performance using a 32-day historical time window to forecast hydrology conditions for the next 3 and 6 days. We set K=3 and p=0.6 based on experimental performance and practical considerations. In the training phase, we used a learning rate of 0.0001 and a batch size of 64. To evaluate the performance of the models, we used three commonly used metrics: MAE (mean absolute error), RMSE (root mean square error), and R^2 (coefficient of determination). A smaller value for both MAE and RMSE indicates better performance, while a higher R^2 value typically indicates a better fitting performance of the model. The range of R^2 is between 0 and 1.

$$\begin{cases} MAE = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}|y_i^{test} - y_i^{pre}| \\ RMSE = \sqrt{\dfrac{\sum_{i=1}^{n}(\overline{y}_i^{pre} - y_i^{test})^2}{n}} \\ R^2 = 1 - \dfrac{\sum_{i=1}^{n}(y_i^{test} - y_i^{pre})^2}{\sum_{i=1}^{n}(y_i^{test} - \overline{y}_i^{test})^2} \end{cases} \tag{11}$$

3) Baseline Methods: We compare GC-SALM with six state-of-the-art methods.

- **HA**: this method predicts future periods by taking the weighted average of previous periods. We utilize the average value of the last 32 time slices to make predictions.
- **ARIMA** [17]: This model combines moving average and autoregressive techniques to predict future values in the hydrological field, making it a classical time series forecasting model.
- **LSTM** [18]: Long Short-Term Memory network, a special RNN model.
- **T-GCN** [19]: This model combines the GCN and GRU to perform time-series forecasting, making it a hybrid approach.
- **ST-GCN** [20]: Spatio-temporal Graph Convolutional Network. This method employs both graph convolution and 2D convolutional networks to capture spatial-temporal correlations within graph data.
- **H-GCN** [21]:Hierarchical Graph Convolution Networks. This method uses a graph differentiable pooling module that enables the learning of representation capabilities between network layers and facilitates the combination of various existing end-to-end structures.



(a) MAE comparsion of models performance at 01

(b) RMSE comparsion of models performance at 01

(c) MAE comparsion of models performance at 04

(a) RMSE comparsion of models performance at 04

Fig. 4: Visualization of evaluation indicators

B. Experimental Results

1) Performance Analysis: Table I provides a comparison between GC-SALM and six state-of-the-art models in terms of their prediction accuracy on the CAMELS datasets for 3-day and 6-day ahead prediction. We adjusted the parameters of the models to achieve the best results. As expected, the prediction errors generally increase with longer prediction intervals, indicating that making accurate predictions becomes more challenging. However, our GC-SALM model outperforms the baseline models in both basins, achieving lower average values of RMSE, MAE, and R^2. While traditional statistical methods such as ARIMA and HA have poor generalization ability, GC-SALM shows the best performance in short-time sequence prediction at T+3. For instance, in the 01 basin, we calculated RMSE, MAE, and R^2 values of 328.213, 171.174, and 0.982, respectively. Similarly, in the 04 basin, we calculated RMSE, MAE, and R^2 values of 146.567, 88.394, and 0.954, respectively. Moreover, our approach outperforms other methods such as LSTM, T-GCN, and ST-GCN in longer time series prediction at T+6, highlighting the importance of assigning more weights to important timing features and utilizing multi-task learning for flow prediction.

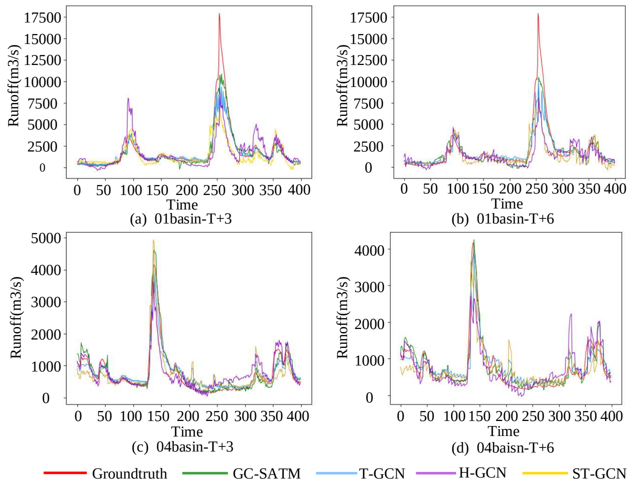


Fig. 5: A comparison between the ground truth flow rates and the predicted runoff flow rates at the 01-basin and 04-basin from the test dataset for a specific flood event. The time axis covers the period from October 2010 to November 2011, with each tick representing a 50-day interval.

2) Flow Groudtruth Comparison: To provide a more intuitive illustration of the models' performance, we present in Fig. 5 the predicted river flow in the 01 River Basin and 04 River Basin. The figure shows that the predicted flow closely aligns with the ground truth at the time of T. As the prediction time increases to T+3 and T+6, the degree of fitting between the prediction and the ground truth gradually decreases. Nevertheless, our model outperforms T-GCN, H-GCN, and ST-GCN in terms of the degree of fitting at both T+3 and T+6. Furthermore, our model is more accurate in predicting sharp increases in the runoff flow, which allows us to forecast natural disasters such as floods more promptly.

IV. Conclusion

In this paper, we proposed a new approach to improving the accuracy of runoff prediction in hydrology and water resources management. Our Graph Convolution-based Spatial-temporal Attention LSTM Multi-Task learning (GC-SALM) model combines spatial and temporal attention mechanisms with graph convolutional networks and multi-task learning to better capture the complex spatio-temporal dependencies in hydrological data. Experimental results demonstrate that our proposed GC-SALM model outperforms several state-of-the-art models, showing its potential as an effective method for predicting runoff. The proposed approach has the potential to help in making better decisions related to water allocation, flood management, and the design and operation of water resource infrastructures in a sustainable way.

References

[1] G. Blöschl, M. Sivapalan, T. Wagener, H. Savenije, and A. Viglione, *Runoff prediction in ungauged basins: synthesis across processes, places and scales*. Cambridge University Press, 2013.

[2] A. Mosavi, P. Ozturk *et al.*, "Flood prediction using machine learning models: Literature review," *Water*, vol. 10, no. 11, p. 1536, 2018.

[3] C. M. Song, "Data construction methodology for convolution neural network based daily runoff prediction and assessment of its applicability," *Journal of Hydrology*, vol. 605, p. 127324, 2022.

[4] Y. Xie, W. Sun, M. Ren, S. Chen, Z. Huang, and X. Pan, "Stacking ensemble learning models for daily runoff prediction using 1D and 2D cnn," *Expert Systems with Applications*, p. 119469, 2022.

[5] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 5668–5675.

[6] S. Gao, Y. Huang, S. Zhang, J. Han, G. Wang, M. Zhang, and Q. Lin, "Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation," *Journal of Hydrology*, vol. 589, p. 125188, 2020.

[7] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proceedings of The Web Conference 2020*, 2020, pp. 1082–1092.

[8] M. Sit, B. Demiray, and I. Demir, "Short-term hourly streamflow prediction with graph convolutional gru networks," *arXiv preprint arXiv:2107.07039*, 2021.

[9] J. Feng, Z. Wang, Y. Wu, and Y. Xi, "Spatial and temporal aware graph convolutional network for flood forecasting," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[10] J. M. Sadler, A. P. Appling *et al.*, "Multi-task deep learning of daily streamflow and water temperature," *Water Resources Research*, vol. 58, no. 4, p. e2021WR030138, 2022.

[11] W. Zhang, H. Liu, P. Li, and L. Han, "A Multi-task two-stream spatiotemporal convolutional neural network for convective storm nowcasting," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 3953–3960.

[12] H. Li, D. Jin, X. Li, J. Huang, and J. Yoo, "Multi-task synchronous graph neural networks for traffic spatial-temporal prediction," in *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, 2021, pp. 137–140.

[13] X. Feng, J. Guo, B. Qin, T. Liu, and Y. Liu, "Effective deep memory networks for distant supervised relation extraction," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 4002–4008.

[14] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 29–38.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[16] N. Addor, A. J. Newman *et al.*, "The CAMELS data set: catchment attributes and meteorology for large-sample studies," *Hydrology and Earth System Sciences*, vol. 21, no. 10, pp. 5293–5313, 2017.

[17] O. A. Fashae, A. O. Olusola, I. Ndubuisi, and C. G. Udomboso, "Comparing ANN and ARIMA model in predicting the discharge of river opeki from 2010 to 2020," *River research and applications*, vol. 35, no. 2, pp. 169–177, 2019.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

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

[21] Z. Zhu, X. Fan *et al.*, "HGCN: A heterogeneous graph convolutional network-based deep learning model toward collective classification," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 1161–1171.