Spatio-temporal network traffic prediction model based on graph attention and adaptive graph convolution

Wenbo Liu[1], Zhijian Qu[1*]
[1](School of Computer Science and Technology, Shandong University of Technology, China)

ABSTRACT: *Network traffic prediction forms the cornerstone of intelligent network management, playing an indispensable role in optimizing network resource allocation, ensuring service quality, reducing latency, enhancing user satisfaction, and preventing network congestion. To address the limitations of existing spatiotemporal prediction models in exploring the physical or logical distance relationships between nodes in network topologies, a Spatio-temporal Graph Attention Convolutional Recurrent Neural Network (STGACRN) model based on graph attention and adaptive graph convolution is proposed. This model combines the original network topology with adaptively generated graphs, and by introducing graph attention and graph convolution operations, it can effectively capture the direct interactions between network nodes as well as deeply mine the potential correlations among them. Moreover, by adding a residual structure to the GRU memory units, it avoids the problems of gradient explosion or vanishing caused by the depth of the model network. The experimental results show that the proposed model has significant effects on the accuracy of network traffic prediction.*

KEYWORDS *-Network traffic prediction, Spatial-temporal correlation, Long-term prediction*

I. INTRODUCTION

In today's digital era, the internet has permeated daily life and has become an indispensable part[1]. People rely on the internet for work, learning, entertainment, and social interaction, whether it's cloud-based office work, online education, video entertainment, or social media interaction, the internet is ubiquitous. However, this digital lifestyle has led to an explosive growth in network traffic. The constant emergence of new applications, services, and devices has placed unprecedented pressure on network operators and enterprises. Issues such as network congestion, service unavailability, and increased latency directly affect user experience, business continuity, and data security. Against this backdrop, the importance of accurately predicting network traffic has become increasingly significant.

Researchers both domestically and internationally have conducted extensive and in-depth studies in the field of network traffic prediction, which can be primarily categorized into two main types: traditional methods and deep learning-based methods. In the field of time series analysis, traditional statistical methods predict future network traffic by establishing data segmentation statistical models, primarily including Autoregressive (AR)[2], Moving Average (MA), Autoregressive Moving Average (ARMA)[3], and Autoregressive Integrated Moving Average (ARIMA)[4] models. These models predict future traffic based on the temporal correlation of historical data, each with its unique advantages. For instance, the ARIMA model is widely popular for its effective prediction capability for non-seasonal data. However, in the face of traffic data with significant seasonal variations, such as mobile network traffic, the Seasonal ARIMA[5] model exhibits better adaptability. Subsequently, researchers have proposed various traditional machine learning methods, such as Multilayer Perceptrons (MLP)[6], Support Vector Regression (SVR)[7], K-Nearest Neighbors (KNN), and Support Vector Machines (SVM)[8], to address the issue of network traffic prediction. The operations of the above two types of algorithms are similar, still lacking the ability to handle the temporal and spatial correlations of traffic data, and their capacity to learn nonlinear patterns in data is limited.

The development of deep learning technologies in recent years has provided new solutions for network traffic prediction. Deep learning models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and their variants, Long Short-Term Memory networks (LSTM)[9], and Gated Recurrent Units (GRU)[10], have significantly improved the accuracy and efficiency of network traffic prediction through their powerful nonlinear fitting capabilities and the ability to capture long-term dependencies. These models can automatically extract and learn complex patterns in traffic data, providing more powerful and flexible prediction tools to meet the demands of modern network traffic prediction. Ramakrishnan and Vinaya kumar[11, 12] et al. conducted an in-depth analysis of the experimental effects of applying RNN and its variants, LSTM and GRU, on different real-world network traffic datasets. The study found that due to the gated mechanisms in the design of LSTM and GRU, they perform better than traditional RNN models in capturing long-term dependencies in time series data. Therefore, LSTM and GRU have shown higher prediction accuracy in network traffic prediction tasks, making them more suitable for dealing with the complexity and variability of network traffic data. Network traffic is correlated not only in time but also in space. Specifically, network traffic is influenced not only by the traffic in the previous time period but also by the traffic on other links or through other nodes. To model the spatial correlation of network traffic data, the topological structure of network traffic is considered as a graph, and many graph neural network-based network traffic prediction methods have been proposed. However, these previous works have not considered two common scenarios in modeling the spatial and temporal relationships in network traffic prediction: non-correlated adjacency and non-adjacency correlation.

(1) Non-correlated Adjacency

The situation of non-correlated adjacency can be indicated by Figures 1 and 2, where although some node pairs are directly adjacent according to the adjacency matrix, this does not imply similarity in their network traffic patterns. By analyzing the traffic time series and calculating the correlation for specific adjacent node pairs (such as nodes 8 and 9), it is found that the traffic correlation coefficient of these node pairs is below a predetermined threshold, indicating that their traffic patterns are not significantly correlated. This finding challenges the traditional notion that physical or topological proximity must be linked to similarity in traffic patterns, suggesting that more factors need to be considered in network traffic prediction, such as routing policies, traffic management measures, and the roles and functions of nodes. Therefore, methods that directly determine spatial correlation between nodes based on the distance between them cannot accurately describe the correlation between network nodes.

(2) Non-adjacency Correlation

The situation of non-adjacency correlation further emphasizes the complexity of network traffic patterns. By analyzing the adjacency matrix, node pairs that are not directly adjacent can be identified, and then the traffic time series and correlation of these node pairs can be further analyzed. For example, nodes 1 and 9 are not adjacent in the topology, but their traffic time series analysis shows a high degree of correlation, indicating that despite not being directly connected in the network, there exists similarity in their traffic patterns. This scenario highlights the dynamic nature of network traffic and the possible complex dependencies between non-directly adjacent nodes, which may be caused by shared traffic sources, similar application requirements, or similar user behavior patterns.

In response to the two common situations in network traffic prediction mentioned above, and the limitations of existing spatiotemporal prediction models in exploring the physical distance relationships between nodes in network topologies, a network traffic spatiotemporal prediction model based on graph attention and adaptive graph convolution, STGACRN (Spatio-temporal Graph Attention Convolutional Recurrent Neural Network), is proposed. We designed a spatial feature extraction module to capture the dynamic spatial correlation of traffic data, which can integrate the original network topology with adaptively generated graphs. By introducing graph attention and graph convolution operations, it can effectively capture the direct interaction between network nodes and delve deeply into the potential correlations between them. Furthermore, a recurrent neural network structure with residual connections was designed to handle nonlinear temporal correlations and extract temporal features. In this way, STGACRN is capable of modeling temporal and spatial dependencies. The main contributions of this paper are as follows:

1) We propose a new network traffic spatiotemporal prediction model that can effectively capture complex nonlinear spatiotemporal dependencies. Moreover, to avoid the gradient vanishing or exploding problems due to the depth of the model network, residual connections were added to the GRU memory units, leading to the proposal of Res-GRU (Residual-connected GRU).

2) The dynamic correlation between network nodes is captured through the combination of graph attention and graph convolutional neural networks.

3) We conducted extensive experiments on three real-world network traffic datasets. The experiments demonstrate that our proposed method achieves higher prediction accuracy.

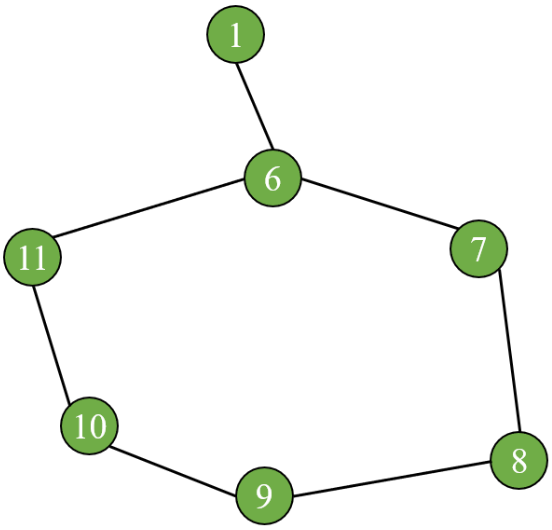


Fig. 1 The adjacency relationship between some network nodes

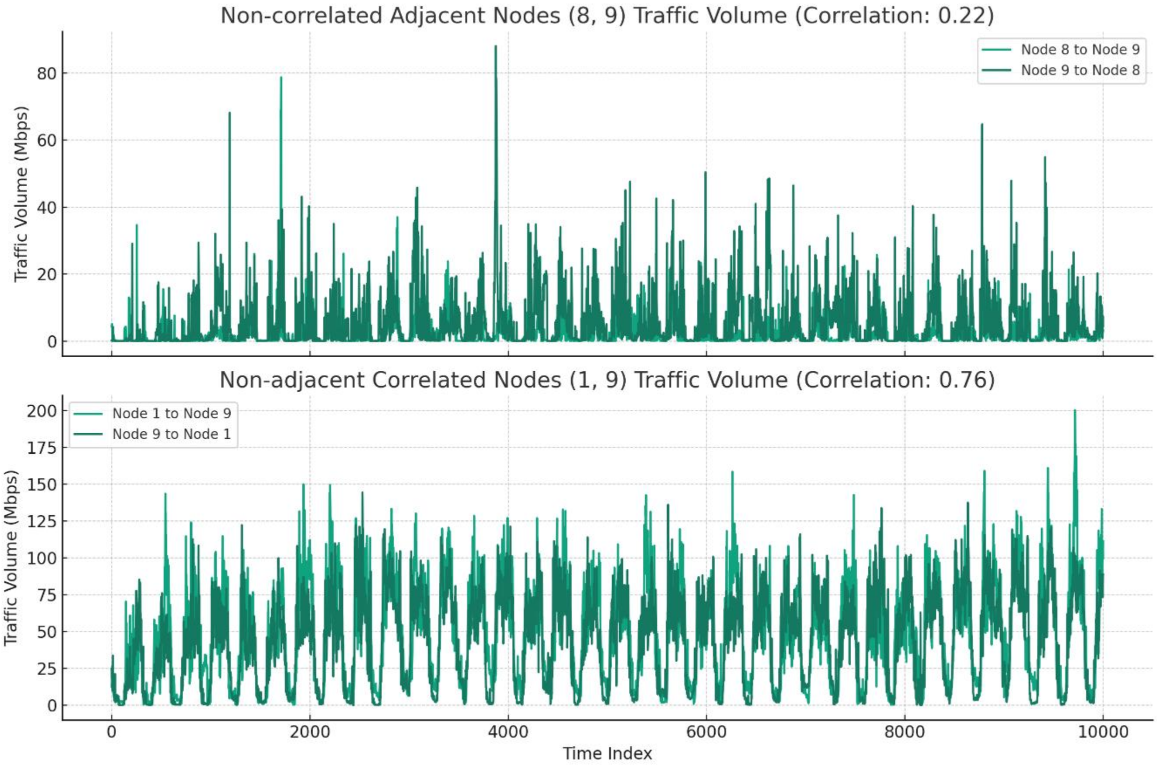


Fig. 2 Two common situations in network traffic prediction

II. METHODOLOGYI

2.1 Temporal feature extraction module

In the field of network traffic prediction, identifying the temporal dependencies of network dynamics is a core challenge. Recurrent neural networks, such as Long Short-Term Memory networks (LSTM) and Gated Recurrent Units (GRU), are widely used in time series prediction tasks and have been proven effective in addressing such issues. GRU is favored for its relatively simplified structure, fewer parameters, and ease of computation and implementation. Related research indicates that with an equal number of parameters, GRU can match LSTM in prediction performance and has a shorter training cycle[13, 14]. Therefore, GRU is employed here to extract the temporal characteristics of network traffic. Moreover, considering the potential issue of vanishing gradients in deep GRU networks, a residual structure[15] is introduced to overcome performance degradation issues that may arise from increased network depth.

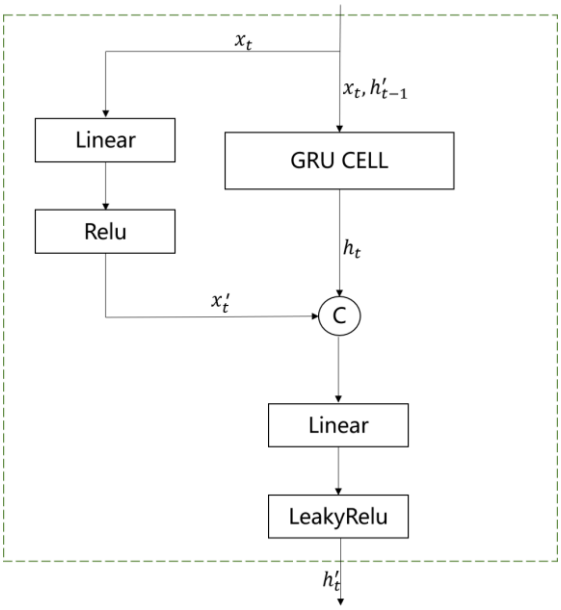


Fig. 3 Res-GRU unit structure

Specifically, a residual connection block is added between each GRU layer, and the structure of the GRU based on residual connections (Res-GRU) is shown in Figure 3, where the update process is shown in Equations (1) to (6).

$$r_t = \sigma([x_t, h_{t-1}^{'}]W_r + b_r)\#(1)$$

$$z_t = \sigma([x_t, h_{t-1}^{'}]W_z + b_z)\#(2)$$

$$c_t = tanh(x_t W_c + r_t * (h_{t-1}^{'} W_c) + b_c)\#(3)$$

$$h_t = z_t * h_{t-1}^{'} + (1 - z_t) * c_t\#(4)$$

$$x_t^{'} = ReLU(x_t W_x + b_x)\#(5)$$

$$h_t^{'} = LeakyReLU([h_t || x_t^{'}]W + b)\#(6)$$

Here, * denotes the Hadamard product (element-wise multiplication operation). If two matrices are multiplied without any special symbols indicated, standard matrix multiplication rules are followed. [·] represents the concatenation operation for vectors or matrices. σ is the sigmoid nonlinear activation function. x_t represents the traffic feature at time t. $h_t^{'}$ denotes the output state at time t. r_t and z_t refer to the reset gate and update gate, respectively. The update gate z_t is responsible for selectively forgetting some information from the previous moment $h_{t-1}^{'}$ and integrating some information from the current c_t. The reset gate r_t is used to determine the relevance of past information $h_{t-1}^{'}$ to future traffic prediction. W_r, W_z, W_c, W_x, W, and b_r, b_z, b_c, b_x, b represent the learnable weights and bias terms during the training process, respectively.

2.2 Spatial feature extraction module

The Graph Attention Network (GAT) dynamically computes the weights between adjacent nodes using the attention mechanism, effectively aggregating the features of adjacent nodes to the central node. This method, by operating on each vertex and traversing all the vertices in the graph for computation, overcomes the limitations of relying on the Laplacian matrix, focusing more on the interactions between nodes rather than the global structure of the graph. Therefore, utilizing the graph attention mechanism to extract spatial features of network traffic data can effectively address the issue of non-correlated adjacency between nodes.

The implementation of GAT relies on two main steps: computing attention coefficients and aggregating node features. In the process of updating the feature vector of node i, attention scores of all adjacent nodes are first calculated, and then these scores are weighted and summed with the features of each adjacent node to obtain the updated feature of node i. The initial feature set of nodes is defined as X, expressed as X={$\vec{x}_1$,$\vec{x}_2$,…,$\vec{x}_N$}, where $\vec{x}_i \in R^F$, N represents the total number of nodes, and F represents the number of features per node. In the Graph Attention Network, the mutual importance between nodes is quantified by calculating the attention coefficient e_{ij}, where e_{ij} is obtained by the dot product of linearly transformed vectors processed by the LeakyReLU activation function, as shown in equation (7).

$$e_{ij} = LeakyReLU\left(\vec{a}^T\left[W\vec{x}_i \parallel W\vec{x}_j\right]\right)\#(7)$$

Here, $\vec{x}_i$ and $\vec{x}_j$ respectively represent the feature vectors of nodes i and j. $W \in R^{F \times F^{'}}$ is the node feature transformation matrix, used for feature extraction. e_{ij} characterizes the importance level of node i to node j. The vector $\vec{a}$ represents the weight parameters. LeakyReLU(·) is a nonlinear activation function.

To optimize the weight distribution of each node to its neighboring nodes, the softmax function is used to normalize the computed attention coefficients, ensuring that the sum of attention weights for all neighboring nodes equals 1. The calculation formula for the normalized attention coefficient α_{ij} is shown in equation (8).The updated representation of node features is shown in equation(9).

$$\alpha_{ij} = softmax(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})}\#(8)$$

$$h_i^{\ '} = \sigma\left(\sum_{j \in N_i} \alpha_{ij} W \cdot h_j\right)\#(9)$$

Here, α_{ij} represents the normalized attention coefficient of node i towards node j. N_i denotes the set of neighboring nodes of node i. W represents the weight matrix. By this method, information from neighboring nodes is aggregated to obtain a set of updated features $X^{'}$ = {$\vec{x}_1^{'}$,$\vec{x}_2^{'}$,…, $\vec{x}_N^{'}$}.

In the field of network traffic prediction, extracting spatial features is key to understanding and predicting network behavior. Although Graph Attention Networks (GAT), with their attention mechanisms, have significant advantages in capturing spatial correlations between nodes and can dynamically learn the weight distribution among adjacent nodes, addressing the issue of non-correlated adjacency between nodes. However, relying solely on GAT does not fully exploit the implicit links in the network, that is, the non-adjacency correlations between nodes.This limitation stems from the inherent complexity of the network traffic prediction problem, which includes not only the network's topology, i.e., the explicit connections between nodes, but also the implicit dependencies between nodes formed during the traffic transmission process, which may not be directly reflected in the network's topology.

Although the standalone application of GAT can capture spatial correlations based on direct interactions between nodes, given the characteristics of network traffic, which include clear starting and ending nodes and their transmission paths, it is difficult to fully understand the implicit, indirect dependencies between nodes through the topology of the graph and direct connections alone.Therefore, it is crucial to consider the implicit connections caused by network flow, in addition to the edges of the graph, for accurate prediction of network traffic.

To overcome this limitation and enhance the model's performance in spatial feature extraction, combining Graph Convolutional Networks (GCN) with adaptive graph generation methods is particularly crucial.GCN can identify local patterns of node features in the topology of the graph through its graph convolution operation, while the adaptive graph generation strategy allows the model to learn and recognize the potential dependencies between nodes, transcending the limits of the original graph's topology.The adaptive adjacency matrix A_{adp} corresponding to this method is shown in equation (10).

$$A_{adp} = softmax\left(ReLU(E_S E_D^T)\right)\#(10)$$

Here, E_S and E_D represent the embeddings of the source and target nodes, respectively, with $E_S, E_D \in R^{N\times d}$. The ReLU function is used to remove weak connections after the multiplication of $E_S$and E_D, and the softmax function is applied to normalize the adaptive adjacency matrix.

Based on the GAT and GCN models, the original network topology structure is combined with the method of adaptive graph generation to form a new spatial feature extraction module—A-GCN. The definition of A-GCN is illustrated in equations (11) - (14).

$$X_{gat} = \hat{g}\left(A_{adj}, X\right)\#(11)$$

$$X_{gcn} = softmax\left(ReLU(E_S E_D^T)\right)XW + b\#(12)$$

$$R_g = \sigma\left(X_{gat} W_{r1} + X_{gcn} W_{r2} + b\right)\#(13)$$

$$S = R_g * X_{gat} + \left(1 - R_g\right) * X_{gcn}\#(14)$$

Herein, $\hat{g}(\cdot)$ represents the graph attention computation process. X_{gat} denotes the direct spatial features obtained through the aggregation of GAT and network topology, while X_{gcn} signifies the indirect spatial features aggregated through GCN and the adaptive generation matrix. S represents the final comprehensive spatial features obtained through A-GCN. R_g is a gating mechanism, the utilization of which can adaptively allocate weights to direct and indirect spatial features. This enables the spatial feature extraction module to not only dynamically capture the importance of explicit dependencies between nodes using GAT but also process the implicit dependencies between nodes through GCN, achieving comprehensive extraction of spatial correlations in network traffic.

2.3

To leverage the spatiotemporal correlation of traffic data using Res-GRU and A-GCN, the Spatiotemporal Graph Attention Convolutional Recurrent Neural Network (STGACRN) with node information enhancement is proposed. The structure of STGACRN is shown in Figure 4. The computation process of STGACRN is demonstrated by equations (15) - (24). $\hat{g}(\cdot)$ represents the computation process of graph attention, with the output s_t being the network traffic sequence with spatial features obtained from A-GCN.

$$X_{gat} = \hat{g}\left(A_{adj}, x_t\right)\#(15)$$

$$X_{gcn} = softmax\left(ReLU(E_S E_D^T)\right)x_t W_{gcn} + b_{gcn}\#(16)$$

$$R_g = \sigma\left(X_{gat} W_{r1} + X_{gcn} W_{r2} + b_r\right)\#(17)$$

$$s_t = R_g * X_{gat} + \left(1 - R_g\right) * X_{gcn}\#(18)$$

$$r_t = \sigma([s_t, h'_{t-1}]W_r + b_r)\#(19)$$

$$z_t = \sigma([s_t, h'_{t-1}]W_z + b_z)\#(20)$$

$$c_t = tanh(s_t W_c + r_t * (h'_{t-1} W_c) + b_c)\#(21)$$

$$h_t = z_t * h'_{t-1} + (1 - z_t) * c_t\#(22)$$

$$x'_t = ReLU(x_t W_x + b_x)\#(23)$$

$$h'_t = LeakyReLU([h_t, x'_t]W + b)\#(24)$$

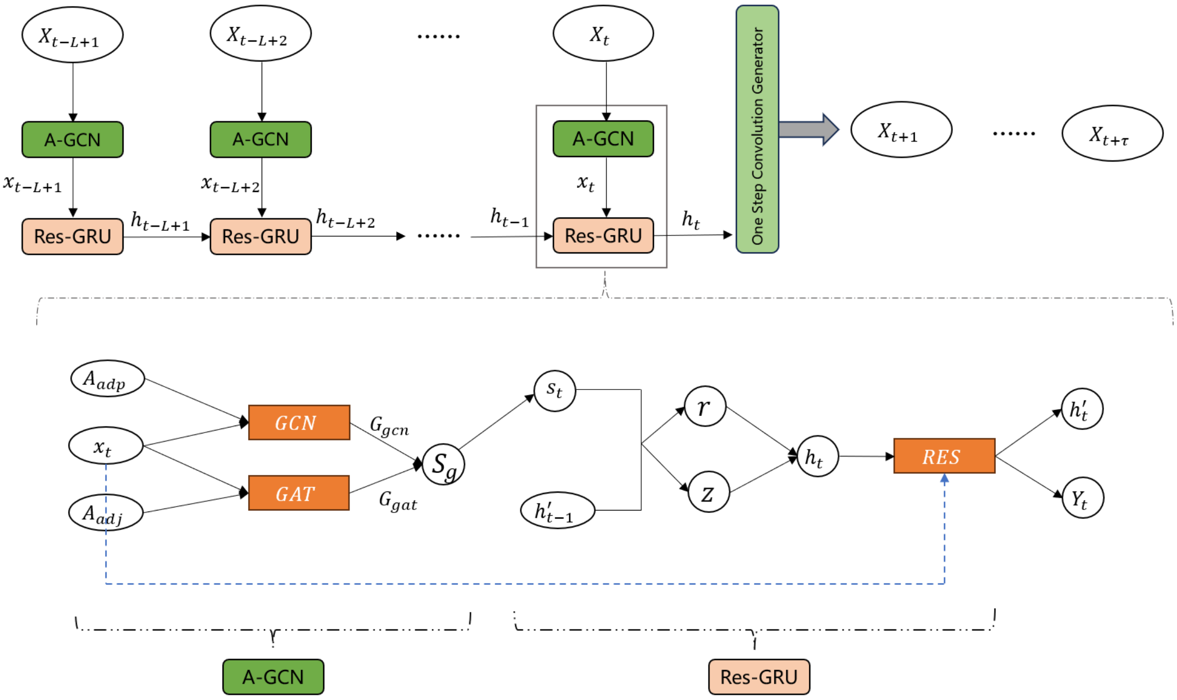


Fig. 4 Structure of STGACRN

To achieve multi-step traffic prediction in one step, multiple layers of STGACRN are stacked as an encoder to capture high-level feature representations of nodes, outputting these as $h_t^{'} \in R^{N\times N\times d}$. Subsequently, a one-step convolution generator can directly obtain traffic predictions for the next τ steps for all nodes by applying a linear transformation to project the representation from $R^{N\times N\times d}$ to $R^{\tau\times N\times N}$. The one-step convolution generator achieves the capability to directly predict multiple future time steps from the current network state, avoiding potential error accumulation in traditional cyclic prediction strategies and significantly improving prediction efficiency. This not only optimizes the prediction process but also enhances the model's usability and accuracy in practical applications.

In summary, the STGACRN model can effectively address the spatiotemporal correlation modeling problem in network traffic prediction. Initially, the A-GCN module leverages the original network topology and adaptive adjacency matrix generation, along with the properties of graph attention networks and graph convolutional neural networks, to effectively capture dynamic spatial correlations between network nodes. Then, the Res-GRU module is utilized to capture the temporal correlation of network traffic. Finally, to avoid error accumulation, the One Step Convolution Generator is used to achieve multi-step prediction outputs.

III. EXPERIMENTS

3.1 Datasets

The experiment utilized three publicly available network traffic datasets: Abilene, CERNET, and GEANT.

The Abilene dataset originates from the Abilene backbone network of the United States Research and Education Network, a wide-area network with a real topology that connects 12 major U.S. cities as network nodes through 30 undirected links, reflecting the network traffic transmission paths between these cities.This dataset covers traffic data from March 1 to September 10, 2004, collecting the network's traffic bandwidth values every 5 minutes, resulting in 48,096 traffic matrices.Each traffic matrix detailedly records the traffic conditions from 12 source nodes to 12 destination nodes, forming 144 OD (Origin-Destination) flow pairs.

The CERNET dataset comes from the China Education and Research Computer Network, which is the largest national academic network in China, covering colleges, universities, and research institutions across the country.As of 2013, this network traffic dataset consists of 12 nodes and 32 undirected links. The CERNET dataset's traffic data collection period was from 22:10 on February 19, 2013, to 15:20 on March 26, 2013, with traffic data collected every 5 minutes, resulting in 9,999 traffic matrices.

The GEANT dataset is derived from the GEANT network, a large research and education network covering Europe, available for use by research institutions and universities. The GEANT network possesses a complex, real network topology. As of 2005, the network included 23 nodes and 74 undirected links, reflecting the network connections and traffic exchange between major European research centers. The GEANT dataset's traffic data collection period was from 15:30 on May 4, 2005, to 07:45 on August 31, 2005, with traffic data collected every 15 minutes, resulting in 10,769 traffic matrices.

In experimental research, the dataset is divided using a 6:2:2 ratio, meaning that based on the timeline of data collection, the first 60% of the dataset is allocated as the training set, the last 20% as the test set, and the remaining portion serves as the validation set. During the model training process, data from the past 12 time steps are used as input to predict data for the next 12 time steps. The initial learning rate is set to 0.003 in the experiment, the batch size is set to 64, and the ADAM optimizer is selected to optimize the training process.

3.2 Evaluation Metrics

The prediction results are evaluated using the commonly used Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The smaller the values of RMSE and MAE, the closer the prediction results are to the actual values, indicating better prediction performance. The corresponding calculation formulas for these evaluation metrics are shown in equations (25) and (26).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}^{n}(\hat{y}_i - y_i)^2 \, \#(25)}$$

$$MAE = \frac{1}{n}\sum_{i}^{n}|\hat{y}_i - y_i| \, \#(26)$$

Here, n denotes the total number of samples, $\hat{y}_i$ represents the predicted value, and y_i represents the actual value.

3.3 Baseline Methods

The comparative models selected include classic time series prediction models LSTM, GRU, TCN, and the spatiotemporal prediction model AGCRN.A brief description of the four comparative models follows:

（1）LSTM: LSTM effectively addresses the long-term dependency issues in long sequence data through a special gating mechanism (input gate, forget gate, and output gate). It can remember and forget information when dealing with complex sequence data, thus performing excellently in various time series prediction tasks.

（2）GRU: GRU is a simplified version of LSTM, combining the forget gate and input gate into a single update gate and introducing a reset gate to control the flow of information, thereby reducing the number of model parameters and improving training efficiency. Despite its simplified structure, GRU can compete with LSTM in many tasks, especially when the dataset is small or computational resources are limited.

（3）TCN: TCN processes time series data through one-dimensional convolutional layers and residual connections, especially employing causal convolutions (ensuring only data up to the current moment is used for predictions) and dilated convolutions (expanding the receptive field to capture long-term dependencies), allowing TCN to handle long sequence data while avoiding the gradient vanishing or explosion issues of traditional RNN models. With the advantage of parallel computation, TCN shows good performance in handling large-scale time series data, especially when capturing long-distance dependencies is required.

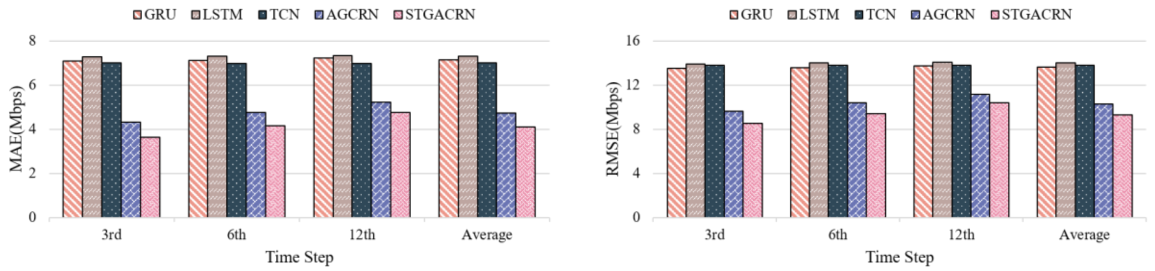
（4）AGCRN[16]: AGCRN automatically captures the spatial and temporal correlations of traffic data by introducing Node Adaptive Parameter Learning (NAPL) and Data Adaptive Graph Generation (DAGG) modules, without relying on predefined spatial connection graphs (the original network topology).

3.4 Experimental Results

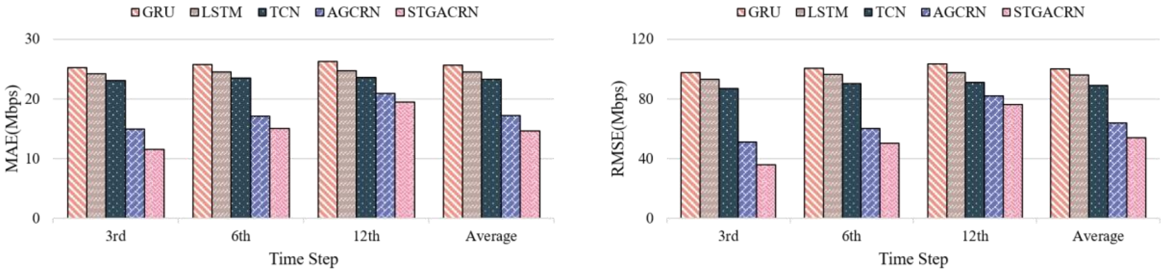
Table 1 presents the prediction results of STGACRN and baseline models on the Abilene, CERNET, and GEANT datasets at different time steps and on average.From Table 1, it can be observed that STGACRN achieves better prediction performance on all three network traffic datasets.

Tab. 1 Traffic prediction results based on different models of Abilene, CERNET and GEANT datasets.

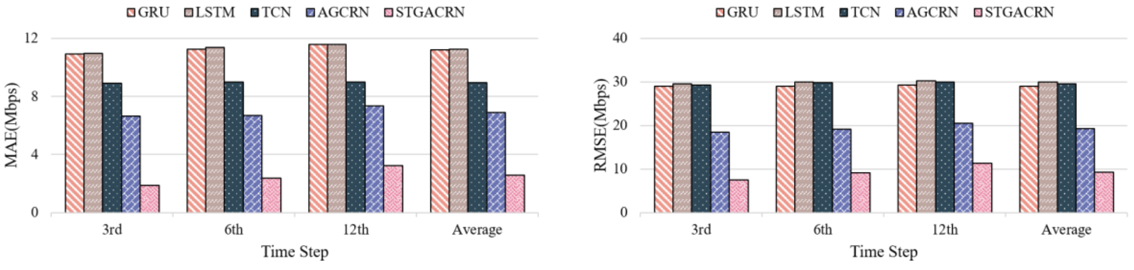
Dataset	Model	3rd		6th		12th		Average	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Abilene	GRU	7.10	13.55	7.12	13.57	7.23	13.72	7.15	13.62
	LSTM	7.27	13.93	7.31	14.02	7.33	14.05	7.31	14.01
	TCN	7.00	13.80	6.99	13.81	6.99	13.81	7.00	13.81
	AGCRN	4.31	9.61	4.77	10.40	5.24	11.19	4.73	10.31
	STGACRN	**3.63**	**8.51**	**4.16**	**9.43**	**4.77**	**10.39**	**4.09**	**9.32**
CERNET	GRU	25.22	97.49	25.74	100.70	26.31	103.49	25.70	100.28
	LSTM	24.18	93.33	24.55	96.57	24.70	97.87	24.50	95.90
	TCN	23.11	86.88	23.50	90.16	23.59	90.91	23.33	88.93
	AGCRN	14.94	51.35	17.15	60.19	20.96	82.15	17.24	63.82
	STGACRN	**11.56**	**35.79**	**15.08**	**50.39**	**19.52**	**76.11**	**14.67**	**54.04**
GEANT	GRU	10.91	28.99	11.25	29.01	11.58	29.22	11.21	29.05
	LSTM	10.95	29.56	11.39	30.03	11.60	30.26	11.26	29.92
	TCN	8.90	29.35	8.98	29.77	8.98	29.95	8.94	29.60
	AGCRN	6.65	18.51	6.68	19.15	7.36	20.47	6.90	19.27
	STGACRN	**1.89**	**7.51**	**2.35**	**9.14**	**3.25**	**11.36**	**2.56**	**9.23**



Comparison of MAE and RMSE indexes of Abilene network traffic dataset



Comparison of MAE and RMSE indexes of CERNET network traffic dataset



Comparison of MAE and RMSE indexes of GEANT network traffic dataset

Fig. 5 Comparison of MAE and RMSE indexes of different models in three network traffic datasets

Figure 5 visualizes the comparative experimental results of the model and the baseline model on three network traffic datasets. It can be seen from Table 1 and Figure 5 that in the prediction of STGACRN on three different datasets (Abilene, CERNET and GEANT) the STGACRN model shows better performance than other models in 3rd time step, 6th time step and 12th time step prediction and overall average performance. Especially in the predicted 12th time step, the MAE and RMSE indicators of the STGACRN model perform particularly well.

For the Abilene dataset: The average MAE and RMSE of the STGACRN model were 4.09 and 9.32, respectively, significantly better than GRU, LSTM, and TCN, demonstrating its effectiveness in capturing and predicting the temporal and spatial characteristics of network traffic.For the CERNET dataset: In a relatively larger and more complex network, the average MAE and RMSE of the STGACRN model were 14.67 and 54.04, respectively, surpassing traditional RNN models and the latest spatiotemporal graph models, proving STGACRN's strong capability in handling large-scale network traffic data.For the GEANT dataset: STGACRN outperformed the comparison models in MAE and RMSE metrics at all time steps, especially at the 12th time step, indicating its excellent long-term prediction ability.

The results above indicate that the STGACRN model effectively integrates temporal and spatial information, innovatively employing two types of graph neural network techniques to process graph-structured data, thereby enhancing spatiotemporal prediction of network traffic. In contrast, while LSTM and GRU have their advantages in handling time series problems, they lack the capability to process graph-structured data. TCN provides a wider receptive field through dilated convolution, but its ability to capture complex network topologies remains limited. Although the AGCRN model introduces an adaptive graph structure, it performs less effectively than STGACRN in experiments, due to STGACRN's more effective temporal feature capture and spatiotemporal feature fusion mechanism.

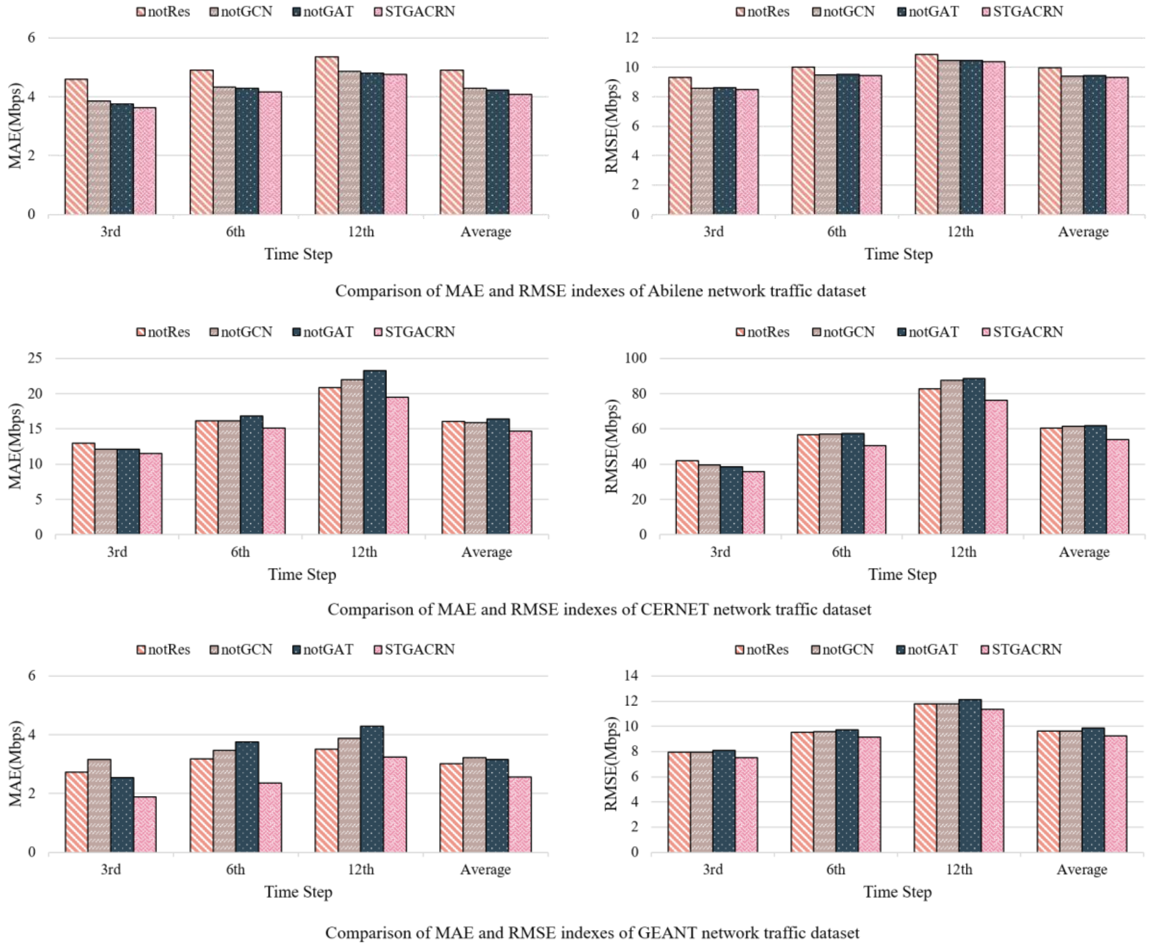
3.5Ablation Studies

To verify the effectiveness of different modules in the proposed model, an ablation study was conducted by removing key modules of the model. For this purpose, three variants of the model were designed: notRes, notGCN, and notGAT. NotRes removed the residual connections in the model and utilized standard GRU units. notGCN removed the GCN and the adaptive adjacency matrix, retaining only the GAT to process the original adjacency matrix. notGAT removed the graph attention network and the original adjacency matrix, keeping only the GCN to process the adaptive adjacency matrix. The results of the ablation study are shown in Table 2.

Tab. 2 Ablation experimental results

Dataset	Model	3rd		6th		12th		Average	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Abilene	notRes	4.60	9.33	4.91	10.03	5.36	10.89	4.91	9.99

	notGCN	3.85	8.59	4.33	9.47	4.87	10.46	4.30	9.40
	notGAT	3.75	8.64	4.30	9.53	4.80	10.49	4.23	9.46
	STGACRN	**3.63**	**8.51**	**4.16**	**9.43**	**4.77**	**10.39**	**4.09**	**9.32**
CERNET	notRes	13.00	42.14	16.13	56.70	20.83	82.77	16.08	60.32
	notGCN	12.15	39.56	16.18	57.02	22.01	87.66	15.92	61.53
	notGAT	12.10	38.70	16.80	57.50	23.26	88.58	16.44	61.91
	STGACRN	**11.56**	**35.79**	**15.08**	**50.39**	**19.52**	**76.11**	**14.67**	**54.04**
GEANT	notRes	2.73	7.94	3.17	9.55	3.50	11.78	3.01	9.62
	notGCN	3.15	7.95	3.47	9.58	3.87	11.80	3.23	9.61
	notGAT	2.55	8.09	3.75	9.71	4.30	12.10	3.16	9.85
	STGACRN	**1.89**	**7.51**	**2.35**	**9.14**	**3.25**	**11.36**	**2.56**	**9.23**



Comparison of MAE and RMSE indexes of Abilene network traffic dataset

Comparison of MAE and RMSE indexes of CERNET network traffic dataset

Comparison of MAE and RMSE indexes of GEANT network traffic dataset

Fig. 6 Comparison of MSE and RMSE indexes of STGACRN model and its variants in three network traffic datasets

Figure 6 visualizes the ablation experimental results of the model and its variants on three network traffic datasets. According to Table 2 and Figure 6, the complete model STGACRN shows better prediction performance than the variant model in all data sets, whether in the future time steps of 3rd, 6th, 12th, or average results. Specifically, the STGACRN model has an average MAE of 4.09 on the Abilene dataset, while the MAE of the notRes model, which lacks residual connections, is 4.91, showing a significant performance decline. Similarly, on the CERNET dataset, the average MAE of STGACRN is 14.67, while the MAE of notRes increases to 16.08.This indicates that residual connections play a crucial role in the model, especially in capturing long-term dependencies and preventing gradient vanishing during training.

For the notGCN variant, the decrease in predictive performance across datasets also highlights the role of GCN based on adaptive adjacency matrices in capturing potential correlations between nodes. For example, on the GEANT dataset, the average MAE of notGCN is 3.23, compared to 2.56 for STGACRN, indicating a performance decrease.

The performance decline of the notGAT model across all datasets confirms the importance of the attention mechanism in highlighting key relationships between nodes. Particularly on the Abilene dataset, the average MAE of not GAT is 4.23, slightly higher than STGACRN's 4.09, indicating that the graph attention mechanism can effectively help the model aggregate features of key adjacent nodes, thereby improving prediction accuracy.

Overall, the ablation study results verified the predictive accuracy of the complete model across multiple time steps, while also highlighting the significant contributions of residual connections, GAT, and GCN based on adaptive adjacency matrices to enhancing network traffic prediction accuracy. The synergistic action of these components enables the STGACRN model to effectively address the issues of non-correlated adjacency and non-adjacency correlation, thereby achieving superior predictive performance. It also proves its effectiveness in spatiotemporal prediction of network traffic.

IV. Conclusion

To enhance the accuracy of network traffic prediction, a Spatio-temporal Graph Attention Convolutional Recurrent Neural Network (STGACRN) based on graph attention and adaptive graph convolution has been proposed. To dynamically capture spatial correlations between nodes, this model addresses the challenges of non-adjacency correlation and non-correlated adjacency by integrating Graph Attention Networks (GAT) with Graph Convolutional Networks (GCN) based on adaptive graphs. Simultaneously, to avoid issues such as gradient vanishing or exploding during cyclic prediction, a Gated Recurrent Unit (GRU) based on residual connections has been designed to capture the temporal correlations in traffic data. The model was evaluated on three real-world network traffic datasets and compared with GRU, LSTM, TCN, and AGCRN models. The experimental results show that the model exhibits superior predictive performance at various forecasting levels, demonstrating its effectiveness in network traffic prediction.

References

[1] Knieps G. Digitalization technologies: the evolution of smart networks [M]. 2021: 43-58.

[2] Gan M, Peng H. Stability analysis of RBF network-based state-dependent autoregressive model for nonlinear time series [J]. Applied Soft Computing, 2012, 12(1): 174-81.

[3] Laner M, Svoboda P, Rupp M. Parsimonious Fitting of Long-Range Dependent Network Traffic Using ARMA Models [J]. IEEE Communications Letters, 2013, 17(12): 2368-71.

[4] Moayedi H Z, Masnadi-Shirazi M A. Arima model for network traffic prediction and anomaly detection[C]; proceedings of the 2008 International Symposium on Information Technology, 2008.

[5] Yu Y, Wang J, Song M, et al. Network Traffic Prediction and Result Analysis Based on Seasonal ARIMA and Correlation Coefficient[C]; proceedings of the 2010 International Conference on Intelligent System Design and Engineering Application, 2010.

[6] Habibi O, Chemmakha M, Lazaar M. Imbalanced tabular data modelization using CTGAN and machine learning to improve IoT Botnet attacks detection [J]. Engineering Applications of Artificial Intelligence, 2023, 118: 105669.

[7] Bermolen P, Rossi D. Support vector regression for link load prediction [J]. Computer Networks, 2009, 53(2): 191-201.

[8] Liu X W, Fang X M, Qin Z H, et al. A Short-Term Forecasting Algorithm for Network Traffic Based on Chaos Theory and SVM [J]. Journal of Network and Systems Management, 2011, 19(4): 427-47.

[9] Hochreiter S, Schmidhuber J. Long Short-Term Memory [J]. 1997, 9(8 %J Neural Comput.): 1735–80.

[10] Cho K, Merrienboer B v, Gülçehre Ç, et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine

Translation[C]; proceedings of the Conference on Empirical Methods in Natural Language Processing, 2014.

[11] Ramakrishnan N, Soni T. Network Traffic Prediction Using Recurrent Neural Networks[C]; proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018.

[12] Vinayakumar R, Soman K P, Poornachandran P. Applying deep learning approaches for network traffic prediction[C]; proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017.

[13] Chung J, Gülçehre Ç, Cho K, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling [J]. 2014, abs/1412.3555.

[14] Li Y J, Yang Y, Zhu K, et al. Clothing Sale Forecasting by a Composite GRU-Prophet Model With an Attention Mechanism [J]. IEEE Transactions on Industrial Informatics, 2021, 17(12): 8335-44.

[15] He K M, Zhang X Y, Ren S Q, et al. Deep Residual Learning for Image Recognition[C]; proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[16] Bai L, Yao L N, Li C, et al. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting [J]. 2020, abs/2007.02842.