Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.057494

ARTICLE

MSSTGCN: Multi-Head Self-Attention and Spatial-Temporal Graph Convolutional Network for Multi-Scale Traffic Flow Prediction

Xinlu Zong*, Fan Yu, Zhen Chen and Xue Xia

School of Computer Science, Hubei University of Technology, Wuhan, 430068, China

*Corresponding Author: Xinlu Zong. Email: zongxinlu@hbut.edu.cn

Received: 19 August 2024 Accepted: 07 November 2024 Published: 17 February 2025

ABSTRACT

Accurate traffic flow prediction has a profound impact on modern traffic management. Traffic flow has complex spatial-temporal correlations and periodicity, which poses difficulties for precise prediction. To address this problem, a Multi-head Self-attention and Spatial-Temporal Graph Convolutional Network (MSSTGCN) for multiscale traffic flow prediction is proposed. Firstly, to capture the hidden traffic periodicity of traffic flow, traffic flow is divided into three kinds of periods, including hourly, daily, and weekly data. Secondly, a graph attention residual layer is constructed to learn the global spatial features across regions. Local spatial-temporal dependence is captured by using a T-GCN module. Thirdly, a transformer layer is introduced to learn the long-term dependence in time. A position embedding mechanism is introduced to label position information for all traffic sequences. Thus, this multi-head self-attention mechanism can recognize the sequence order and allocate weights for different time nodes. Experimental results on four real-world datasets show that the MSSTGCN performs better than the baseline methods and can be successfully adapted to traffic prediction tasks.

KEYWORDS

Graph convolutional network; traffic flow prediction; multi-scale traffic flow; spatial-temporal model

1 Introduction

The development of urbanization has brought increasingly serious traffic congestion [1] while giving people convenience. Intelligent Transportation System (ITS) [2] macroscopically regulates urban traffic through sensing, and control combined with data analysis and other information and communication technologies, which contributes greatly to smart cities. As a significant task of ITS [3], traffic prediction can not only provide a decision basis for traffic managers [4] but also provide advice for people's travel [5]. The main goal of traffic forecasting is to predict future traffic trends by analyzing historical traffic data. Early methods that have been used for traffic prediction in past research include classical statistical methods [6] and methods based on machine learning [7]. However, traditional methods do not perform well due to the complex nonlinear nature of traffic data. At present, approaches based on deep learning have shown their superiority in traffic prediction [8].

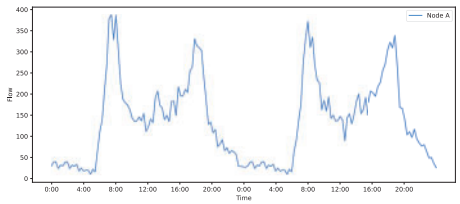
Traffic flow data differs from other simple time series data due to the influence of spatial factors. Deep learning methods can effectively capture high-dimensional features, leading to better prediction

results, such as recurrent neural network (RNN) and its variants long short-term memory (LSTM) [9] and gated recurrent unit (GRU) [10] which have demonstrated excellent prediction performance on sequence data. In addition, it has been observed that traffic flow is influenced not only by historical patterns but also by the spatial relative position within the topological road network [11]. Therefore, many studies have used graph neural networks (GNN) combined with RNN for spatial-temporal traffic prediction [12,13]. For example, Zhao et al. [14] proposed a diffusion convolutional recurrent neural network (DCRNN). Li et al. [15] presented a temporal graph convolutional network (T-GCN). The two models use graph convolutional networks (GCN) and RNNs to learn the spatial features of traffic flow.

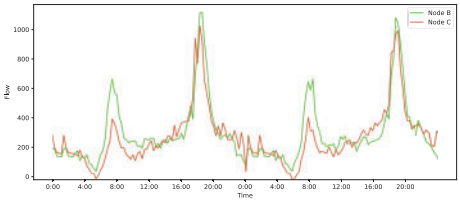
While most traffic prediction methods prove effective, three crucial issues still require attention:

First, traffic flow on the same road, in reality, exhibits periodicity, which correlates with the cyclic nature of people's activities [16]. Fig. 1a represents the location distribution of different road nodes. As depicted in Fig. 1b, the traffic state of node A shows periodicity after a certain number of cycles. Guo et al. [17] integrated daily and hourly periodic data as part of the input. Although their method performed well in short-term prediction, the weekly periodic impact was not considered, and thereby long-term temporal correlations were ignored. Therefore, it is necessary to extract periodic multi-scale time levels (e.g., hourly, weekly, daily) for traffic flow patterns.

(a) Location distribution of road nodes



(b) Traffic flow distribution of node A in two days



(c) Distribution of traffic flow at nodes B and C over two days

Figure 1: Spatial-temporal correlation of nodes at different geographic locations

Second, most existing traffic prediction methods primarily focus on spatial road information within the vicinity. However, road traffic in cities that are far apart should be correlated [18] and more complex compared to the neighborhood space. For example, nodes B and C in Fig. 1c have similar traffic flows, but the two areas are not adjacent to each other, which implies that there may be hidden traffic pattern associations between them. Therefore, modeling cross-regional traffic dependency patterns in a global context is necessary.

Third, traffic flows are not only sequentially dependent on time, but may also have hidden long-term dependencies [19]. The long loop structure of LSTM and GRU fixation makes each temporal node strongly dependent on the previous temporal node, and long-term dependencies cannot be

effectively captured when the temporal spacing between node flows is too long. In addition, although the attention-based methods can capture long-term dependence, they ignore the sequential nature of the traffic sequence.

Aiming at the above problems, we investigate prediction methods for multi-scale periodic traffic flow to fuse the adjacent local spatial and cross-regional global spatial information. Thus, both the short-term and long-term dependencies of traffic data can be captured. This paper introduces a Multi-head Self-attention and Spatial-Temporal Graph Convolutional Network (MSSTGCN) for multi-scale traffic flow prediction.

The primary contributions of this paper are as follows:

The traffic flow is divided into multi-scale (i.e., hourly, daily, weekly) forms to explore the periodicity from multiple time dimensions, which allows for a comprehensive analysis of traffic patterns that vary over different time scales. Multi-scale modeling can enhance the ability to identify and predict complex traffic patterns within these distinct temporal contexts.

A graph attention residual network (GARN) layer which is capable of fusing spatial information of different dimensions is introduced to learn global spatial dependencies. This layer can not only enhance the model's ability to capture intricate relationships within the traffic data but also improve the generalization ability across various traffic scenarios.

This hybrid architecture leverages the strengths of GRU in capturing sequential dependencies while benefiting from the transformer attention mechanism. The incorporated positional encoder enables the attention mechanism to recognize the order of sequences. This combination can enhance the model's capability to capture complex temporal patterns, thereby improving prediction accuracy.

Experiments on real traffic datasets of the MSSTGCN model and other baseline methods are conducted. The experimental results indicate that MSSTGCN outperforms other models.

The remainder of this paper is structured as follows: Section 2 reviews related work on traffic prediction. Section 3 defines the traffic prediction problem and outlines the proposed model. Section 4 describes experiments conducted with real traffic datasets and includes a comparative analysis with baseline approaches. Finally, Section 5 offers a summary and outlook for the paper.

2 Related Work

In the early stages, traffic forecasting is often considered a type of time series forecasting. Early classical traffic forecasting approaches include Historical Average (HA) [20] and Auto-Regressive Integrated Moving Average (ARIMA) [21]. However, simple regression statistics do not model the nonlinearity and complexity of traffic data, making traditional methods limited. To tackle this issue, various machine learning algorithms have been introduced to handle complex traffic data. For instance, Wang et al. [22] introduced a two-pattern recognition K-Nearest Neighbors (KNN) model for traffic prediction, while Luo et al. combined discrete Fourier transform with Support Vector Regression (SVR) [23] to predict residual sequences. However, these machine learning algorithms require manual design of artificial features and cannot effectively model dynamic and complex traffic data.

In recent years, deep learning has increasingly become a dominant approach in traffic prediction. Deep learning [24], as a branch of machine learning, utilizes multi-layer neural networks for feature learning and pattern recognition. It is characterized by its powerful ability to automatically extract complex feature representations from raw data. Fu et al. [9] utilized LSTM and GRU and achieved

better results than the traditional methods. Xu et al. [25] constructed an LSTM-based sequential network model for predicting the demand of urban taxis. A standalone RNN model does not incorporate the spatial information of traffic data. Therefore, studies [26] have employed Convolutional Neural Networks (CNNs) to capture the spatial dependencies among roads. Zhang et al. [27] embedded CNN and LSTM into a Generative Adversarial Nets (GAN) framework to capture spatial-temporal dependencies. However, CNNs can only handle 2D grid images, and real traffic topology roads are intricate and complex, making the spatial fitting ability of CNNs limited. GNNs are widely employed for traffic prediction because they can model the complex relationships and dependencies inherent in traffic data. Given the temporal nature of traffic data, GNNs are often combined with temporal modeling techniques such as RNNs for learning the spatial-temporal dependencies. Chen et al. [28] combined GCN with LSTM to predict traffic flow. Su et al. [29] combined spatial gated linear unit block, GCN, and LSTM to explore the interaction between multiple traffic parameters. Wang et al. [30] integrated GAN and GCN to automatically model dynamic spatial-temporal states. In addition, considering that GCN only aggregates spatial information for local neighborhood nodes and ignores the global spatial relationships between regional nodes. Zhao et al. [31] devised a spatial attention mechanism by evaluating relationships between nodes. Zhang et al. [32] designed a graph-neural hierarchical structure to maintain spatial dependencies at both local and global scales, integrating the Graph Attention Network (GAT) and graph diffusion mechanism. GAT and graph diffusion mechanisms can focus on more spatial dependencies. But vanishing gradient may occur in deep network learning, leading to weaker learning of spatial dependencies.

In particular, RNN maintains the temporal relationships of traffic flow data through a gating mechanism. However, its long cyclic structure makes it challenging to extract links between time nodes that are far apart [33]. Bai et al. [34] proposed the Attention Temporal Graph Convolutional Network (A3T-GCN) model, which introduced the attention mechanism by adjusting weights between different time nodes. However, a single attention mechanism tends to give equal importance to each time node and fails to recognize the sequential nature of traffic sequences. The Multi-Head SpatioTemporal Attention Graph Convolutional Network (MHSTA-GCN) [35] combines GCN, GRU, and multi-head attention modules. In the model, GCN is used to learn the basic features of nodes by capturing the complex spatial topology of the graph, and GRU is introduced to capture the dynamic time dependence through multi-head attention mechanism. The integration of GRU and attention mechanism takes into account both the global and temporal sequential nature of traffic flow. Dong et al. [36] proposed a Multi-scale Temporal and Enhance Spatial transformer (MTESformer) model, which combined multi-head self-attention mechanism with a multi-scale convolutional unit to learn traffic flow patterns at different scales and capture long-term dependencies. However, the model ignored short-term temporal correlation and spatial correlation. Chai et al. [37] presented a Spatio-Temporal Dynamic Multi-hop Network (ST-DMN) to update the iterative traffic network graph by combining multi-hop operation with diffusion convolution technique. The novel graph generation technique and diffusion graph convolution can effectively capture the global dynamic spatial dependencies, but they lead to excessive parameters and increased training time. In addition, all these methods ignore the periodicity of traffic flow, which is a hidden temporal dependency determined by people's living habits.

Based on the above problems, we propose a new deep learning model MSSTGCN, which utilizes multi-head attention and spatial-temporal graph convolutional networks for multi-scale traffic flow prediction. By dividing traffic flow data into hourly, daily, and weekly scales, the periodic features in multiple time dimensions can be learned. A graph attention residual neural network layer is designed to capture global spatial dependencies and identify hidden traffic patterns in different regions. A

layer based on graph attention residual network and a T-GCN module is designed to extract the global spatial dependencies across regions and local spatial-temporal dependencies, respectively. A transformer layer that contains self-attention mechanism, positional encoder, layer normalization, and a fully connected output layer is proposed.

3 Methodology

The structure of the MSSTGCN model is shown in Fig. 2. It integrates the GARN, GCN, GRU, and multi-head self-attention mechanisms, each designed to extract global spatial characteristics, local spatial-temporal characteristics, and comprehensive temporal features, respectively.

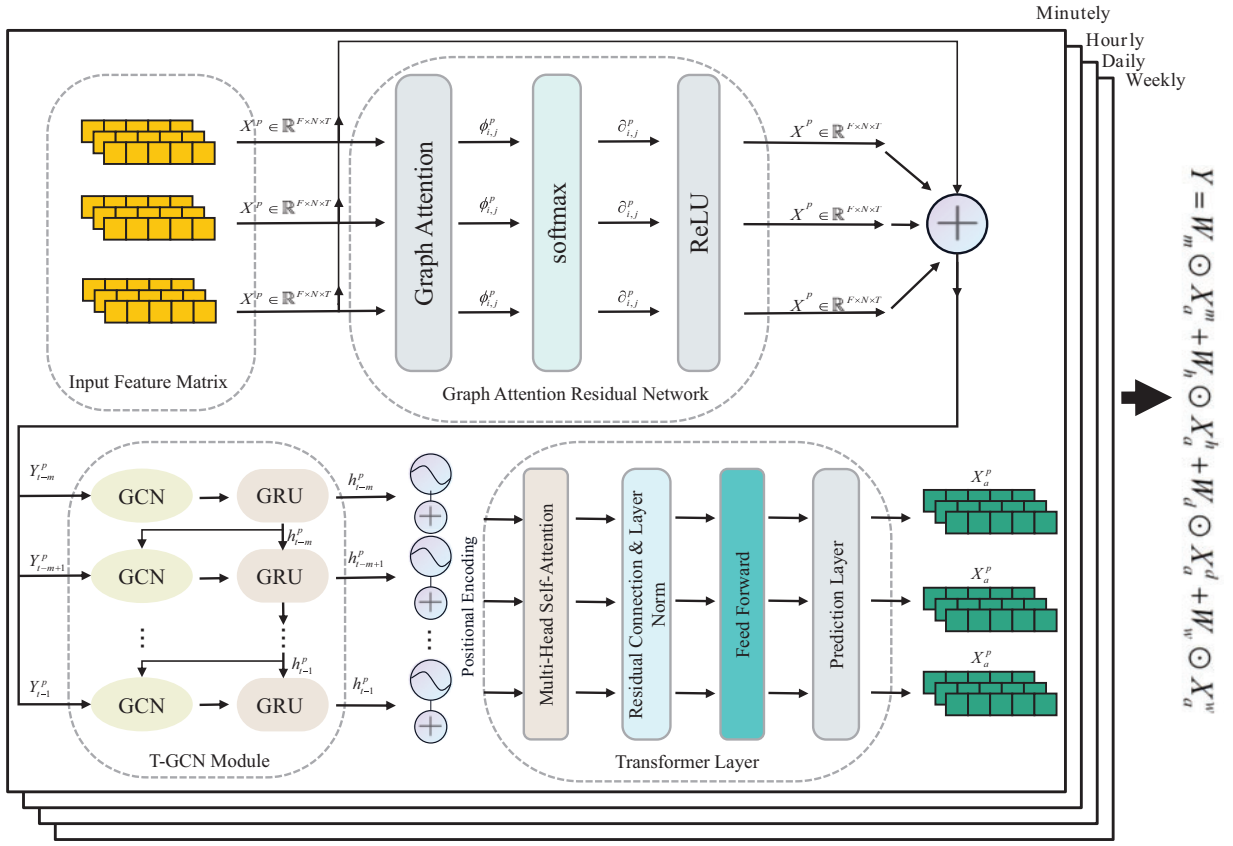


Figure 2: Overall structure of the MSSTGCN model

3.1 Preliminaries

Definition 1: Spatial graph structure: the topology of a transportation network is represented as an unweighted graph $G = (V, E, A)$ where $V = (v_1, v_2, \ldots, v_N)$ denotes the set of sensor nodes and N denotes the number of sensor nodes. E denotes the set of edges connecting the nodes. $A \in \mathbb{R}^{n \times n}$ is denoted as the adjacency matrix of the graph G.

Definition 2: Node attributes: the attribute features of a sensor node at moment t are represented as vector $x^t \in \mathbb{R}^F$, where F is the number of the sensor node. Denote $X_t = (X_1^t, X_2^t, \ldots, X_N^t) \in \mathbb{R}^{F \times N}$ as the feature matrix at time t.

Traffic prediction task [38] refers to predicting future traffic flow $X = (X_t, X_{t+1}, \ldots, X_{t+T})$ through the historical traffic flow $X = (X_{t-m-1}, X_{t-m}, \ldots, X_{t-1})$ and traffic spatial graph G. The learned mapping function can be expressed as Eq. (1).

$$f\left((X_{t-m-1}, X_{t-m}, \ldots, X_{t-1}), G\right) \rightarrow (X_t, X_{t+1}, \ldots, X_{t+T}) \tag{1}$$

3.2 Global Spatial Module

To investigate the underlying cyclical patterns in traffic flow, a multilevel temporal structure is proposed, where the time-difference period of the road nodes is set to hourly, daily, and weekly, i.e., $p \in hour, day, week$, respectively. In particular, the periodicity of the original traffic flow data is preserved and considered as an additional distinct scale cycle to learn the spatial-temporal features. The data input to the model can be denoted as $X^p = (X^p_{t-m-1}, X^p_{t-m}, \ldots, X^p_{t-1}) \in \mathbb{R}^{F \times N \times T}$.

The structure of the graph attention residual network layer is illustrated in Fig. 3. Initially, the historical feature vector $X^p \in \mathbb{R}^{F \times T^p}$ of each node at a specific resolution is mapped to a different feature space through a linear transformation to enhance the model's feature fitting capability. Any two nodes are connected for dot-product with the learnable weight vector $\eta^T \in \mathbb{R}^{2T^p}$. The attention coefficient $\phi_{i,j}$ between nodes is obtained after the activation function of $LeakReLU$. The correlation weights $\partial_{i,j}$ between nodes are further computed by the $Softmax$ function, which is expressed as shown in Eqs. (2)–(4):

$$\tilde{x}^p = x^p w^p \tag{2}$$

$$\phi^p_{i,j} = \text{LeakReLU}\left(\eta\left[\tilde{x}^p_i \| \tilde{x}^p_j\right]\right) \tag{3}$$

$$\partial_{i,j} = \text{Softmax}(phi^p_{i,j}) \tag{4}$$

where w^p represents the parameter weight matrix, $\|$ represents the vector-connected embedding.

The current node gets updated by evaluating its relationship with all other nodes through the relevance weights $\partial_{i,j}$. Expressing attention as a polytope form. The model integrates feature information from multiple subspaces to improve feature representation. In addition, the residual connection enables the model to integrate the underlying features with the high-dimensional features to improve the generalization ability of the model, and finally the feature semantic matrix containing the global spatial dependencies can be obtained as shown in Eqs. (5)–(6):

$$\widehat{X}^p = \|_{head=1}^{Head} \text{ReLU}\left(\sum_{(i,j)\in N(i,j)} \partial^p_{i,j} \tilde{x}^p_i\right) \tag{5}$$

$$Y^p = \widehat{X}^p + X^p \tag{6}$$

where ReLU is the activation function and $Head$ denotes the number of heads of the multi-head attention.

3.3 Local Spatial-Temporal Module

Local spatio-temporal dependencies are real-time and impactful because any unexpected traffic condition (e.g., car accident, etc.) can affect the traffic flow in a short period. We use T-GCN [14] as a module to capture local spatio-temporal dependencies.

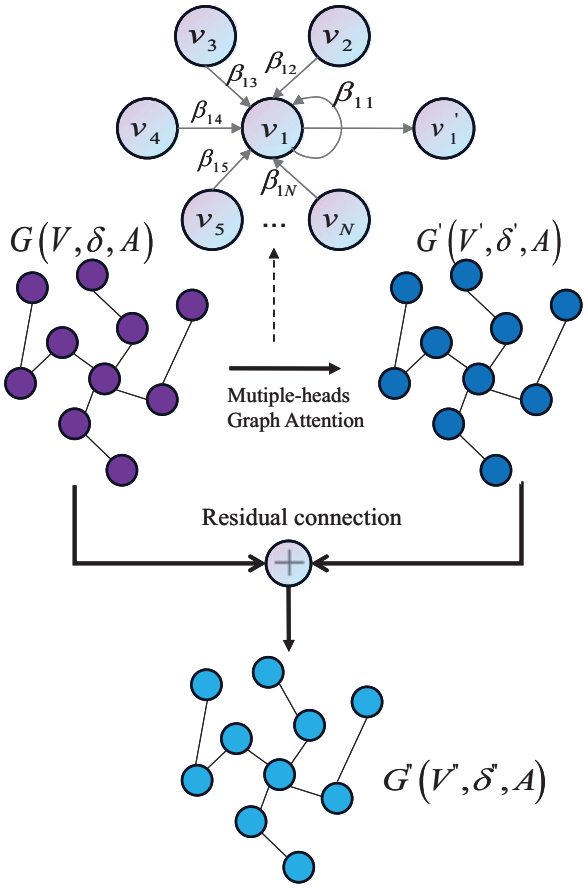


Figure 3: Graphical attention residual network layer structure

The feature normalized laplace matrix can be expressed as $A_{lap} = D^{-\frac{1}{2}}(A + I_N)D^{-\frac{1}{2}}$, where I_N is identity matrix, and D is degree matrix of $(A + I_N)$. The GCN acts directly on the neighboring roads of the traffic node to aggregate the local spatial information of traffic, which is expressed as shown in Eq. (7):

$$GC(X) = \sigma(A_{lap}XW_{GC}) \tag{7}$$

where $\sigma(.)$ is the sigmoid activation function, and W_{GC} is the learnable matrix.

GRU, as a commonly used time series data prediction model, solves the gradient vanishing and gradient explosion problems of RNN to a certain extent. GRU controls the amount of information to be retained from the historical information of the previous hidden state and the current state through the reset gate r_t^p and the update gate u_t^p. The GCN is embedded into the linear transformation of GRU to capture the local spatio-temporal dependencies within the current time step. The GCN is embedded into the linear transformation of the GRU to capture the local spatio-temporal dependencies at the current time step, taking values from 0 to 1, and the closer the value to 1, the greater the degree of retention. The recursive layer within the t time step can be represented as Eqs. (8)–(11):

$$u_t^p = \sigma(GC(Y_t^p, h_{t-1}^p) + b_u) \tag{8}$$

$$r_t^p = \sigma(GC(Y_t^p, h_{t-1}^p) + b_r) \tag{9}$$

$$c_t^p = \tanh(GC(Y_t^p, r_t^p * h_{t-1}^p) + b_c) \tag{10}$$

$$h_t = u_t^p * h_{t-1}^p + (1 - u_t^p) * c_t^p \tag{11}$$

where tanh represent activation function, and b_u, b_r, b_c is the biases. $GC(.)$ represents the graph convolution defined by Eq. (7).

3.4 Global Temporal Module

The final output $H^p = (h_{t-m-1}^p, h_{t-m}^p, \ldots, h_{t-1}^p)$ of T-GCN contains global spatial information as well as local spatio-temporal information. However, the long loop structure of GRU makes it impossible to effectively capture long-term temporal dependencies when the traffic flow data is too long-spaced. For this reason, a transformer layer is proposed to solve this problem, and the structure is shown in Fig. 4. First, the traffic flow data are tagged with sequential information through positional encoding. Specifically, each temporal position po is encoded as a vector of length d_{model}. The traffic flow data is added and fused with the traffic flow data to obtain a vector with length d. Fusion with the traffic flow data results in a feature matrix with sequential features $\widehat{H}^p$. Its representation is shown in Eqs. (12)–(13):

$$\eta_{po(i)} = \begin{cases} \sin\left(\mathrm{po}/10000^{i/d_{\mathrm{model}}}\right), \text{if i is even,} \\ \cos\left(\mathrm{po}/10000^{i-I/d_{\mathrm{model}}}\right), \text{ if i is odd.} \end{cases} \tag{12}$$

$$\widehat{H}^p = H^p + \eta_{po(i)} \tag{13}$$

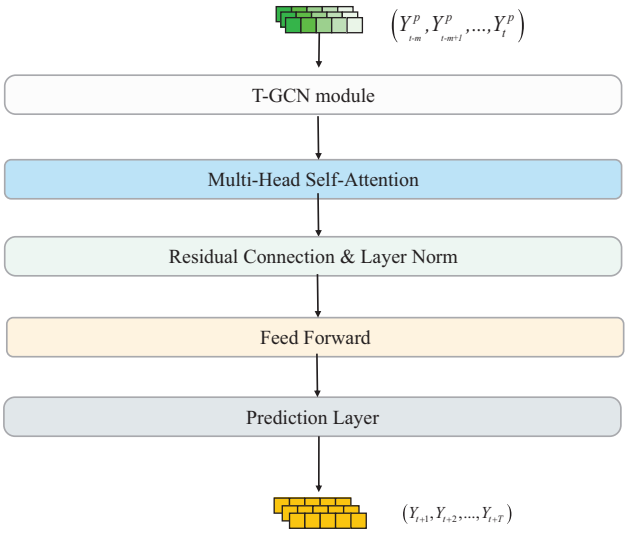


Figure 4: Schematic diagram of transformer layer structure

In this paper, d_{model} denotes the feature size of the transformer layer, *sin* and *cos* are trigonometric functions.

To better capture the long-term time dependence of traffic data, a multi-head self-attention mechanism is used to adaptively assign weights to the historical time steps of the feature matrix. Specifically, the spatio-temporal feature matrix $\widehat{H}^p$ is mapped to three feature spaces denoted as series, keys, and values, and the dot product of series and keys is used to compute the attention coefficient. This is then divided by a specific value $d_k = d_{model}/Head$ to stabilize the gradient training. Softmax function will output the attention score [33]. The final output is a linear aggregation of all *Head* spaces. The expressions are shown in Eqs. (14)–(16):

$$Q = \widehat{H}^p W_q, K = \widehat{H}^p W_k, V = \widehat{H}^p W_v \tag{14}$$

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{15}$$

$$\widehat{H}^p_{attention} = \|_{head=1}^{Head} \text{Attention}(Q, K, V) \tag{16}$$

where $W_q \in \mathbb{R}^{d_{model} \times d_k}$, $W_k \in \mathbb{R}^{d_{model} \times d_k}$ and $W_v \in \mathbb{R}^{d_{model} \times d_k}$ are the learnable parameters.

The layer normalization is further used to stabilize the training of the neural network and interact with the feature information of different dimensions through residual connection. Similarly, the feed-forward layer is used to enhance the nonlinear fitting ability of the model to the traffic data as shown in Eqs. (17)–(18):

$$\widehat{H}^p_a = \text{LayerNorm}(\widehat{H}^p_{attention} + H^p) \tag{17}$$

$$H^p_a = W_2 \,\text{ReLU}(W_1 \widehat{H}^p_a + b_1) + b_2 \tag{18}$$

where *LayerNorm* represents layer normalization and W_1 and W_2 are learnable parameters, b_1 and b_2 are the biases.

Ultimately, the output of the proposed model is a linear aggregation containing four structurally identical multiscale components, i.e., original (X^o_a) hourly (X^h_a), daily (X^d_a) and weekly (X^w_a) The computational procedure is shown in Eq. (19):

$$Y = W_o \odot X^o_a + W_h \odot X^h_a + W_d \odot X^d_a + W_w \odot X^w_a \tag{19}$$

where W_o, W_h, W_d and W_w represent learnable matrix, respectively.

Self-attention in the Transformer layer attends to all-time nodes in a parallelized manner to capture long-term temporal dependencies. The training objective of the model is to minimize the difference between the real traffic data x and the predicted data y. The loss function is shown in Eq. (20), where λ and W represent penalty terms and weight parameters, respectively.

$$L = \frac{1}{2}\sum(y - \tilde{y})^2 + \lambda \| W \| \tag{20}$$

4 Experiments

Experiments are carried out on four real-world traffic datasets, including Los-loop, SZ-taxi, METR-LA, and PEMS-BAY, to evaluate the prediction performance of the MSSTGCN model. The Los-loop dataset contains the data from 207 freeway sensors in Los Angeles County for the period of 01 March to 07 March 2012. The SZ-taxi dataset was derived from taxi data in Shenzhen City from

01 January to 31 January 2015. The METR-LA dataset is the traffic data of Los Angeles freeways from 01 March to 30 June 2012. The PEMS-BAY dataset developed by the California Transportation Agencies Performance Measurement System includes the data of 325 sensors in total from 01 January 2017, to 31 May 2017.

All of the data feature vehicle speeds on the road, and all contain a matrix of collected traffic flow features as well as a road network adjacency matrix. It is worth noting that all the datasets collect traffic flow data every five minutes, except for the SZ-taxi dataset, which collects traffic flow data every fifteen minutes. The details of the four datasets are shown in Table 1.

Table 1: Datasets

Datasets	Detectors	Time steps	Interval
Los-loop	207	2016	5 min
SZ-taxi	156	2976	15 min
METR-LA	207	34,272	5 min
PEMS-BAY	325	52,116	5 min

4.1 Evaluation Metric

We use six metrics to test the difference between model predictions and real data to evaluate model prediction performance. RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error) are employed to quantify the prediction error magnitude, where smaller values indicate lower prediction errors in the model. Accuracy measures the precision of model predictions. Additionally, R^2 and Var represents the fitting degrees between the predicted and true values. The higher values of the three metrics mean the performance is better. All evaluation indexes can be calculated by Eqs. (21)–(26).

$$\mathrm{RMSE}(y,\tilde{y}) = \sqrt{\frac{1}{N_{pv}}\sum_{i=1}^{N_{pv}}\left(y_i-\tilde{y}_i\right)} \tag{21}$$

$$\mathrm{MAE}(y,\tilde{y}) = \frac{1}{N_{pv}}\sum_{i=1}^{N_{pv}}\left|y_i-\tilde{y}_i\right| \tag{22}$$

$$\mathrm{Accuracy}(y,\tilde{y}) = 1-\frac{\|y-\tilde{y}\|_F}{\|y\|_F} \tag{23}$$

$$R^2(y,\tilde{y}) = 1-\frac{\sum_{i=1}\left(y_i-\tilde{y}_i\right)^2}{\sum_{i=1}\left(y_i-\bar{y}\right)^2} \tag{24}$$

$$\mathrm{Var}(y,\tilde{y}) = 1-\frac{Var(y-\tilde{y})}{Var(y)} \tag{25}$$

$$\text{MAPE}(y, \tilde{y}) = \frac{1}{N_{pv}} \sum_{i=1}^{N_{pv}} \left| \frac{y_i - \tilde{y}_i}{y_i} \right| \times 100\% \tag{26}$$

where N_{pv} represents the number of predicted traffic data. RMSE is the root mean square error, which measures the standard deviation of the prediction error or the distribution of the error, which details the degree of concentration of the predicted data along the line of best fit. MAE is calculated by the mean sum of the absolute error which is the absolute difference between the predicted value and actual value. Accuracy describes how close the predicted values are to the actual values, perfect accuracy will result in a score of 1. The R^2 is used to measure the explanatory degree of the independent variable on the variation of the dependent variable. It measures the ability of the model to correctly predict the new data and its best value is 1. VAR is the proportion of the model's variance that is affected by the actual factors in the data for the explained variance metric. MAPE is the mean absolute percentage error.

4.2 Experimental Settings and Baseline Methods

For deep learning models, different hyperparameters will affect the prediction results. In this paper, the hyperparameter settings are as follows. The learning rate is 0.001, and the number of training epochs is 3000. The batch size values for the SZ-taxi dataset and other datasets are 64 and 16, respectively. For the METR-LA and PEMS-BAY datasets, the feature size of the transformer layer, the number of hidden units, and the feature size of GRU are set to 64.

80% of all datasets are used for training, while the remaining 20% are reserved as test data. To expedite model convergence during training, we employ the maximum normalization method to scale the data within the range [0, 1], expressed as. All models were trained on a computer with an NVIDIA 3060 GPU using the Pytorch framework.

For the Los-loop and SZ-taxi datasets, methods including Historical Average (HA) [20], AutoRegressive Integrated Moving Average (ARIMA) [21], Support Vector Regression (SVR) [23], Gated Recurrent Unit (GRU) [10], Temporal Graph Convolutional Network (T-GCN) [14], Attention Temporal Graph Convolutional Network (A3T-GCN) [34], and Diffusion Convolutional Recurrent Neural Network (DCRNN) [15] are compared. For the METR-LA and PEMS-BAY datasets, Spatio-Temporal Graph Convolutional Networks (STGCN) [39], DCRNN [15], Graph Multi-Attention Network (GMAN) [40], Fully Connected gated graph architecture (FC-GAGA) [41], Spatio-Temporal data using deep Meta learning Network model (ST-MetaNet) [42], Graph Wavenet [43], Multivariate Time series forecasting with Graph Neural Networks (MTGNN) [44], Mixed Hop diffuse Ordinary Differential Equation (MHODE) [45], Multi-Head SpatioTemporal Attention Graph Convolutional Network (MHSA-GCN) [35], Multi-scale Temporal and Enhance Spatial transformer (MTESformer) [36], and SpatioTemporal Dynamic Multi-Hop network (ST-DMN) [37] are compared.

4.3 Result Analysis

Table 2 shows the prediction results on the Los-loop and SZ-taxi datasets, with the prediction time steps of 15, 30, and 60 min, respectively. The MAPE cannot be computed in the SZ-taxi dataset because there is a large amount of noisy data with the value close to 0. The other evaluation metrics are adequate to substantiate the predictive capability of the baseline method.

Table 2: The prediction performance of different model on the Los-loop and SZ-taxi dataset

Time	Models	Los-loop				SZ-taxi			
		RMSE	MAE	Acc	MAPE	RMSE	MAE	Acc	MAPE
15 min	HA	7.31	3.88	0.88	0.10	4.23	2.78	0.70	*
	ARIMA	10.08	7.70	0.83	0.21	6.80	4.68	0.38	*
	SVR	6.70	3.54	0.89	0.11	4.17	2.78	0.71	*
	GRU	5.13	3.02	0.91	0.08	4.16	2.71	0.71	*
	T-GCN	5.11	3.22	0.91	0.08	4.09	2.79	0.72	*
	DCRNN	5.10	2.84	0.91	0.07	4.23	3.09	0.71	*
	A3T-GCN	5.03	3.21	0.91	0.08	4.08	2.75	0.72	*
	MHSA-GCN	4.88	3.01	0.92	0.07	3.80	2.66	0.73	*
	MSSTGCN	**2.16**	**1.09**	**0.96**	**0.03**	**1.30**	**0.65**	**0.91**	*
30 min	HA	7.31	3.88	0.88	0.10	4.23	2.78	0.70	*
	ARIMA	10.08	7.70	0.83	0.21	6.80	4.68	0.38	*
	SVR	7.47	3.92	0.87	0.12	4.21	2.78	0.71	*
	GRU	6.36	3.75	0.89	0.10	4.12	2.84	0.71	*
	T-GCN	5.95	3.81	0.90	0.11	4.11	2.79	0.71	*
	DCRNN	6.20	3.27	0.89	0.09	4.13	2.62	0.71	*
	A3T-GCN	5.95	3.69	0.90	0.10	4.09	2.77	0.72	*
	MHSA-GCN	5.67	3.48	0.90	0.08	3.88	2.71	0.74	*
	MSSTGCN	**3.91**	**2.20**	**0.93**	**0.06**	**3.57**	**2.21**	**0.75**	*
60 min	HA	7.31	3.88	0.88	0.10	4.23	2.78	0.70	*
	ARIMA	10.08	7.70	0.83	0.21	6.80	4.68	0.38	*
	SVR	8.69	4.57	0.85	0.15	4.27	0.86	0.70	*
	GRU	7.80	4.68	0.86	0.14	4.16	2.87	0.71	*
	T-GCN	6.98	4.65	0.88	0.13	4.12	2.80	0.71	*
	DCRNN	7.60	3.87	0.87	0.11	4.40	3.27	0.69	*
	A3T-GCN	6.94	4.37	0.88	0.12	4.11	2.72	0.72	*
	MHSA-GCN	7.01	4.21	0.88	0.10	3.97	2.74	0.73	*
	MSSTGCN	**5.71**	**3.56**	**0.90**	**0.09**	**3.91**	**2.60**	**0.73**	*

Table 2 illustrates that traditional HA, ARIMA, and SVR methods struggle to accurately model the complex nonlinear traffic data, showing inferior prediction performance compared to deep learning-based methods. In the 15-min prediction of the two datasets, the prediction performance of GRU is better than that of the traditional methods, and the RMSE is reduced by about 23.48% and 0.08%, respectively, compared with SVR. However, GRU solely captures the temporal dependencies in traffic data while overlooking spatial dependencies, thereby limiting its prediction performance. T-GCN and DCRNN integrate GRU with GCN to capture both temporal and spatial dependencies in traffic data. In the 30-min prediction task on the Los-loop dataset, RMSE values have been reduced by about 6.42% and 2.49%, respectively, compared to GRU alone. A3T-GCN utilizes the attention

mechanism to capture the long-term temporal dependence and achieves better prediction performance. However, it neglects the fact that spatial dependence should also be global, and the attention mechanism uniformly treats all temporal nodes. The MHSTA-GCN combines graph convolutional network and multi-head attention module, with nodes in the network acting as feature representations of road traffic speeds. GCN is used to capture spatial correlations of the graph, and the multi-head attention models global temporal correlations. However, similar to A3T-GCN, global spatial dependencies are ignored. In the 15-min and 60-min predictions on the two datasets, MSSTGCN reduces the RMSE by about 55.74%, 65.79%, 18.54%, and 1.51%, respectively, and improves the accuracy by about 5.05%, 21.29%, 2.27%, and 0.68%, respectively. MSSTGCN captures the spatial-temporal dependence from the local to the global level and captures the temporal information of the traffic more comprehensively. Furthermore, it considers the implicit traffic periodicity and obtains richer feature information. In both datasets, MSSTGCN exhibits the best prediction performance among all baseline methods.

The prediction results on the METR-LA and PEMS-BAY datasets are shown in Table 3. Combining graph convolution and gated spatial-temporal convolution, STGCN applies multiple spatial-temporal convolutional blocks. However, these convolutional operations are mainly based on the domain information and ignore the global context. GMAN generates future feature representations by adding an attention layer to the encoder and decoder structure, with a multi-headed attention mechanism that captures global spatial-temporal correlations and outputs the representations through a gated fusion mechanism. Compared with the 15-min prediction results of STGCN and GMAN, RMSE values on the two datasets are reduced by about 3.31% and 1.01%, and the MAPEs are reduced by about 2.49% and 0.69%, respectively. MTGNN learns an adaptive adjacency matrix through the traffic sequences, which are coupled with a spatial-temporal convolutional module. Compared with the 30-min prediction of GMAN, the RMSE of MTGNN is reduced by 4.49% and 1.06%, respectively. MHODE utilizes a gated spatial-temporal convolution network and a hybrid jump-diffusion ordinary differential equation to capture long-term temporal dependence. MTESformer develops a multi-scale temporal transformer. It focuses on temporal correlations in time series data and combines the self-attention mechanism with the multi-scale convolutional unit to identify different traffic flow patterns and capture long-term dependencies. However, MTESformer does not perform well in short-term prediction since the short-term temporal correlations are ignored. ST-DMN captures long-range spatial dependencies by combining multi-hop operations with diffusion convolution techniques, but it fails to take into account the periodic changes in traffic flow. Compared with MTGNN, MTESforme reduces the MAE of the 60-min prediction by 3.44% and 3.61%, respectively. MSSTGCN learns the hidden traffic periodicity through a multi-scale form and outperforms the baseline methods in most time steps.

Table 3: The prediction performance of different model on the METR-LA and PEMS-BAY dataset

Time	Models	METR-LA			PEMS-BAY		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE
	STGCN	5.74	2.88	7.62%	2.96	1.36	2.90%
	DCRNN	5.38	2.77	7.30%	2.95	1.38	2.90%
	GMAN	5.55	2.81	7.43%	2.93	1.36	2.88%
15 min	FC-GAGA	5.34	2.75	7.25%	2.86	1.36	2.87%

(Continued)

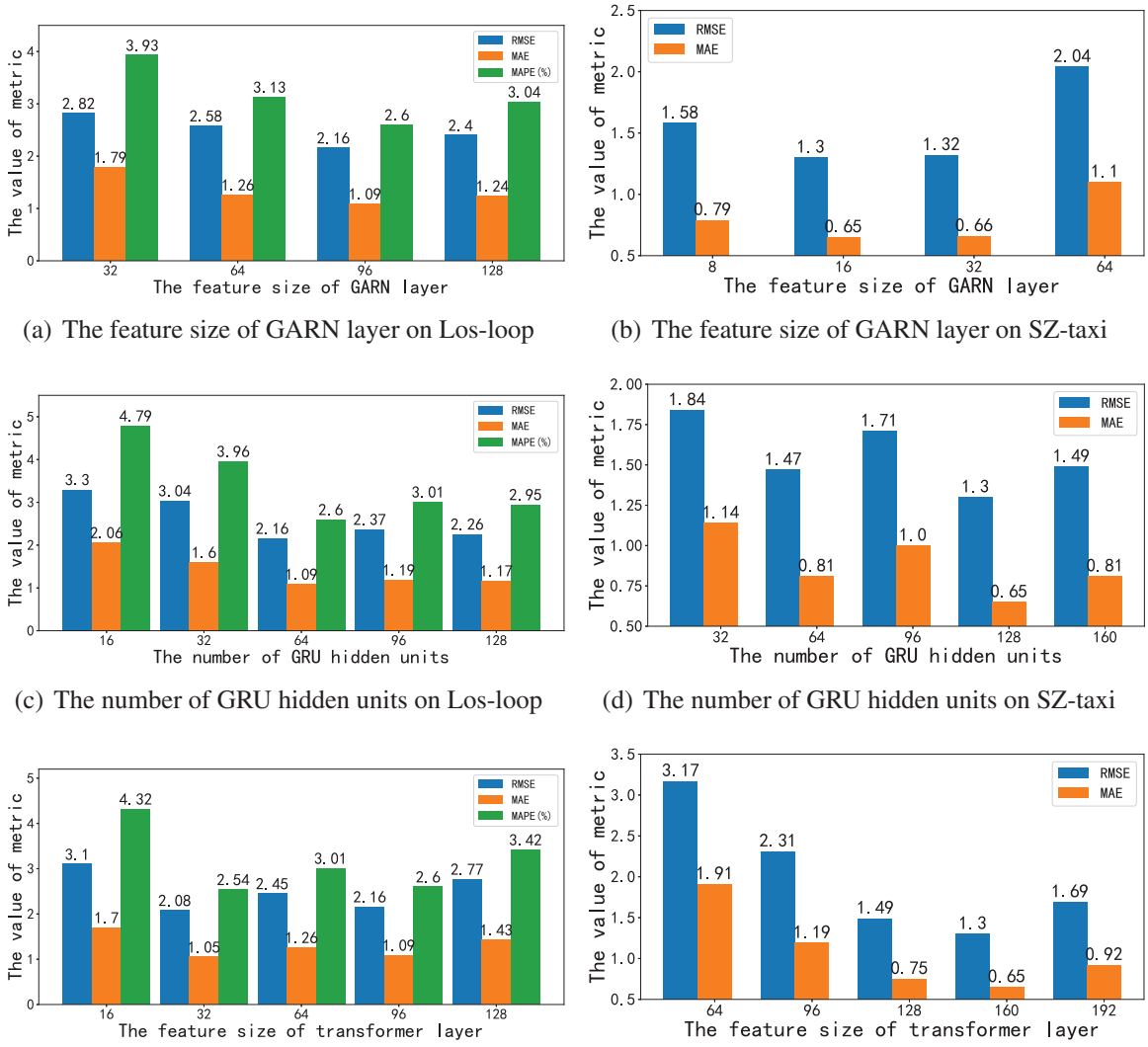
Table 3 (continued)

Time	Models	METR-LA			PEMS-BAY		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE
	ST-MetaNet	5.17	2.69	6.91%	2.90	1.36	2.82%
	Graph wavenet	5.15	2.69	6.90%	2.74	1.30	2.73%
	MTGNN	5.18	2.69	6.86%	2.79	1.32	2.77%
	MHODE	5.17	2.69	6.88%	2.72	1.30	2.67%
	MTESformer	5.34	2.73	7.10%	2.80	1.31	2.75%
	ST-DMN	5.09	2.63	6.65%	2.74	1.30	2.73%
	MSSTGCN	**2.83**	**1.36**	**3.01%**	**1.36**	**0.89**	**1.63%**
30 min	STGCN	7.24	3.47	9.57%	4.27	1.81	4.17%
	DCRNN	6.45	3.15	8.80%	3.97	1.74	3.90%
	GMAN	6.46	3.12	8.35%	3.78	1.64	3.71%
	FC-GAGA	6.30	3.10	8.57%	3.80	1.68	3.80%
	ST-MetaNet	6.28	3.10	8.57%	4.02	1.76	4.00%
	Graph wavenet	6.22	3.07	8.37%	3.70	1.63	3.67%
	MTGNN	6.17	3.05	8.19%	3.74	1.65	3.69%
	MHODE	6.15	3.04	8.23%	3.62	1.61	3.55%
	MTESformer	6.22	3.03	8.30%	3.73	1.62	3.62%
	ST-DMN	6.15	3.01	8.08%	3.73	1.62	3.67%
	MSSTGCN	**4.37**	**2.12**	**6.91%**	**1.92**	**0.97**	**2.35%**
60 min	STGCN	9.40	4.59	12.70%	5.69	2.49	5.79%
	DCRNN	7.60	3.60	10.50%	4.74	2.07	4.90%
	GMAN	7.37	3.46	10.06%	4.40	1.90	4.45%
	FC-GAGA	7.31	3.51	10.14%	4.52	1.97	4.67%
	ST-MetaNet	7.52	3.59	10.63%	5.06	2.20	5.45%
	Graph wavenet	7.37	3.53	10.01%	4.52	1.95	4.63%
	MTGNN	7.23	3.49	9.87%	4.49	1.94	4.53%
	MHODE	7.21	3.47	9.77%	4.49	1.90	4.34%
	MTESformer	**7.14**	**3.37**	**9.62%**	4.38	**1.87**	**4.35%**
	ST-DMN	7.32	3.45	9.91%	4.46	1.89	4.51%
	MSSTGCN	7.16	4.18	10.34%	**4.02**	2.24	4.59%

4.4 Hyperparameters Analysis

The significance of various hyperparameters on the predictive performance of the model is substantial. Three key hyperparameters affect the learning ability of MSSTGCN to capture spatial-temporal correlations. Fig. 5 depicts how RMSE, MAE, and MAPE vary across Los-loop and SZ-taxi datasets under different hyperparameter configurations. In the Los-loop dataset, the model achieves optimal results with minimal prediction error when setting the feature size of the GARN layer to 96, the number of GRU hidden units to 64, and the feature size of the transformer layer to 32. For the SZ-taxi dataset, the model performance is best when these three values are set to 16, 128, and 160, respectively. Moreover, adjusting hyperparameter values either downward or upward can constrain the model's performance or heighten its complexity, thereby diminishing predictive accuracy. When the values of

these hyperparameters are smaller, the model cannot capture more hidden correlations and cannot achieve the expected effects. However, this does not mean that larger values of these parameters are better, as the hidden correlations between traffic flows are limited. External factors such as weather and holidays that affect traffic flow can be treated as learnable hidden features. However, when the hyperparameters are too large, they may dilute the learning of these correlations, resulting in poor predictive performance.



(a) The feature size of GARN layer on Los-loop

(b) The feature size of GARN layer on SZ-taxi

(c) The number of GRU hidden units on Los-loop

(d) The number of GRU hidden units on SZ-taxi

(e) The feature size of transformer layer on Los-loop

(f) The feature size of transformer layer on SZ-taxi

Figure 5: The impact of different hyperparameters on prediction performance on the Los-loop and SZ-taxi datasets

4.5 Ablation Study

To assess the influence of each component of MSSTGCN on prediction performance, five variants are tested on the Los-loop and SZ-taxi datasets for ablation study. (1) w/o GS: Removes the GARN layer in MSSTGCN and fails to capture the global spatial dependence. (2) w/o LST: Removes the

T-GCN module preventing MSSTGCN from capturing local spatial-temporal dependencies. (3) w/o GT: The transformer layer responsible for capturing long-term temporal dependencies is removed. (4) w/o PE: In this variant, the positional encoding in the transformer layer is removed and the self-attention mechanism is unable to recognize sequential features. (5) w/o PB: The multi-scale module in MSSTGCN is removed, and the hidden traffic data periodicity cannot be learned.

The experimental results are shown in Table 4. It indicates that the MSSTGCN model has the best performance. Therefore, it is crucial to consider the local-to-global spatial-temporal dependencies and the hidden periodicity in traffic prediction tasks. The GARN layer of the w/o GS variant is designed to capture global spatial correlation, and the removal of the GARN layer reduces the prediction effect considerably, which suggests that not only local spatial correlation but also global implicit spatial correlation should be taken into account in the traffic flow. The w/o LST module has less impact on the model's overall performance because it only removes the T-GCN module which is mainly concerned with local temporal correlation. It is shown that compared to the MSSTGCN model, the RMSE and MAE metrics of the w/o LST variant for 30-min and 60-min predictions have increased by 11.4%, 21.9%, 3.7%, and 3.8% on the Los-loop dataset, and 9.6%, 22.5%, 2.1%, and 1.7% on the SZ-taxi dataset. However, the RMSE and MAE metrics for 15-min prediction are quite different from those of MSSTGCN. They have increased by 40.7% and 53.9% on the Los-loop dataset, and 51.6% and 60.4% on the SZ-taxi dataset, which indicates that T-GCN is more effective in modeling local temporal correlations. Among all variants, w/o GT (without transformer layer) has the greatest impact. In the w/o GT variant, the transformer layer is used to focus on global temporal correlations. This indicates that it is necessary to consider the global temporal correlations in traffic flow. The w/o PE variant removes positional embedding. As shown in Table 4, the w/o PE variant has the smallest impact. The results of the w/o PE variant are closer to those of MSSTGCN for 30-min and 60-min prediction because multi-head attention can effectively capture long-term temporal dependency in long-term prediction even without positional encoding. However, in short-term prediction, positional encoding is still necessary while focusing on the temporal correlations. The w/o PB variant (without multi-scale module) is unable to learn the hidden periodicity of traffic data, which leads to a decrease in prediction effects, especially in 30-min and 60-min predictions. It can be seen from Table 4 that w/o GT has the worst RMSE metric for 15-min prediction, but for 30-min and 60-min predictions, the RMSEs of w/o PB are the worst. It indicates that the periodic features in traffic flow can not be neglected.

Table 4: Results of ablation study

Time	Models	Los-loop			SZ-taxi		
		RMSE	MAE	Acc	RMSE	MAE	Acc
15 min	w/o GS	3.8702	2.2495	0.9335	2.1439	1.4201	0.8505
	w/o LST	3.6489	2.3667	0.9374	2.6827	1.6417	0.8130
	w/o GT	4.7246	2.6535	0.9287	3.8549	2.5526	0.7373
	w/o PE	3.2066	2.0864	0.9452	1.6548	1.0631	0.8846
	w/o PB	3.5384	2.3052	0.9394	2.9412	1.9896	0.7949
	MSSTGCN	**2.1639**	**1.0894**	**0.9621**	**1.2993**	**0.6507**	**0.9094**
30 min	w/o GS	4.8201	2.8582	0.9171	3.7694	2.5592	0.7348
	w/o LST	4.4182	2.8233	0.9241	3.9076	2.7041	0.7173
	w/o GT	5.6756	3.0424	0.9188	3.9167	2.5232	0.7330

(Continued)

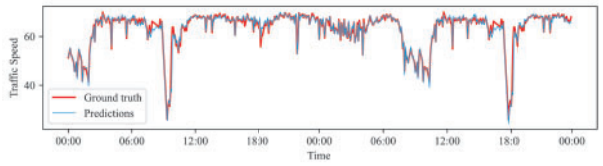
Table 4 (continued)

Time	Models	Los-loop			SZ-taxi		
		RMSE	MAE	Acc	RMSE	MAE	Acc
	w/o PE	4.0867	2.6296	0.9300	3.5735	2.3970	0.7504
	w/o PB	4.5248	2.8730	0.9222	4.0193	2.8928	0.7106
	MSSTGCN	**3.9125**	**2.2044**	**0.9311**	**3.5652**	**2.2074**	**0.7513**
60 min	w/o GS	6.2143	4.1995	0.8929	3.9102	2.6040	0.7326
	w/o LST	5.9255	3.7061	0.8978	3.9877	2.6506	0.7301
	w/o GT	6.4056	3.7913	0.8917	4.0130	2.7017	0.7275
	w/o PE	**5.6259**	3.6492	**0.9031**	**3.8855**	**2.5758**	**0.7390**
	w/o PB	5.7321	3.7261	0.9003	4.0767	2.6809	0.7226
	MSSTGCN	5.7089	**3.5639**	0.8996	3.9052	2.6041	0.7323

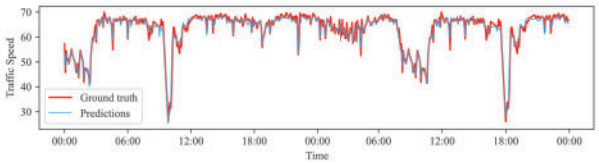
4.6 Analysis of Visualization

Two roads of the Los-loop and SZ-taxi datasets are selected to visually demonstrate the model's prediction performance.

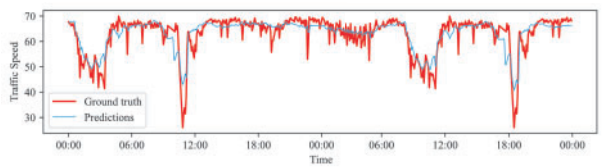
We randomly selected two roads from each dataset and visualized the predicted values at various time steps alongside the original values. Fig. 6a–c shows the visualized prediction results on the Los-loop dataset for the whole day from 06 March to 07 March 2012. Fig. 6d–f depicts the visualization of the prediction results on the SZ-taxi dataset from 26 January to 30 January 2015. MSSTGCN demonstrates enhanced accuracy in predicting changes in traffic flow within a short prediction step of 15 min. As the prediction step increases, the results become smoother and smoother. The trends of all prediction curves are consistent with the original curves, which proves the robustness and effectiveness of MSSTGCN performance. In visualizing the 15-min prediction results for the Los-loop dataset, MSSTGCN demonstrates robust adaptation to sudden changes in traffic data and achieves accurate predictions. This shows that MSSTGCN has the potential to predict unexpected situations such as automobile accidents.



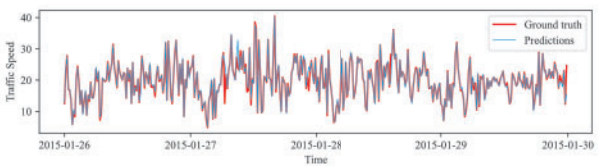
(a) 15 min of Los-loop



(b) 30 min of Los-loop

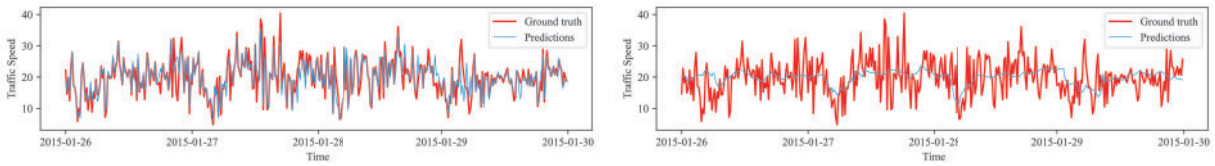


(c) 60 min of Los-loop



(d) 15 min of SZ-taxi

Figure 6: (Continued)



(e) 30 min of SZ-taxi (f) 60 min of SZ-taxi

Figure 6: Visualization of MSSTGCN results on Los-loop and SZ-taxi datasets with different prediction time steps

5 Conclusion

In this paper, a spatial-temporal graph convolutional traffic flow prediction model for multi-scale prediction is presented. The traffic data is encoded into a multi-scale form to explore the hidden periodicity. The T-GCN module can capture the local spatial-temporal dependencies. The combination of the graph attention residual network and a transformation layer can capture the global spatial-temporal dependencies. Extensive experiments conducted on four real traffic datasets demonstrate that our model outperforms baseline methods. Ablation experiments further verify the effectiveness of each module. Visualization analysis shows that the model predicts results similar to actual traffic flows and is able to predict roadway emergencies. MSSTGCN performs well in short-term traffic flow prediction and is able to effectively respond to sudden changes in traffic flow, such as accidents, severe weather conditions, and other unforeseen events. This adaptability of MSSTGCN shows a wide range of applications and potential values. In urban traffic management, MSSTGCN can be used to monitor and predict traffic flow in real time, thus helping traffic managers adjust signal timing and traffic flow in time to reduce congestion. In addition, by using the model's prediction results, reasonable navigation suggestions can be provided to drivers to avoid accident-prone areas and improve driving safety. In Intelligent Transportation Systems (ITS), MSSTGCN can also be used in conjunction with other sensors and data sources, such as social media data and weather forecasting data, to further enhance the accuracy of prediction. This enables transportation planners to make more scientific decisions when designing infrastructure and optimizing transportation networks. In conclusion, the forecasting capability of MSSTGCN can not only enhance the efficiency of traffic management but also provide important support for the development of smart cities and drive the realization of safer and more efficient transportation systems. In future research, traffic external factors (e.g., weather, number of vehicles, holidays, etc.) that significantly influence the prediction results should be considered and incorporated into the model to better reflect real-world scenarios and achieve more accurate results.

Acknowledgement: This work was supported by the National Natural Science Foundation of China.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Grant Nos. 62472149, 62376089, 62202147), Hubei Provincial Science and Technology Plan Project (2023BCB04100).

Author Contributions: Xinlu Zong conceptualized the research design, Fan Yu designed the methodology, Zhen Chen took charge of data compilation, Xue Xia verified the paper. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: A publicly available Los-loop, SZ-taxi, METR-LA, and PEMS-BAY datasets were used for analyzing our model. These datasets can be found at https://github.com/lehaifeng/T-GCN and https://github.com/liyaguang/DCRNN (accessed on 06 November 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] W. Zhang, S. Yan, and J. Li, "TCP-BAST: A novel approach to traffic congestion prediction with bilateral alternation on spatiality and temporality," *Inf. Sci.*, vol. 608, no. C, pp. 718–733, Aug. 2022. doi: 10.1016/j.ins.2022.06.080.

[2] H. Li, Y. Chen, K. Li, C. Wang, and B. Chen, "Transportation internet: A sustainable solution for intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15818–15829, Dec. 2023. doi: 10.1109/TITS.2023.3270749.

[3] Z. Li, J. Zhou, Z. Lin, and T. Zhou, "Dynamic spatial aware graph transformer for spatiotemporal traffic flow forecasting," *Knowl. Based Syst.*, vol. 297, no. 11, May 2024, Art. no. 111946. doi: 10.1016/j.knosys.2024.111946.

[4] X. Feng, Y. Chen, H. Li, T. Ma, and Y. Ren, "Gated recurrent graph convolutional attention network for traffic flow prediction," *Sustainability*, vol. 15, no. 9, May 2023, Art. no. 7696. doi: 10.3390/su15097696.

[5] X. Qi, G. Mei, J. Tu, N. Xi, and F. Piccialli, "A deep learning approach for long-term traffic flow prediction with multifactor fusion using spatiotemporal graph convolutional network," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8687–8700, 2023. doi: 10.1109/TITS.2022.3201879.

[6] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *Eur. Transp. Res. Rev.*, vol. 7, no. 3, Jun. 2015, Art. no. 21. doi: 10.1007/s12544-015-0170-8.

[7] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. Part C Emerg. Technol.*, vol. 62, no. 5, pp. 21–34, 2016. doi: 10.1016/j.trc.2015.11.002.

[8] C. Shuai, W. Wang, G. Xu, M. He, and J. Lee, "Short-term traffic flow prediction of expressway considering spatial influences," *J. Transp. Eng. A-Syst.*, vol. 148, no. 6, 2022, Art. no. 62. doi: 10.1061/JTEPBS.0000660.

[9] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *2016 31st Youth Acad. Annu. Conf. Chinese Assoc. Autom. (YAC)*, 2016, pp. 324–328.

[10] K. Cho *et al.*, "Learning phrase representations using RNN rncoder-decoder for statistical machine translation," in *Proc. 2014 Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[11] D. Wang, H. Yang, and H. Zhou, "Dynamic spatial-temporal self-attention network for traffic flow prediction," *Future Internet*, vol. 16, no. 6, May 2024. doi: 10.3390/fi16060189.

[12] S. He *et al.*, "STGC-GNNs: A GNN-based traffic prediction framework with a spatial-temporal granger causality graph," *Physica A*, vol. 623, no. 6, 2023. doi: 10.1016/j.physa.2023.128913.

[13] B. Cai, Y. Wang, C. Huang, J. Liu, and W. Teng, "GLSNN network: A multi-scale spatiotemporal prediction model for urban traffic flow," *Sensors*, vol. 22, no. 22, 2022. doi: 10.3390/s22228880.

[14] 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, 2020. doi: 10.1109/TITS.2019.2935152.

[15] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *6th Int. Conf. Learn. Rep., ICLR 2018*, Vancouver, BC, Canada, 2018.

[16] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 922–929, Jul. 2019. doi: 10.1609/aaai.v33i01.3301922.

[17] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Trans. Knowl. Data. Eng.*, vol. 34, no. 11, pp. 5415–5428, 2022. doi: 10.1109/TKDE.2021.3056502.

[18] X. Zhang, Y. Xu, and Y. Shao, "Forecasting traffic flow with spatial-temporal convolutional graph attention networks," *Neural. Comput. Appl.*, vol. 34, no. 18, pp. 15457–15479, Sep. 2022. doi: 10.1007/s00521-022-07235-z.

[19] Y. Yao, B. Gu, Z. Su, and M. Guizani, "MVSTGN: A multi-view spatial-temporal graph network for cellular traffic prediction," *IEEE Trans. Mob. Comput.*, vol. 22, no. 5, pp. 2837–2849, 2023. doi: 10.1109/TMC.2021.3129796.

[20] G. Wei, "A summary of traffic flow forecasting methods," *J. Highw. Transp. Res. Dev.*, vol. 21, no. 3, pp. 82–85, 2004.

[21] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi-passenger demand using streaming data," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1393–1402, Sep. 2013. doi: 10.1109/TITS.2013.2262376.

[22] J. Wang, P. Shang, and X. Zhao, "A new traffic speed forecasting method based on bi-pattern recognition," *Fluct. Noise. Lett.*, vol. 10, no. 1, pp. 59–75, 2011. doi: 10.1142/S0219477511000405.

[23] X. Luo, D. Li, and S. Zhang, "Traffic flow prediction during the holidays based on DFT and SVR," *J. Sens.*, vol. 2019, no. 10, pp. 1–10, 2019. doi: 10.1155/2019/6461450.

[24] W. Zhang, G. Yang, Y. Lin, C. Ji, and M. Gupta, "On definition of deep learning," in *2018 World Autom. Congr. (WAC)*, 2018, pp. 1–5.

[25] J. Xu, R. Rahmatizadeh, L. Bölöni, and D. Turgut, "Real-time prediction of taxi demand using recurrent neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2572–2581, 2018. doi: 10.1109/TITS.2017.2755684.

[26] H. Zhang, G. Yang, H. Yu, and Z. Zheng, "Kalman filter-based CNN-BiLSTM-ATT model for traffic flow prediction," *Comput. Mater. Contin.*, vol. 76, no. 1, pp. 1047–1063, 2023. doi: 10.32604/cmc.2023.039274.

[27] Y. Zhang, S. Wang, B. Chen, J. Cao, and Z. Huang, "TrafficGAN: Network-scale deep traffic prediction with generative adversarial nets," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 219–230, 2021. doi: 10.1109/TITS.2019.2955794.

[28] Z. Chen *et al.*, "Spatial-temporal short-term traffic flow prediction model based on dynamical-learning graph convolution mechanism," *Inf. Sci.*, vol. 611, pp. 522–539, 2022. doi: 10.1016/j.ins.2022.08.080.

[29] Z. Su, T. Liu, X. Hao, and X. Hu, "Spatial-temporal graph convolutional networks for traffic flow prediction considering multiple traffic parameters," *J. Supercomput.*, vol. 79, no. 16, pp. 18293–18312, Nov. 2023. doi: 10.1007/s11227-023-05383-0.

[30] J. Wang, W. Wang, X. Liu, W. Yu, X. Li and P. Sun, "Traffic prediction based on auto spatiotemporal multi-graph adversarial neural network," *Physica A*, vol. 590, no. 5, 2022. doi: 10.1016/j.physa.2021.126736.

[31] J. Zhao *et al.*, "2F-TP: Learning flexible spatiotemporal dependency for flexible traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15379–15391, 2023. doi: 10.1109/TITS.2022.3146899.

[32] X. Zhang *et al.*, "Traffic flow forecasting with spatial-temporal graph diffusion network," *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 17, pp. 15008–15015, May 2021. doi: 10.1609/aaai.v35i17.17761.

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

[34] J. Bai *et al.*, "A3T-GCN: Attention temporal graph convolutional network for traffic forecasting," *ISPRS Int. J. Geoinf.*, vol. 10, no. 7, 2021. doi: 10.3390/ijgi10070485.

[35] A. Oluwasanmi, M. U. Aftab, Z. Qin, M. S. Sarfraz, Y. Yu and H. T. Rauf, "Multi-head spatiotemporal attention graph convolutional network for traffic prediction," *Sensors*, vol. 23, no. 8, 2023. doi: 10.3390/s23083836.

[36] X. Dong, W. Zhao, H. Han, Z. Zhu, and H. Zhang, "MTESformer: Multi-scale temporal and enhance spatial transformer for traffic flow prediction," *IEEE Access*, vol. 12, pp. 47231–47245, 2024. doi: 10.1109/ACCESS.2024.3381987.

[37] W. Chai, Q. Luo, Z. Lin, J. Yan, J. Zhou and T. Zhou, "Spatiotemporal dynamic multi-hop network for traffic flow forecasting," *Sustainability*, vol. 16, no. 14, 2024. doi: 10.3390/su16145860.

[38] S. Modi, Y. Lin, L. Cheng, G. Yang, L. Liu and W. Zhang, "A socially inspired framework for human state inference using expert opinion integration," *IEEE-ASME T MECH*, vol. 16, no. 5, pp. 874–878, 2011. doi: 10.1109/TMECH.2011.2161094.

[39] Y. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 3634–3640.

[40] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, pp. 1234–1241, Apr. 2020. doi: 10.1609/aaai.v34i01.5477.

[41] B. Oreshkin, A. Amini, L. Coyle, and M. Coates, "FC-GAGA: Fully connected gated graph architecture for spatio-temporal traffic forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 10, pp. 9233–9241, 2021. doi: 10.1609/aaai.v35i10.17114.

[42] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD)*, 2019, pp. 1720–1730.

[43] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," in *Proc. Twenty-Eighth Int. Joint Conf. Artif. Intell. (IJCAI)*, 2019, pp. 1907–1913.

[44] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD)*, 2020, pp. 753–763.

[45] X. Huang, Y. Lan, Y. Ye, J. Wang, and Y. Jiang, "Traffic flow prediction based on multi-mode spatial-temporal convolution of mixed hop diffuse ODE," *Electronics*, vol. 11, no. 19, 2022. doi: 10.3390/electronics11193012.