Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 2811961, 15 pages
https://doi.org/10.1155/2022/2811961

Research Article

MSASGCN: Multi-Head Self-Attention Spatiotemporal Graph Convolutional Network for Traffic Flow Forecasting

Yang Cao, Detian Liu, Qizheng Yin, Fei Xue, and Hengliang Tang

School of Information, Beijing Wuzi University, Beijing 101149, China

Correspondence should be addressed to Hengliang Tang; tanghengliangbwu@163.com

Received 6 February 2022; Revised 22 April 2022; Accepted 14 May 2022; Published 16 June 2022

Academic Editor: Yong Zhang

Traffic flow forecasting is an essential task of an intelligent transportation system (ITS), closely related to intelligent transportation management and resource scheduling. Dynamic spatial-temporal dependencies in traffic data make traffic flow forecasting to be a challenging task. Most existing research cannot model dynamic spatial and temporal correlations to achieve well-forecasting performance. The multi-head self-attention mechanism is a valuable method to capture dynamic spatial-temporal correlations, and combining it with graph convolutional networks is a promising solution. Therefore, we propose a multi-head self-attention spatiotemporal graph convolutional network (MSASGCN) model. It can effectively capture local correlations and potential global correlations of spatial structures, can handle dynamic evolution of the road network, and, in the time dimension, can effectively capture dynamic temporal correlations. Experiments on two real datasets verify the stability of our proposed model, obtaining a better prediction performance than the baseline algorithms. The correlation metrics get significantly reduced compared with traditional time series prediction methods and deep learning methods without using graph neural networks, according to MAE and RMSE results. Compared with advanced traffic flow forecasting methods, our model also has a performance improvement and a more stable prediction performance. We also discuss some problems and challenges in traffic forecasting.

1. Introduction

With the development of society and accelerated urbanization, the demand for urban transportation is growing. The problems arising from traffic congestion and road planning make it essential to have effective traffic management and planning. The rapid development of information technology makes intelligent transportation systems (ITS) gradually become an indispensable and critical part of urban transportation. It can bring efficient traffic management, accurate resource allocation, and traffic service support [1]. Advanced ITS needs efficient traffic data processing, and modeling of traffic data is the first task of ITS. Currently, there are multiple data collection methods in intelligent transportation, and the number of sensors deployed on the roadways has increased significantly. These sensors recorded information about vehicles passing through different road nodes' speed, flow, and size [2]. How to effectively process and analysis these multidimensional data to use them further for traffic prediction is an important research problem.

Traffic forecasting is an integral part of ITS, and timely and accurate traffic forecasting information helps managers make decisions and helps vehicle drivers choose smoother road trips, which can alleviate or avoid problems such as traffic congestion and traffic accidents [3]. Traffic flow forecasting is a crucial task, aiming to use historical traffic data from road networks to predict traffic flow in future time steps [4]. Traffic flow forecasting can be divided into short-term (within 30 min) and long-term (over 30 min) scales based on the future length of the forecast in the time dimension. Traditional forecasting approaches are ineffective in predicting medium- and long-term situations and only have some advantages in short-term forecasting [5]. In addition, traffic flow forecasting relies on sequential patterns in the time dimension and road networks in the spatial dimension. The connectivity relationships between different road nodes can affect each other to influence the overall prediction accuracy [6]. Traffic flow is highly dynamic and spatial-temporal correlated as it changes with time and space and is a nonlinear problem that combines complexity and uncertainty.

For traffic flow forecasting problems, the existing research approaches can be divided into three categories which are classical statistics-based models, traditional machine learning-based models, and deep learning-based models. Due to the massive data generation and growth in the computing power of devices, the primary approaches for traffic flow forecasting are gradually evolving into data-driven deep learning methods [7]. Classical statistical-based forecasting models use limited data for analysis, regression, and optimization but fail to enable forecasting at large data scales and long-term forecasting. Traditional machine learning-based forecasting models mainly use machine learning methods to mine historical traffic flow data trends to predict future traffic status. But the complexity of historical traffic flow data is not effectively handled, making it impossible to achieve good prediction performance. Deep learning-based forecasting models often utilize neural network models, such as CNN and RNN, to model temporal and spatial dependencies. The overall performance of traffic forecasting is improved compared with the previous two approaches. Using deep learning has improved the overall performance of traffic forecasting models, but it is not the best solution yet. The main reason is the lack of adequate consideration of the spatiotemporal correlation of rapidly growing traffic data and the complexity of traffic networks.

Traffic flow forecasting is vital for intelligent transportation applications. Traffic flow data are mainly collected by sensors on the road, with the dynamic influence of the data collected by sensors between different location nodes in a specific time interval. Therefore, modeling the traffic flow forecasting problem is difficult due to the dynamic spatial-temporal correlation of traffic flows. It makes timely and accurate traffic flow forecasting very challenging. Exploring the nonlinear and complex traffic data to capture the temporal dependence and spatial dependence to get the potential spatiotemporal patterns is an essential issue in traffic flow forecasting [8].

Recently, graph neural networks (GNNs) [9] as a novel deep learning method have received a lot of research and attention due to their ability to directly model complex relationships. Representing non-Euclidean data as graphs with complex relationships and interdependencies between objects, GNNs can be effective methods for solving complex problems [10]. Graph neural networks are well suited for the field of traffic forecasting. The spatiotemporal correlation of traffic data can be effectively handled using graph neural networks, which can simultaneously deal with the temporal dynamics and the complexity of road networks, significantly improving the forecasting performance [11].

Although GNNs-based methods have achieved some advantages in traffic flow forecasting, the ability to model dynamic spatiotemporal correlation of traffic data is not perfect. Most current studies have not addressed the highly dynamic nonlinear spatiotemporal correlation challenges in traffic flow forecasting. The information on traffic data observed at different nodes is not entirely independent and is influenced by adjacent nodes and time steps, which are dynamically correlated. Figure 1 illustrates the complex spatial-temporal correlation, showing the spatial and temporal dynamics in traffic forecasting. In subfigure 1(a), three sensors in the road network are distributed in different ways, and even though they are geographically close in the road network, correlations do not always exist. The data information recorded by the sensors all differ. In subfigure 1(b), the correlation between traffic conditions at different time steps is different. For instance, sensor B is more correlated at time step $t + h + 1$ and $t - 1$ than with the nearest time step.

As mentioned above, traffic flow data exhibit strong dynamics and complexity in spatial and temporal dimensions. An accurate traffic flow forecast will depend on the effective treatment of spatiotemporal correlations in complex nonlinear traffic data. We propose a multi-head self-attention spatiotemporal graph convolutional network (MSASGCN) model to address these issues. Our model can effectively capture the potential spatial correlation and dynamic temporal features in the traffic road network. It can be adapted to the dynamic changes of the road network and used for traffic flow forecasting of different time lengths.

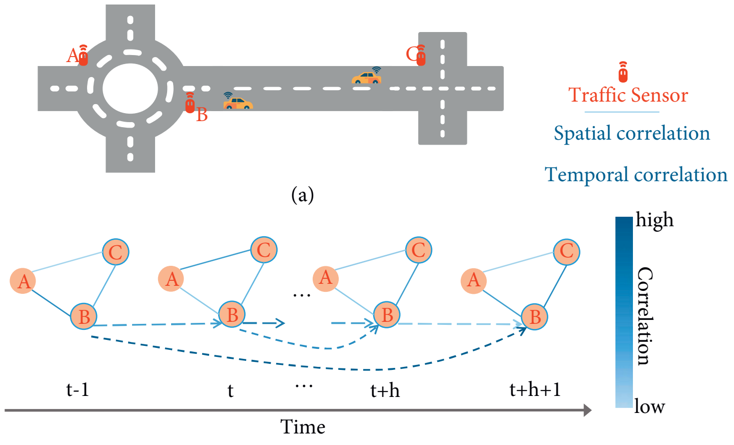
The main contributions of this article are as follows.

(1) To address the challenge of spatial-temporal correlation in traffic flow forecasting, we propose a novel deep learning model, the multi-head self-attention spatiotemporal graph convolutional network (MSASGCN). It can learn the temporal and spatial dependencies of dynamic traffic data and effectively forecast traffic flow in different periods.

(2) We use GCN to construct a spatial correlation model for road networks based on connection relations and a multi-head self-attention mechanism to capture the hidden spatial correlation between road networks and aggregate information among different nodes. A temporal convolution module is added to capture the dynamically changing temporal correlations. And based on the periodic characteristics of time series, an extended MSASGCN model is proposed to handle better the traffic flow forecasting problem with different temporal attributes.

(3) We conducted extensive experiments on real-world datasets to verify the proposed model validity and prediction performance. The experimental results show that our proposed model can achieve better prediction performance than the baseline model. In addition, this article concludes with a short description of some critical issues in traffic flow forecasting research.

The remainder of this article is as organized follows. Section 2 introduces related work on traffic flow forecasting and graph neural network models. Section 3 describes the traffic flow forecasting problem, and Section 4 illustrates our model (MSASGCN) in detail. Section 5 gives the experiment description and analysis of the results. Section 6 provides



(b)

Figure 1: Dynamic spatial-temporal correlation in traffic forecasting. (a) Illustration of sensor distribution in the road network. (b) Dynamic spatial and temporal correlation.

some discussion to explain some potential problems in traffic flow forecasting. Finally, we conclude in Section 7.

2. Related Work

In this section, we first provide a summary of research on graph neural networks and then give an overview of recent traffic flow forecasting research.

2.1. Graph Neural Networks. Graph neural network is a novel model that captures graph dependencies through message passing between graph nodes to solve complex problems [12]. A new classification of graph neural networks was made in [10], respectively, recurrent GNNs (RecGNNs), convolutional GNNs (ConvGNNs), graph autoencoders (GAEs), and spatial-temporal GNNs (STGNNs). Reference [13] provided a comprehensive overview of the general design process, application classification, and some open problems for graph neural network models. Graph convolutional networks (GCNs) extend convolutional operations from traditional to graph data and are the foundation of many complex graph neural network models [14]. There are two categories of GCNs, spectral-based and spatial-based. The spectral-based approach introduced filters to define the graph convolution from the perspective of graph signal processing [15], while the spatial-based approach represents the graph convolution as aggregating feature information from the neighborhood [16]. To improve the effectiveness of long-range information dissemination, combining the gating mechanism of RNNs [17], such as GRU [18] or LSTM [19], with graph neural networks is an effective way. Gated graph neural networks (GGNNs) [20] use GRUs in forwarding propagation to expand RNNs in fixed time steps and compute gradients using a temporal backpropagation algorithm. As research on graph neural networks grows, combining them with other deep learning techniques is becoming a trend. Attention mechanisms [21] have been widely applied to sequence-based tasks, and combining attention mechanisms with graph neural networks yields better aggregation capabilities, integrating information from various components. Graph attention networks (GAT) [22] can efficiently handle the hidden states of nodes and perform well in tasks such as semi-supervised node classification. Apart from GAT, gated attention network (GAAN) [23] can assign different weights to different attention heads using additional soft gating computations. Graph neural network models can be widely used, but some methodological limitations, depth of model, dynamics, and heterogeneity need to be further explored.

2.2. Traffic Flow Forecasting. Research on traffic flow forecasting is an evolutionary process, [24] provided a detailed survey of urban traffic flow forecasting and analyzed some representative methods. The early traffic flow forecasting methods were mainly based on statistics, such as historical averages (HA) [25], time series methods [26], and Kalman filters [27]. Autoregressive integrated moving average (ARIMA) [28] and vector autoregressive (VAR) [29] are two classical methods that both have good performance for time series processing. Although these methods are helpful for traffic flow forecasting, all of them have some limitations. The road network's dynamic time dependence and spatial dependence in traffic data cannot be effectively exploited. Therefore, data-driven deep learning-based forecasting methods gradually become popular and bring good performance improvements. When a large amount of traffic data is accumulated, [4] used deep neural networks to explore the intrinsic relationships hidden in them and improve forecasting accuracy. Reference [30] proposed deep spatial-temporal convolutional network (DSTCN) to learn the spatial features of convolutional neural network and the temporal features of LSTM, but it needs to convert the traffic data into grid data. Reference [31] analyzed the data loss problem in traffic data collection and proposed a reconstruction method with low-rank matrix decomposition to reconstruct road traffic data accurately. Reference [32] designed an enhanced graph convolutional network based on cross-attention fusion with better performance superiority and robustness.

Due to the characteristics of road networks, graph neural networks can directly model the road network and better capture spatial-temporal correlations. Seo et al. [33] proposed graph convolutional recurrent network (GCRN) to extract the topology of traffic networks and find dynamic patterns to optimize traffic forecasting. In [34], a combination of LSTM with graph convolutional networks is proposed, the streaming graph convolutional long short-term memory neural network (TGC-LSTM), which can address the dynamic time variation and complex spatial constraints of road networks. Reference [35] converted the dynamic traffic flow modeling into a diffusion process that can capture spatial dependencies using diffusion convolution operations. Reference [36] proposed the STGCN method, which can model multi-scale traffic networks, effectively capture comprehensive spatial-temporal correlations, and obtain better traffic forecasting results. Reference [37] designed a learnable position attention mechanism that can effectively aggregate information from adjacent roads and better exploit local and global spatial-temporal correlations. Guo et al. [38] used the attention mechanism for traffic flow prediction and proposed an attention mechanism spatial-temporal graph convolutional network (ASTGCN), which can extract temporal features more effectively to improve forecasting performance. Inspired by the low-rank representation and dynamic decomposition model, a low-rank dynamic decomposition model for traffic flow forecasting [39] is proposed for effective short-term traffic flow forecasting. An optimized graph convolutional recurrent network for traffic forecasting was proposed in [40] to improve the forecasting performance by learning the optimized graph data-driven during the training phase to reveal the potential relationships between road segments. Reference [41] considered road scalable and changing road networks, combining continuous learning with GNNs, and proposed the TrafficStream method. Reference [42] proposed a hierarchical graph convolutional network (HGCN) for traffic forecasting, which uses the road network's natural hierarchy to operate on micro and macro traffic maps to achieve traffic forecasting. Reference [43] proposed the spatial-temporal fusion graph neural network (STFGNN), which integrates the fusion graph module with the gated convolution module into one layer and can learn more spatiotemporal dependencies to handle long sequence situations. Reference [44] proposed the transformer network for traffic flow forecasting, which can jointly exploit dynamic directional spatial dependence and long-term temporal dependence to improve the forecasting accuracy. Considering the limitations of acquiring temporal and spatial dependencies separately, [45] designed a spatiotemporal synchronous modeling mechanism to construct the spatial-temporal synchronous graph convolutional network (STSGCN) to acquire complex local spatiotemporal correlations. Reference [46] proposed the STGSA, a spatial-temporal graph self-attention model, to learn graph-level spatial embeddings using graph self-attention layers and gated cyclic units integrated with RNN units to learn temporal embeddings. Reference [47] proposed the graph multi-attention network (GMAN) method, which applies an encoder-decoder structure and can solve the error propagation problem in forecasting. A multi-sensor data-correlated graph convolutional network model is proposed in [48], named MDCGCN, mainly designed with an adaptive benchmark mechanism and multi-sensor data-correlated convolutional blocks that can eliminate the differences between periodic data and capture dynamic spatial-temporal correlations. Reference [49] proposed a multi-range attention bicomponent graph convolutional network that uses bicomponent graph convolution to implement node and edge interaction aggregate information about different neighbors with a multi-range attention mechanism, and automatically learns the importance of different ranges. Therefore, the research of traffic forecasting approach based on graph neural networks can be effective for the time and spatial dependence and has some advantages for dynamic changes. Our work is based on the attention mechanism and combined with graph convolutional neural networks for traffic flow forecasting.

3. Preliminaries

In our work, traffic flow forecasting issues using the graph neural network approach require building the traffic network and defining the forecasting problem representation first. In addition, graph convolutional networks and attention mechanisms are the essential parts of our approach, and we give a brief description of them here.

3.1. Problem Statement. We first need to construct the traffic road network as a graph and illustrate the traffic flow forecasting problem. Define the traffic network as an undirected graph $G = (V, E, A)$, where V denotes a finite set of nodes corresponding to the observations of N sensors in the traffic network, $|V| = N$ nodes, and E is the set of edges to represent the connectivity of nodes. The graph structure of traffic data is shown in Figure 2, and each data node can be considered a graph signal defined on Graph G. If v_i and v_j are two nodes in V with a connection, then (v_i, v_j) is an edge in set E. These connections between nodes can be described by the adjacency matrix $A = (A_{ij})^{N\times N} \in \mathbb{R}^{N\times N}$. A is the adjacency matrix constructed based on the distance relationship between different sensor distributions, which can define and describe the relationships between different nodes in the graph G. The threshold Gaussian kernel approach is used to process the adjacency matrix A, where A_{ij} is the $i, j -$ th element.

$$A_{ij} = \begin{cases} \exp\left(-\dfrac{d_{ij}^2}{\sigma^2}\right), & \text{if } \exp\left(-\dfrac{d_{ij}^2}{\sigma^2}\right) \geq \varepsilon \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where d_{ij} denotes the distance between the sensors v_i and v_j, σ is the standard deviation of the distance between each

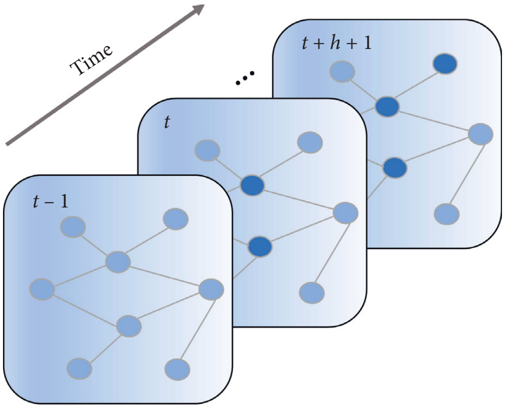


Figure 2: Graph-structured traffic data.

sensor node, and ε is a preset threshold value, which is taken as 0.1.

In the traffic network, each node on G samples F observations at the same frequency, which means that each node generates a feature vector of length F at each time step. We use $x_t^i \in \mathbb{R}^F$ as the value of all F features of node i at time t, and $X_t = (x_t^1, x_t^2, \ldots, x_t^N)^T \in \mathbb{R}^{N\times F}$ denotes the values of all F features of all nodes at time t. Therefore, we can denote the target feature value of all nodes predicted at time step t as $Y_t = (y_t^1, y_t^2, \ldots, y_t^N)^T \in \mathbb{R}^{N\times 1}$. The traffic flow forecasting problem is to predict future traffic conditions based on historical traffic data and the topology of the road network, which can be summarized as follows. The graph structure of traffic data is shown in Figure 2, and sensors are represented by nodes in the graph, allowing the acquisition of information on traffic conditions at different times and spaces.

$$f\left(X_{t-F+1}, X_{t-F+2}, \ldots, X_t, A\right) = \left(Y_{t+1}, Y_{t+2}, \ldots, Y_{t+M}\right), \tag{2}$$

where $X_{t-F+1}, X_{t-F+2}, \ldots, X_t$ denotes historical traffic data, and with the processing of function f, future traffic data series $Y_{t+1}, Y_{t+2}, \ldots, Y_{t+M}$ can be obtained, and A is the adjacency matrix of graph G.The crucial to the traffic forecasting problem is the need to find the function f, the traffic forecasting model, which maps the data series to the future traffic data series.

3.2. Graph Convolutional Networks. Graph convolutional network is a feature extractor for processing unstructured data with strong advantages for non-Euclidean graph-structured data processes. Suppose that a graph containing each node of K is given and the adjacency matrix $A \in \mathbb{R}^{k\times k}$ of this graph is obtained. The output of node i at layer l of the GCN is represented here as h_i^l, and h_i^0 represents the initial state of node i at the time of input to layer 1 of the GCN. For one l-layer GCN, $l \in [1, 2, \ldots, L]$, the final state of node i can be expressed as h_i^L. The following (3) is the computational procedure for the graph convolution of node i.

$$h_i^l = \sigma\left(\sum_{j=1}^{k} A_{ij} W^l h_i^{l-1} + b^l\right), \tag{3}$$

where W^l is the linear transformation weight, b^l is the deviation term, σ represents the activation function, commonly used activation functions such as *ReLU*.

3.3. Attention Mechanism. There are three matrix inputs, key $K \in \mathbb{R}^{n\times d_k}$, query $Q \in \mathbb{R}^{m\times d_q}$, and value $V \in \mathbb{R}^{m\times d_v}$, where n and m represent the lengths of these two inputs, and d_k and d_v represent the dimensional dimensions of the key and value. The attention mechanism is also set up with multiple heads, each of which can pay attention to different location information and learn different features. It computes the weighted sum by calculating the key and value dot product, and then normalizes it by *SoftMax*. And finally using the value projection (V) output. The concrete expression of the formula is as follows.

$$\text{Attention}\, Q, K, V = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V. \tag{4}$$

The shape of Q is $N \times d_q$, which represents the matrix consisting of query vectors of N nodes. The shape of K is $N \times d_k$, representing the matrix consisting of key vectors of N nodes. The shape of V is $N \times d_v$, representing the matrix consisting of value vectors of N nodes. In traffic flow forecasting with a spatial attention mechanism, it is necessary to aggregate node information in the spatial dimension, for which parameters are shared between different time steps.

3.4. Multi-Head Self-Attention Mechanism. The multi-head self-attention mechanism is mainly a process of multiple groups of self-attention on the original input sequence. It is worth noting that the process can be computed in parallel, improving the efficiency of feature extraction. Then, each group of self-attention results is concatenated, and then a linear transformation is performed to obtain the final output results.

$$\text{Multi Head}\, Q, K, V = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_h\right)W^O. \tag{5}$$

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right). \tag{6}$$

The specific calculation is expressed as shown in (5) and (6), where $W_i^Q \in \mathbb{R}^{p_q\times d_q}$, $W_i^K \in \mathbb{R}^{p_k\times d_k}$, and $W_i^V \in \mathbb{R}^{p_v\times d_v}$. The output of multi-head attention requires a linear transformation, which corresponds to the result after concatenate h heads, and therefore, its learnable parameter is $W_i^O \in \mathbb{R}^{p_o\times h_{pv}}$.

$$W_O\begin{bmatrix} \text{head}_1 \\ \vdots \\ \text{head}_n \end{bmatrix} \in \mathbb{R}^{p_O}. \tag{7}$$

Based on this design, each of the heads may focus on a different part of the input and can represent more complex functions than a simple weighted average.

4. Multi-Head Self-Attention Spatiotemporal Graph Convolutional Neural Network

This section describes our forecasting method by detailing the modules that make up the multi-head self-attention spatiotemporal graph convolutional network (MSASGCN) model. We provide a detailed description of the multi-head self-attention and graph convolution module, temporal convolution module, and the extended MSASGCN. It can effectively handle different temporal periods in historical data.

4.1. Architecture of Model. The architecture of our proposed model is illustrated in Figure 3. In addition, based on the research idea of ASTGCN [38], we added parallel sub-models of the same structure to improve the accuracy of prediction. Figure 4 illustrates the overall architecture with the addition of sub-models capturing the daily and weekly characteristics of traffic flow data. We also call the overall architecture an extension of the MSASGCN model, extended MSASGCN.

The multi-head self-attention mechanism and graph convolutional network are combined to capture local and global spatial dependencies, and the information obtained is fused using a gating mechanism. Meanwhile, the temporal convolution is used to capture the temporal dependence to get different influence levels at different times to improve forecasting accuracy. This structure is stacked to obtain more substantial processing power for long sequences or large-scale data. Extend MSASGCN model by adding weekly and daily periods to traffic flow forecasting. Each of these components has the same structure and has the same capability to handle spatial-temporal correlation.

MSA refers to the multi-head self-attention mechanism, GCN denotes graph convolutional network, Temp-Conv denotes temporal convolution, Gated Fusion denotes the gating mechanism to fuse spatial information, and Conv is the convolution operation. GCN is used to acquire local spatial information, and MSA is used to acquire global spatial correlations. The gating mechanism can fuse the extracted spatial correlations. A simple fully connected layer is used at the input layer to map the information to a high-dimensional space to improve the expressiveness of the model. Two convolution layers are used in the output layer for the decay of feature dimensions and the transformation of time series length. More details about the significant component modules of the model are described as follows.

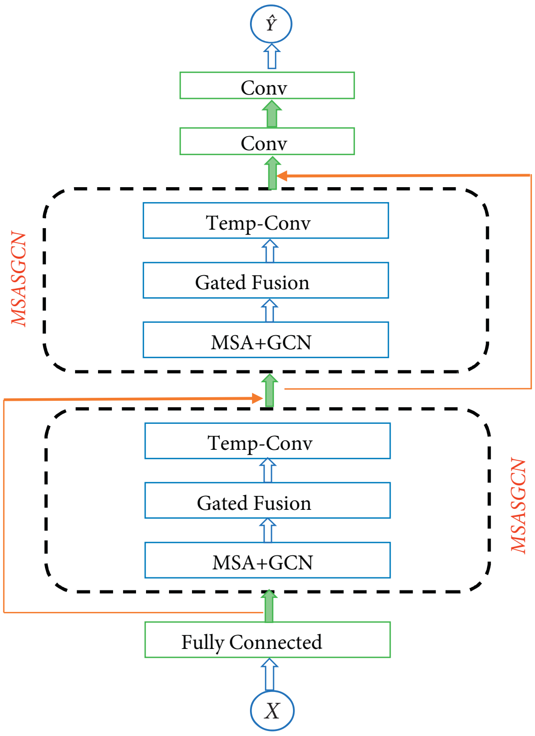


Figure 3: Architecture of MSASGCN. MSA : Multi-head self-attention; GCN : graph convolutional network; Temp-Conv : temporal convolution; Conv : convolution.

4.2. Graph Convolution and Multi-Head Self-Attention Module. Traffic conditions on a road segment are influenced not only by the road segments that are spatially connected to it but also by other factors, and two road nodes that are far apart may still exhibit similar traffic patterns. The spatial correlation of traffic conditions can be influenced by the connectivity between road segments and geographic position attributes. Therefore, local spatial correlation and global correlation need to be considered. We use GCN to aggregate node information from local based on the connectivity between roads and use the multi-head self-attention mechanism to aggregate the hidden global correlations.

Initially, the features of each node are considered as signals on the graph, and then spectral graph-based graph convolution is used to capture the spatial patterns in the traffic network. According to the spectral theory, the traffic graph is represented by the normalized Laplace matrix L in the graph. It can be defined as follows.

$$L = I_N - D^{-1/2} A D^{-1/2}, \tag{8}$$

where I_N is an $N \times N$ unit matrix, N denotes the number of nodes, and A is the adjacency matrix. D is the degree matrix, which is a diagonal matrix with diagonal elements of $D_{ii} = \sum_{j=1}^{N} A_{ij}$, and A_{ij} is the element of the i – th row and j – th column of the adjacency matrix A. The graph convolution can be defined as follows:

$$\theta_{*G} x \approx \sum_{K=0}^{K-1} \theta_K T_K (\tilde{L}) x, \tag{9}$$

where θ_{*G} denotes the graph convolution operation on the signal x in the graph G, $\tilde{L} = 2/\lambda_{\max} L - I_N$ is the normalized Laplace matrix after scaling, $\lambda_{\max}$ is the maximum feature value of L, θ_k is the coefficient of the k – th term of the Chebyshev polynomial, and T_K is k – th order Chebyshev polynomial. Graph convolution with Chebyshev polynomials is used to aggregate information from neighbor nodes to capture local spatial correlations.

To capture the global spatial correlation, it is necessary to consider the changes in the road network structure and the

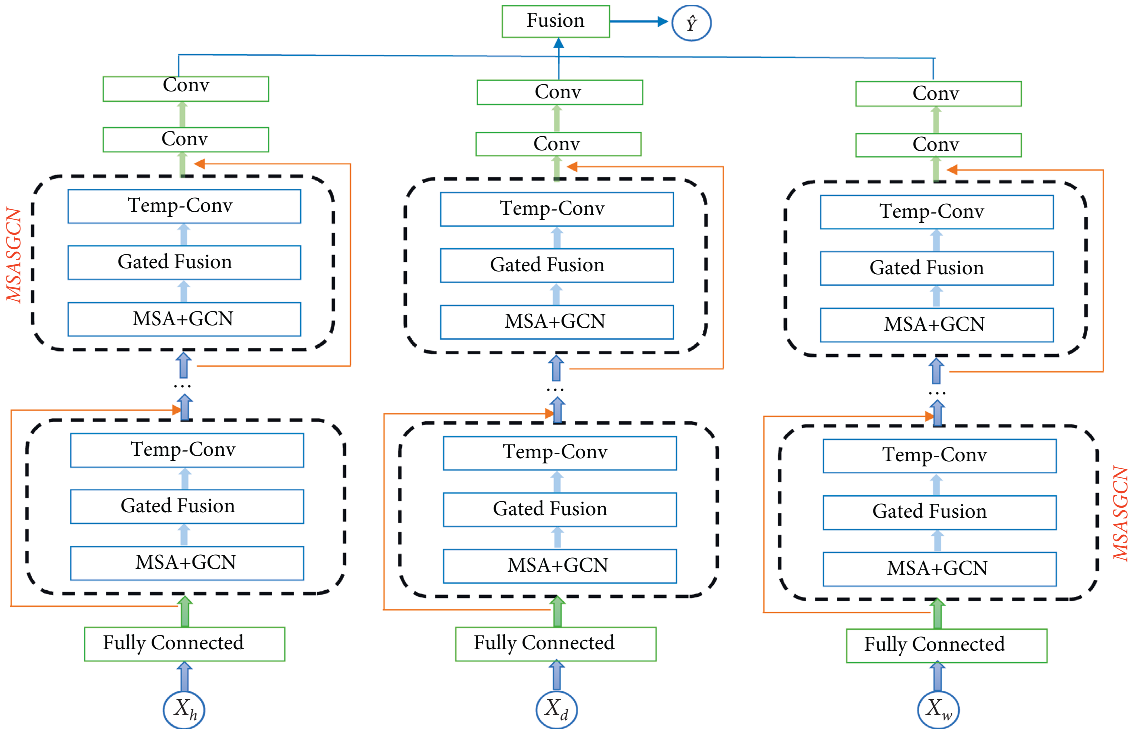


FIGURE 4: Extended MSASGCN architecture with weekly and daily period time dependencies. MSA : Multi-head self-attention; GCN : graph convolutional network; Temp-Conv : temporal convolution; Conv : convolution.

hidden spatial correlation in the road network, and we employ a multi-head self-attention mechanism to aggregate the information. Firstly, the feature vectors of each node are mapped with three different matrices W^Q, W^K, and W^V. Three vectors can be obtained, as described in preliminaries, as Query, Key, and Value. W^Q, W^K, and W^V are learnable parameters that are continuously optimized and updated during the training of the model. With the inner product of the Query vector of each node and the Key vector of all nodes, the *SoftMax* function can compress the vector to between 0 and 1. After normalization by the *SoftMax* function, the attention score of this node with all nodes can be obtained. The *SoftMax* function is defined as follows, where z_i denotes the $i-$ th dimension of the vector and K denotes the dimension of the vector.

$$\text{SoftMax}\left(z_i\right) = \frac{e^{z_i}}{\sum_{K=1}^{K} e^{z_k}}. \tag{10}$$

We represent the attention mechanism in matrix form, which can be calculated using (4). A multi-head self-attention mechanism can aggregate information in several different feature subspaces simultaneously, with different subspaces expressing different implicit spatial correlations. The multi-head self-attention mechanism is performed by linearly mapping Query, Key, and Value n times (n is the number of heads) to get multiple sets of different subspace representations, then performing the attention mechanism on each set, and then stitching them together to get a final result by doing another linear mapping. The following equation can express the multi-head self-attention mechanism.

$$\begin{aligned} h_i &= \text{Attention}\left(XW_i^Q, XW_i^K, XW_i^V\right), \\ \text{Multihead} &= \text{Concat}\left(h_1, h_2, h_3, \ldots, h_n\right)W^O. \end{aligned} \tag{11}$$

The h_i denotes the output of the $i-$ th group of the self-attention mechanism, n denotes the number of heads, Multihead denotes the output of the multi-head self-attention mechanism, Concat denotes the stitching operation on the tensor along the feature dimension that is the $i-$ th group of linear mapping matrices, and W^O is the matrix that maps the result of the stitching. The spatial multi-head self-attention mechanism can learn the implied spatial correlation between nodes based on the features of each node in the input data. It is practical to capture their correlations in the spatial dimension using the multi-head self-attention mechanism. Meanwhile, it can capture when the topology of the road network changes because the attention scores among nodes are dynamically calculated based on the input. In addition, it is also able to capture the spatial correlation of the road network globally since the spatial self-attention aggregates the information of all nodes.

After obtaining local spatial correlation and global correlation, the information gained should be fused using a gating mechanism. The gating mechanism can be used to learn the importance of two kinds of spatial information and fuse the two kinds of information based on the learned weights, represented by the following equations.

$$\begin{aligned} g &= \sigma\left(H_{\text{GCN}}^{(l)}W_1 + H_{\text{Att}}^{(l)}W_2 + b\right), \\ H^{(l)} &= g \odot H_{\text{GCN}}^{(l)} + (1-g) \odot H_{\text{Att}}^{(l)}, \end{aligned} \tag{12}$$

where $H_{GCN}^{(l)}$ denotes the output of the first graph convolution module, $H_{\text{Att}}^{(l)}$ denotes the output of the first multi-head self-attention module, W_1 and W_2 are the mapping matrices, and b is the bias value. $\odot$ denotes the Hadamard product, where the corresponding position elements of the matrix are multiplied together. Moreover, g denotes the output of the gate, using the sigmoid activation function. $H^{(l)}$ is the result of the fusion of two spatial information.

4.3. Temporal Convolution Module. The improved convolution operation captures the temporal correlation in the time dimension. We combine dilated convolution with causal convolution methods for time series correlation prediction. Causal convolution can abstract the sequential problem and make the predicted value closer to the actual value, but it requires many layers or a large filter to increase the perceptual field of the convolution. Dilated convolution can make the filter apply to regions larger than filter length by skipping some inputs and expanding the receptive field without increasing the model complexity. The combination of these two convolution methods forms the temporal convolution (Temp-Conv) module, which facilitates the acquisition of long-term temporal correlation, improves processing efficiency, and can avoid information forgetting when the sequence is too long. Node i has an output value for the q channel at time t that can be expressed by the following equation.

$$Y_{i,t,q} = \sum_{k=1}^{\tau} \sum_{p=1}^{p} W_{k,p,q} * x_{i,t-d(k-1),p}, \qquad (13)$$

where $W_{k,p,q}$ is the element in the convolution kernel, $x_{i,t-d(k-1),p}$ is the element of the input feature, p is the number of input channels, τ is the convolution kernel size, and d is the dilation rate. If the number of output channels is denoted by S, then S sets of convolution kernels are needed. The parameters of these S sets of convolution kernels can be expressed as a tensor $W^{\tau\times P\times S}$ of shape $\tau \times P \times S$, which are learnable parameters that are continuously updated iteratively by minimizing the loss function during the model training.

In Temp-Conv, to maintain the length of the input time series unchanged, a complementary 0 operation is required, but complementary 0 at both sides of the sequence will increase the length of the sequence, so the sequence ends will be cropped before proceeding to the next layer. Moreover, Temp-Conv contains multiple layers of dilated causal convolution, where the parameters of the convolution kernel are shared among different nodes. The tensor H_t of shape $N \times F \times P$ is used to denote the features of N nodes F time steps, and d denotes the dilated causal convolution operation with expansion rate d_*. Thus, the Temp-Conv operation for H_t can be shown as follows.

$$T = W_{d_*} H_t. \qquad (14)$$

The T is result after convolution, and to expand the receptive field more, it is necessary to stack multiple layers of dilated causal convolution. Each layer's expansion rate increases exponentially, and the expansion rate of the l – th layer is $d_*^l = 2^{l-1}$. Hence, the output of the l – th layer can be expressed as follows.

$$T^l = \text{ReLu}\left(W_{d_l^*}^l Y^{l-1}\right). \qquad (15)$$

Different layers get different outputs with different receptive fields, with shallow layers to obtain short-term temporal correlation and deep layers to obtain long-term temporal correlation. Then, the output features of each layer are concatenated according to their dimensions, and the output channels are transformed using a 1×1 convolutional layer to form the final output of Temp-Conv.

$$T = \text{Conv}\left(\text{Concat}\left(T^1, T^2, \ldots, T^c\right)\right), \qquad (16)$$

where Concat denotes concatenation along the feature dimension, Conv denotes a 1×1 convolutional layer, and c denotes the number of layers of the dilated causal convolution.

4.4. Framework of Extended MSASGCN. The extended MSASGCN model is designed to model and process the dependencies of recent, daily, and weekly periods in historical data rather than a single time series input. As shown in Figure 5, we intercept three time series segments of lengths T_h, T_d, and T_w along the time axis as inputs for the recent, daily, and weekly period components, respectively, where T_h, T_d, and T_w are all multiples of the integer T_p, and T_p is the target time to be predicted.

We assume that the data sampling frequency is m times per day and the current time is t_0. The time series input for different time periods is denoted by X_h, X_d, X_w. The recent time segment is X_h, $X_h = \left\{X_{t_{0-T_h+1}}, X_{t_{0-T_h+2}}, \ldots, X_{t_0}\right\} \in \mathbb{R}^{N\times F\times T_h}$, a segment of the historical time series that is directly adjacent to the forecast period T_p. The daily periodic time segment is X_d, $X_d = \left\{X_{t_{0-(T_d/T_p-1)*q+1}}, \ldots, X_{t_{0-(T_d/T_p-1)*q+T_p}}, X_{t_{0-(T_d/T_p-1)*q+1}}, \ldots, X_{t_{0-(T_d/T_p-1)*q+T_p}}, \ldots, X_{t_{0-q+1}}, \ldots, X_{t_{0-q+T_p}}\right\} \in \mathbb{R}^{N\times F\times T_d}$, and consists of the same segments of the past few days as the prediction period. The weekly periodic time segment is X_w, $X_w = \left\{X_{t_{0-7*(T_w/T_p)*q+1}}, \ldots, X_{t_{0-7*(T_w/T_p)*q+T_p}}, X_{t_{0-7*(T_w/T_p-1)*q+1}}, \ldots, X_{t_{0-7*(T_w/T_p-1)*q+T_p}}, \ldots, X_{t_{0-7*q+1}}, \ldots, X_{t_{0-7*q+T_p}}\right\} \in \mathbb{R}^{N\times F\times T_w}$, and consists of time segments from the most recent weeks, with the same weekly attributes and time intervals as the forecast period.

Therefore, in the extended MSASGCN model architecture, the model components dealing with different periods have the same network structure, all with the same setup as in MSASGCN. Finally, the outputs of the three components are combined based on the parameter matrix to obtain the final prediction results, which can better predict the dynamic traffic flow. Extended MSASGCN is a multi-component fusion model, and the correlation of different periods needs to be handled. The sensitivity of the input traffic flow data to the components is inconsistent, so the sub-models within

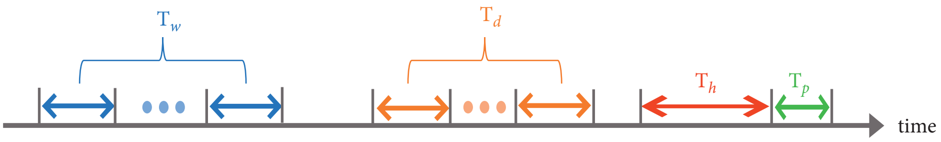


Figure 5: Example of time series segment input.

Table 1: Dataset profiles.

Attributes	PeMSD4	PeMSD8
Time periods	January-February 2018	July-August 2016
Detectors (nodes)	3848	1979
Distance information of sensors (edges)	340	295
Selected detectors	307	170
Sequence length	16,992	17,856
Selected features	3	3

the model have varying degrees of influence on the results. Different time components have different impact levels on each node and should be learned from the historical data and fused according to the different weights to obtain the forecasting results. The updated formula of the forecasting results is as follows.

$$\widehat{Y} = W_w \odot \widehat{Y_w} + W_d \odot \widehat{Y_d} + W_h \odot \widehat{Y_h}, \quad (17)$$

where $\odot$ is Hadamard product, and W_w, W_d, and W_h are learnable parameters that reflect the degree of influence of the three different time-dimensional components on the predicted target.

In some regions, traffic flows may have significant peaks in the morning or evening, making the output of the daily and weekly period components more critical. However, for some other regions, there is no prominent traffic period. Therefore, by fusing the outputs of different components from the above equation, we can obtain traffic flow forecasting results suitable for different regions or periods with different weights.

5. Experiments and Analysis

In this section, to verify the efficacy of our proposed model, we conduct experiments on two real datasets. Firstly, we describe and introduce the experimental datasets and the baseline method of comparison and then define the metrics for the experiments. Finally, the experimental settings and results analysis are given.

5.1. Datasets. PeMSD4 and PeMSD8 are two freeway traffic datasets from California on which we validate our model. The datasets are collected in real time every 30 seconds by the Caltrans Performance Measurement System (PeMS) [49, 50]. Traffic flow data are aggregated from the raw data into intervals of every 5 minutes. The system has over 39,000 detectors deployed on freeways in major metropolitan areas in California. The geographic information of the sensor stations is recorded in the dataset. Three traffic measures are considered in our experiments, including total flow, average speed, and average occupancy. Future traffic flow is our forecasting target. PeMSD4 and PeMSD8 are from different regions, for which details of the dataset are given in Table 1.

PeMSD4 is the San Francisco Bay Area traffic data and contains 3,848 detectors on 29 roads. The periods of this dataset span from January to February 2018. The first 50 days of data were chosen as the training set and the rest as the test set. PeMSD8 is the traffic data of San Bernardino from July to August 2016, which contains 1979 detectors on 8 roads. The first 50 days of data are used as a training set, and the last 12 days of data as the test set.

$$X' = \frac{X - \text{mean}(X)}{\sigma_x}. \quad (18)$$

The selection of detectors required the distance between adjacent detectors to be greater than 3.5 miles. In addition, the missing data are filled linearly. Data processing was performed using zero-mean normalization to make the training process more stable. As shown in (18), mean(X) denotes the mean of the original data, σ_x is the standard deviation of the original data X, and X' is the normalized data.

5.2. Baselines and Experiment Metrics. We compare our model with the following baselines:

(i) HA [25]: Historical average, using the average of historical data to predict the next value.

(ii) VAR [29]: Vector autoregressive, capture pairwise relationships in traffic flow sequences for prediction.

(iii) ARIMA [28]: Autoregressive integrated moving average method is a classical time series forecasting algorithm that combines autoregressive models, moving average models, and differencing methods.

(iv) LSTM [19]: Long short-term memory network, a variant of RNN.

(v) GRU [18]: Gated recurrent unit network, a variant of RNN.

(vi) DCRNN [35]: Diffusion convolutional recurrent neural network, modeling traffic flow as a diffusion question process on a directed graph.

(vii) STGCN [36]: Spatial-temporal graph convolutional networks that model temporal and spatial dependencies.

(viii) ASTGCN [38]: Attention-based spatial-temporal graph convolutional networks, exploiting spatial-temporal attention mechanisms to model spatio-temporal correlations.

(ix) GeoMAN [51]: An attention-based multilevel recurrent neural network model for geo-aware time series prediction problems.

The baseline and MSASGCN method are compared with the same metrics in our experiments. We use the mean absolute error (MAE) and root mean square absolute error (RMSAE) as performance metrics for experimental evaluation, expressed in the following equations.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{N}\left|X_i - X'_i\right|,$$
$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{N}\left(X_i - X'_i\right)^2}. \tag{19}$$

In addition, we also compared with the mean absolute percentage error (MAPE) in some of the baselines, with the following definition.

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{N}\left|\frac{X_i - X'_i}{X_i}\right|, \tag{20}$$

where X_i and X'_i denote the $i -$ th element in the true and predicted values, respectively, and n denotes the total number of elements.

5.3. Experiment Settings. We have implemented our model using the PyTorch deep learning framework. Future traffic flow is our forecasting target. On the one hand, we use the 1-hour historical traffic flow to forecast the future 1-hour traffic flow situation. Both the input time series and output time series lengths are set to 12, and the time series input length can be adjusted depending on the prediction time. We set the batch size to 64, the learning rate to 0.001, and the Chebyshev polynomial K to 3. The dimensions of the input layer, the implicit layer, and the output layer of the graph convolution module are taken to be 16, 64, and 128, and the input dimension, the dimension of key and value, and the number of heads of the multi-head self-attention module are taken as 16, 128, 128, and 4, respectively. The L_1 loss function is used to minimize the difference between the predicted results and the true value, and the L_1 loss for multi-step prediction is defined as follows.

$$L_1\left(W_\theta\right) = \sum_{i=t+1}^{t=P}\left|X_{:,i} - X'_{:,i}\right|. \tag{21}$$

Table 2: Average performance comparison of future 1-hour traffic flow prediction experiments.

Model	PeMSD4		PeMSD8	
	MAE	RMSE	MAE	RMSE
HA	36.76	54.14	29.52	44.03
VAR	33.76	51.73	21.41	31.21
ARIMA	32.11	68.13	24.04	43.30
LSTM	29.45	45.82	23.08	37.06
GRU	28.65	45.11	22.22	36.95
DCRNN	22.93	33.44	16.82	28.06
STGCN	25.15	38.29	17.51	27.09
ASTGCN	21.80	32.84	16.63	26.51
GeoMAN	23.64	37.84	17.84	28.91
MSASGCN (ours)	**21.22**	**32.09**	**16.23**	**26.24**

Bold is to highlight our experimental results.

The purpose of training the model is to continuously and iteratively update W_θ to minimize L_1, and $X_{:,i}$ and $X'_{:,i}$ denote the labels and predicted values of all nodes at time step i, respectively. On the other hand, we verified the efficiency of our method for traffic flow forecasting in different time prediction intervals. We conducted experiments to predict the future 10, 20, 30, and 40 minutes and analyzed the performance evaluation metrics.

5.4. Comparison and Result Analysis. We compared our model with the baseline approaches on PeMSD4 and PeMSD8. The average results of the future one-hour traffic flow prediction performance are shown in Table 2. It could be seen that our method achieves excellent performance in both datasets for MAE and RMSE evaluation metrics. In the case of traditional time series forecasting methods, they have limited analytical power to deal with spatial-temporal dependence, and the forecasting results are not very satisfactory.

Through comparison and analysis, it is clear that the performance of the deep learning-based approach is significantly better than that of the traditional approach. However, methods such as LSTM and GRU, which fail to handle temporal and spatial correlations effectively, also perform much weaker than methods that capture spatial-temporal correlations. Therefore, it can be concluded that the use of graph neural networks and their variants is effective in handling traffic flow forecasting. The algorithms GeoMAN and ASTGCN, which apply the attention mechanism, also outperform the other algorithms, demonstrating the effectiveness of using the attention mechanism to obtain spatial-temporal correlations.

Based on the experimental results obtained on the PeSMD4 dataset, a detailed description is given in Table 3. The analysis of the traffic flow prediction results for the next 10, 20, 30, and 40 minutes shows the effectiveness of our proposed method for different prediction intervals. Our method's MAE and RMSE evaluation metrics constantly change with increasing time intervals, which is a normal trend. Despite slight fluctuations, the performance is still within an average and the excellent band as the prediction interval increases.

Table 3: Average performance of experiments with different prediction intervals on PeMSD4.

Dataset	Model	Prediction interval (min)	MAE	RMSE
PeMSD4	MSASGCN	10	19.73	29.16
		20	20.42	30.24
		30	20.78	31.33
		40	21.01	31.82

Table 4: Average performance of experiments with different prediction intervals on PeMSD8.

Dataset	Model	Prediction interval (min)	MAE	RMSE
PeMSD8	MSASGCN	10	15.31	24.01
		20	15.72	25.24
		30	15.93	25.93
		40	16.01	26.13

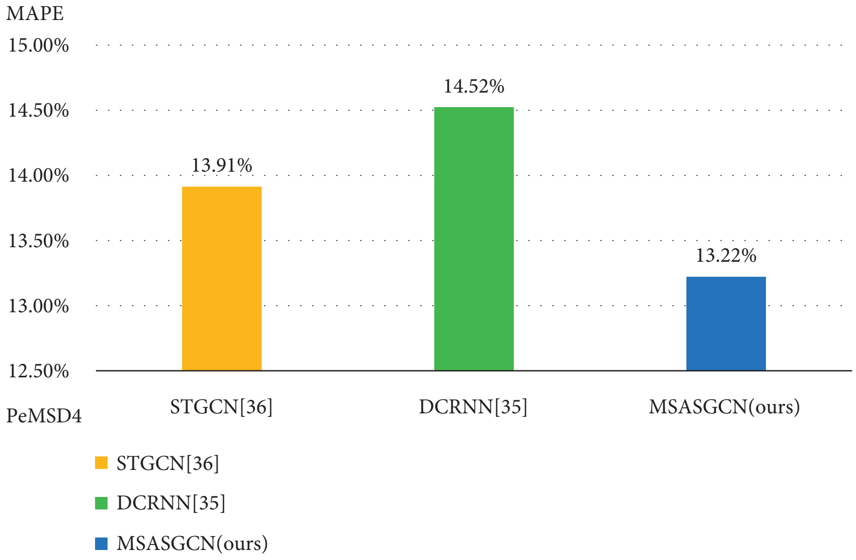


Figure 6: Comparison of different algorithms MAPE on PeMSD4.

Similarly, experimental results were obtained on the PeMSD8 dataset, detailed in Table 4. The experimental results on PeMSD8 are better than the PeMSD4 dataset, and although different datasets present different results, the overall trend is considered similar. MAE and RMSE performance evaluation metrics show that our proposed method can cope with traffic flow situations with different prediction intervals. According to the experimental results of different prediction intervals performed on two datasets, our proposed method can solve the traffic flow forecasting issue in the short or long term. Our proposed method has some advantages and reasonableness to capture the temporal and spatial characteristics nicely.

Moreover, we also compared the mean absolute percentage error (MAPE) of the MSASGCN model with STGCN and DCRNN, which also obtained better results than these two methods. This indicated that MSASGCN could better handle spatial-temporal correlations to capture the dynamically changing temporal and spatial dependence. The results obtained from the experiments on the PeMSD4 and PeMSD8 datasets are shown in Figures 6 and 7, respectively. Our method uses a multi-head self-attention mechanism to effectively fuse local and global spatial correlations, aggregate information from multiple nodes, and extract implicit information to improve prediction accuracy.

5.5. Validation of Module Effectiveness. To better represent the effectiveness of our proposed model, we modify the MSASGCN model by removing the MSA module and using only GCN to process the spatial dependencies, and all other experimental settings are consistent with the original

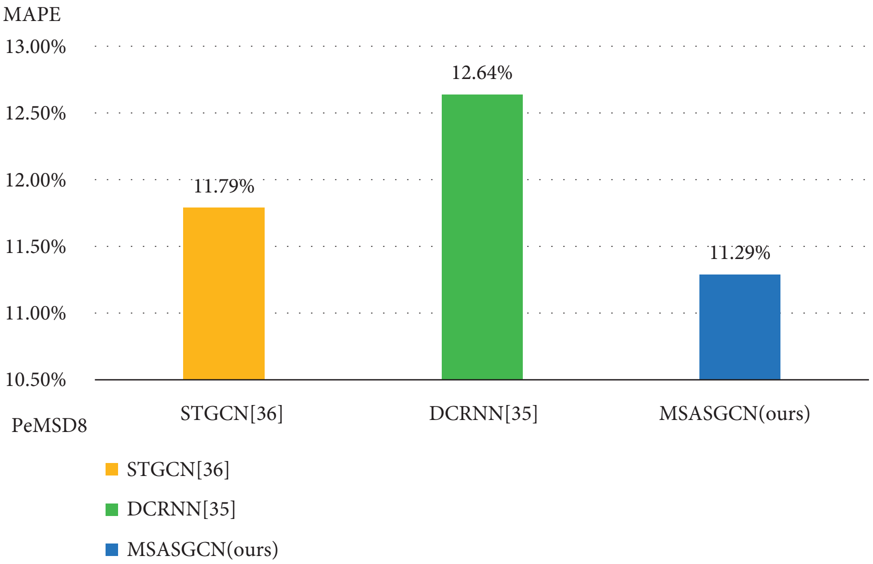


Figure 7: Comparison of different algorithms MAPE on PeMSD8.

Table 5: Introduction of the model name.

Name	Description
MSASGCN	Our original method
MSASGCN-s	Method for removing the multi-head self-attention mechanism

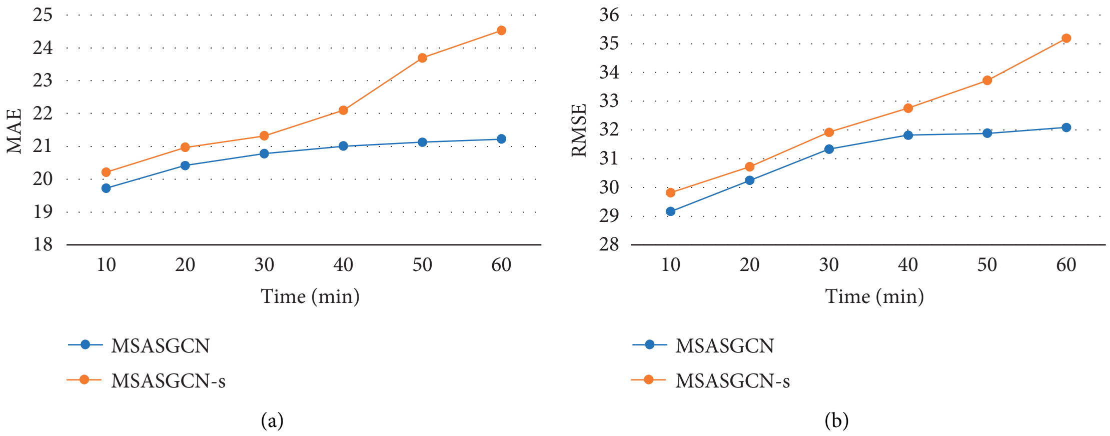


Figure 8: Module validation on the dataset PeMSD4. (a) The evolution of MAE. (b) The evolution of RMSE.

method. Table 5 shows our description of removing the MSA module. We have conducted experiments on PeMSD4 and PeMSD8 to remove the multi-head self-attention mechanism, respectively, and the MAE and RMSE metrics have a significant change in magnitude, which is not as good as the performance of the original MSASGCN method. The experimental results are shown in Figure 8 and Figure 9. Therefore, the multi-head self-attention mechanism plays an essential role in our method.

6. Discussion on Traffic Flow Forecasting

Although the performance of traffic flow forecasting has been significantly improved by applying graph neural networks, there are still some challenges for traffic flow forecasting. Reference [52] provided a summary of the challenges and future directions of traffic forecasting.

(1) From the data perspective, traffic data are heterogeneous and involve spatial-temporal factors and external factors. How well the heterogeneous data are handled can directly affect the forecasting accuracy. Data quality issues can also bring additional challenges.

(2) The timely accuracy of traffic forecasting is critical, but most graph neural network models require much computation and cannot make some real-time forecasting. Building a lightweight and general

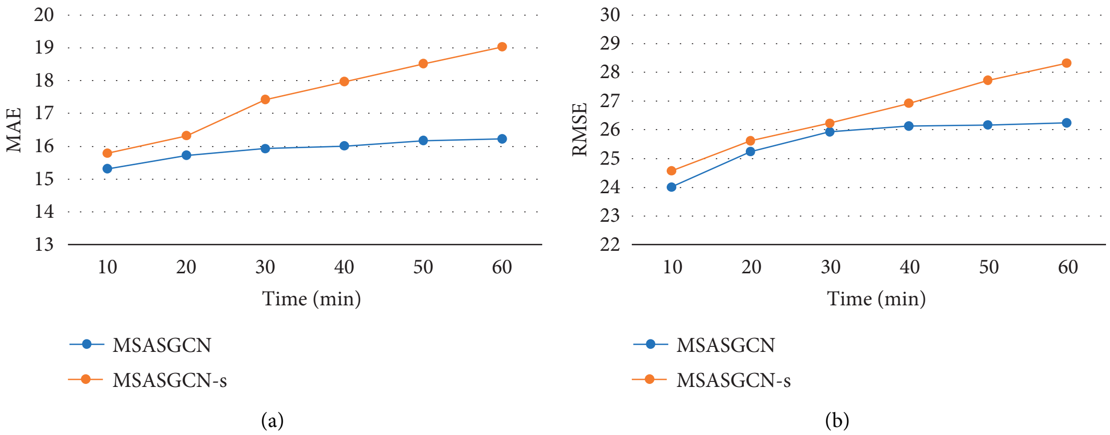


Figure 9: Module validation on the dataset PeMSD8. (a) The evolution of MAE. (b) The evolution of RMSE.

framework is a challenge for traffic forecasting and an essential requirement for intelligent transportation systems.

(3) Intelligent transportation systems need to integrate a different traffic information for analysis and processing simultaneously. Traffic forecasting models not only need to be able to process a specific task demand, but more importantly, they may process multiple tasks at the same time, which is a crucial challenge for multi-task forecasting.

(4) Privacy and security issues in traffic forecasting. The large-scale traffic data collection by IoT devices such as sensors has potential data security and privacy threats. The use of federated learning for graph neural network models in traffic forecasting in [53] is an approach of future interest, applying the distributed structure of federated learning, which allows some data protection.

7. Conclusions

In this article, we propose the multi-head self-attention spatiotemporal graph convolutional network (MSASGCN) model. Combining the multi-head self-attention mechanism with graph convolutional network can effectively handle the spatial-temporal correlation of traffic data. Our model can accommodate both the road network's dynamic time dependence and spatial dependence and performs better in capturing the spatial-temporal characteristics. Experiments on two real-world datasets showed that the prediction accuracy of our proposed model outperformed the baseline model. Our model verified the ability of processing spatial-temporal features simultaneously to construct the graph convolutional module and the spatiotemporal attention module to improve the prediction performance. In this article, we also summarize some of the challenges of traffic forecasting, and in the future, we will investigate the critical challenges of traffic flow forecasting intensively.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the General Program of Science and Technology Development Project of Beijing Municipal Education Commission of China (No. KM202110037002), Humanity and Social Science Research of Ministry of Education (No. 20YJCZH200), Research on Intelligent Inventory Optimization Decision Driven by Data (No.2021XJKY01), Beijing Intelligent Logistics System Collaborative Innovation Center Open Topic (No. BILSCIC-2019KF-05).

References

[1] A. Boukerche, Y. Tao, and P. Sun, "Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems," *Computer Networks*, vol. 182, Article ID 107484, 2020.

[2] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: trends, methods and challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, p. 1, 2020.

[3] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, "Road traffic forecasting: recent advances and new challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 93–109, 2018.

[4] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[5] E. Bolshinsky and R. Friedman, “Traffic Flow Forecast survey,” No. CS Technion Report CS-2012-06, Computer Science Department, Technion, Haifa, Israel, 2012.

[6] M. Treiber and K. Arne, “Traffic flow dynamics,” *Traffic Flow Dynamics: Data, Models and Simulation*, Springer-Verlag, Berlin Germany, pp. 983–1000, 2013.

[7] H. Zhu, Y. Xie, W. He et al., “A novel traffic flow forecasting method based on RNN-GCN and BRB,” *Journal of Advanced Transportation*, vol. 202011 pages, Article ID 7586154, 2020.

[8] J. An, L. Guo, W. Liu et al., “IGAGCN: information geometry and attention-based spatiotemporal graph convolutional networks for traffic flow prediction,” *Neural Networks*, vol. 143, pp. 355–367, 2021.

[9] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How Powerful Are Graph Neural Networks?,” 2018, https://arxiv.org/abs/1810.00826.

[10] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[11] J. Ye, J. Zhao, K. Ye, and C. Xu, “How to build a graph-based deep learning architecture in traffic domain: a survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 3904–3924, 2022.

[12] F. Scarselli, M. Gori, and A. C. Tsoi, M. Hagenbuchner and G. Monfardini, The graph neural network model,” *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[13] J. Zhou, G. Cui, S. Hu et al., “Graph neural networks: a review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, 2020.

[14] T. N. Kipf and M. Welling, “Semi-supervised Classification with Graph Convolutional Networks,” 2016, https://arxiv.org/abs/1609.02907.

[15] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, “Spectral Networks and Locally Connected Networks on Graphs,” 2013, https://arxiv.org/abs/1312.6203.

[16] A. Micheli, “Neural network for graphs: a contextual constructive approach,” *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009.

[17] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent Neural Network Regularization,” 2014, https://arxiv.org/abs/1409.2329.

[18] K. Cho, B. Van Merriënboer, C. Gulcehre et al., “Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation,” 2014, https://arxiv.org/abs/1406.1078.

[19] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated Graph Sequence Neural Networks,” 2015, https://arxiv.org/abs/1511.05493.

[21] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph Attention Networks,” 2017, https://arxiv.org/abs/1710.10903.

[23] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, “Gaan: Gated Attention Networks for Learning on Large and Spatiotemporal Graphs,” 2018, https://arxiv.org/abs/1803.07294.

[24] P. Xie, T. Li, J. Liu, S. Du, X. Yang, and J. Zhang, “Urban flow prediction from spatiotemporal data using machine learning: a survey,” *Information Fusion*, vol. 59, pp. 1–12, 2020.

[25] A. G. Hobeika and C. K. Kim, “Traffic-flow-prediction systems based on upstream traffic,” in *Proceedings of the VNIS’94 - 1994 Vehicle Navigation and Information Systems Conference*, pp. 345–350, Okohama, Japan, September 1994.

[26] M. S. Ahmed and A. R. Cook, *Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques*, Transportation Research Board, Washington, D.C, USA, 1979.

[27] I. Okutani and Y. J. Stephanedes, “Dynamic prediction of traffic volume through Kalman filtering theory,” *Transportation Research Part B: Methodological*, vol. 18, no. 1, pp. 1–11, 1984.

[28] B. M. Williams and L. A. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results,” *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[29] E. Zivot and J. Wang, “Vector Autoregressive Models for Multivariate Time Series,” *Modeling Financial Time Series with S-Plus®*, pp. 385–429, Springer, Berlin, Germany, 2006.

[30] J. Zhang, Y. Zheng, and D. Qi, “Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction,” in *Proceedings of the Thirty-first AAAI conference on artificial intelligence*, pp. 1655–1661, San Francisco, CA, USA, February 2017.

[31] Y. Wang, Y. Zhang, X. Piao, H. Liu, and K. Zhang, “Traffic data reconstruction via adaptive spatial-temporal correlations,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1531–1543, April 2019.

[32] G. Huo, Y. Zhang, J. Gao, B. Wang, Y. Hu, and B. Yin, “CaEGCN: cross-attention fusion based enhanced graph convolutional network for clustering,” *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2021.

[33] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, “Structured sequence modeling with graph convolutional recurrent networks,” *Structured Sequence Modeling with Graph Convolutional Recurrent Networks*, Springer, Cham, Switzerland, pp. 362–373, 2018.

[34] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, “Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, Nov. 2020.

[35] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting,” 2017, https://arxiv.org/abs/1707.01926.

[36] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting,” 2017, https://arxiv.org/abs/1709.04875.

[37] X. Wang, Y. Ma, Y. Wang et al., “Traffic flow prediction via spatial temporal graph neural network,” in *Proceedings of the Web Conference 2020*, pp. 1082–1092, Taipei Taiwan, April 2020.

[38] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, “Attention based spatial-temporal graph convolutional networks for traffic flow forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 922–929, Honolulu, Hawaii, February 2019.

[39] Y. Yu, Y. Zhang, S. Qian, S. Wang, Y. Hu, and B. Yin, “A low rank dynamic mode decomposition model for short-term traffic flow prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6547–6560, 2021.

[40] K. Guo, Y. Hu, Z. Qian et al., “Optimized graph convolution recurrent neural network for traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 1138–1149, 2021.

[41] X. Chen, J. Wang, and K. Xie, "TrafficStream: A Streaming Traffic Flow Forecasting Framework Based on Graph Neural Networks and Continual Learning," 2021, https://arxiv.org/abs/2106.06273.

[42] K. Guo, Y. Hu, and Y. Sun, "Hierarchical graph convolution networks for traffic forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, pp. 151–159, February 2021.

[43] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, pp. 4189–4196, February 2021.

[44] M. Xu, W. Dai, C. Liu et al., "Spatial-temporal Transformer Networks for Traffic Flow Forecasting," 2020, https://arxiv.org/abs/2001.02908.

[45] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: a new framework for spatial-temporal network data forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 914–921, New York, NY, USA, February 2020.

[46] Z. Kang, H. Xu, J. Hu, and X. Pei, "Learning Dynamic Graph Embedding for Traffic Flow Forecasting: A Graph Self-Attentive Method," in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference*, pp. 2570–2576, ITSC, Auckland, New Zealand, October 2019.

[47] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: a graph multi-attention network for traffic prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 1234–1241, 2020.

[48] W. Li, X. Wang, Y. Zhang, and Q. Wu, "Traffic flow prediction over muti-sensor data correlation with graph convolution network," *Neurocomputing*, vol. 427, pp. 50–63, 2021.

[49] W. Chen, L. Chen, Y. Xie, W. Cao, Y. Gao, and X. Feng, "Multi-range attentive bicomponent graph convolutional network for traffic forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 3529–3536, New York, NY, USA, February 2020.

[50] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1748, no. 1, pp. 96–102, 2001.

[51] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: multi-level attention networks for geo-sensory time series prediction," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, vol. 2018, pp. 3248–3434, Stockholm, Sweden, July 2018.

[52] W. Jiang and J. Luo, "Graph Neural Network for Traffic Forecasting: A Survey," 2021, https://arxiv.org/abs/2101.11174.

[53] C. Meng, S. Rambhatla, and Y. Liu, "Cross-node federated graph neural network for spatio-temporal data modeling," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*, pp. 1202–1211, Singapore, August 2021.