Scientific Research Publishing

Journal of Computer and Communications, 2024, 12, 52-67
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Multi-Head Attention Spatial-Temporal Graph Neural Networks for Traffic Forecasting

Xiuwei Hu[1], Enlong Yu[2], Xiaoyu Zhao[3]

[1]School of Computer Science and Technology, Shandong University of Technology, Zibo, China
[2]Shandong Deyun Land Real Estate Appraisal Consulting Co., Zibo, China
[3]College of Computer and Control Engineering, Qiqihar University, Qiqihar, China
Email: hxw_sdut@163.com

How to cite this paper: Hu, X.W., Yu, E.L. and Zhao, X.Y. (2024) Multi-Head Attention Spatial-Temporal Graph Neural Networks for Traffic Forecasting. *Journal of Computer and Communications*, **12**, 52-67.
https://doi.org/10.4236/jcc.2024.123004

Received: February 13, 2024
Accepted: March 12, 2024
Published: March 15, 2024

Abstract

Accurate traffic prediction is crucial for an intelligent traffic system (ITS). However, the excessive non-linearity and complexity of the spatial-temporal correlation in traffic flow severely limit the prediction accuracy of most existing models, which simply stack temporal and spatial modules and fail to capture spatial-temporal features effectively. To improve the prediction accuracy, a multi-head attention spatial-temporal graph neural network (MSTNet) is proposed in this paper. First, the traffic data is decomposed into unique time spans that conform to positive rules, and valuable traffic node attributes are mined through an adaptive graph structure. Second, time and spatial features are captured using a multi-head attention spatial-temporal module. Finally, a multi-step prediction module is used to achieve future traffic condition prediction. Numerical experiments were conducted on an open-source dataset, and the results demonstrate that MSTNet performs well in spatial-temporal feature extraction and achieves more positive forecasting results than the baseline methods.

Keywords

Traffic Prediction, Intelligent Traffic System, Multi-Head Attention, Graph Neural Networks

1. Introduction

ITS is capable of providing efficient traffic management and accurate traffic resource allocation. Accurate prediction of future traffic conditions is the core of intelligent transportation systems. Accurate traffic prediction is an important guideline for resource rationalization and dynamic traffic planning [1]. As a typical regression problem of traffic time series, traffic prediction aims to predict future traffic conditions (such as traffic flow and speed) in the road network based

on historical observation sequences (such as historical data recorded by sensors). Each time slot in **Figure 1(a)** corresponds to a traffic flow state, and the traffic flow state of the next M time steps is predicted based on the traffic flow state of the previous N time steps. **Figure 1(b)** provides an intuitive illustration of traffic flow prediction, where all attribute values are normalized to [0, 1]. Traffic prediction is challenging because it involves complex spatial relationships and temporal dependencies. Firstly, the spatial relationships of real roads between different regions are very complex. Secondly, there is also a strong temporal dependence across time dimensions in the road network. The traffic conditions of roads vary nonlinearly and non-stationarily over time. For example, as shown in **Figure 2(a)**, in a complex traffic road, if a traffic accident occurs at node 4, the traffic condition of that node will suddenly become congested, thereby affecting the traffic conditions of nodes 1, 2, and 3. Therefore, capturing spatial correlations and temporal dependencies has become the core of the prediction task.

In the early days, classical statistical models were popular due to their relatively simple principles and rigorous mathematical theory verification. For instance, C.K. Moorthy *et al.* proposed an autoregressive moving average model (ARMA) for traffic prediction [2], and M. Van Der Voort *et al.* improved the ARMA model by adding differencing to capture time-varying relationships, resulting in the autoregressive integrated moving average model (ARIMA) [3]. However, many studies have shown that although extending the ARIMA model to seasonal autoregressive integrated moving average model (SARIMA) can improve model performance [4], these models are still challenged to handle the nonlinear and uncertain features of traffic flow data due to the predominance of linear and stationary assumptions.

To alleviate some issues with classical statistical models, researchers have developed machine learning-based traffic prediction methods to handle relatively complex traffic flow prediction tasks. For example, M. Lippi *et al.* proposed the use of Support Vector Regression models in traffic prediction, N. Zarei *et al.* suggested using the Random Forest prediction method for traffic prediction [5], and P. Cai *et al.* proposed the use of k-Nearest Neighbor models in short-term traffic prediction [6]. However, the effectiveness of these models heavily depends on complex feature engineering, and these methods may not be sufficient in capturing the complex spatial-temporal relationships in large-scale data.

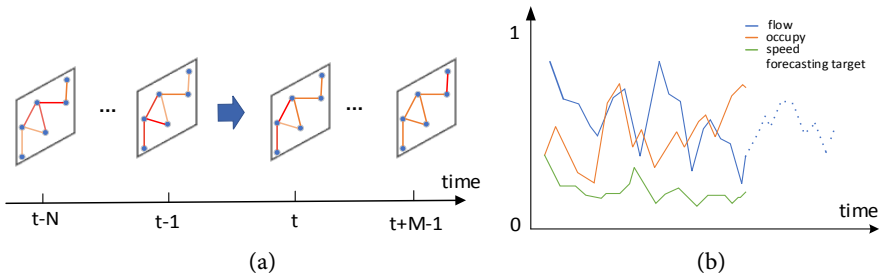

Figure 1. Illustration of the spatial-temporal traffic flow forecasting. (a) Changes in the spatial-temporal state of traffic; (b) Detect three attribute values at a node and predict future traffic flow.

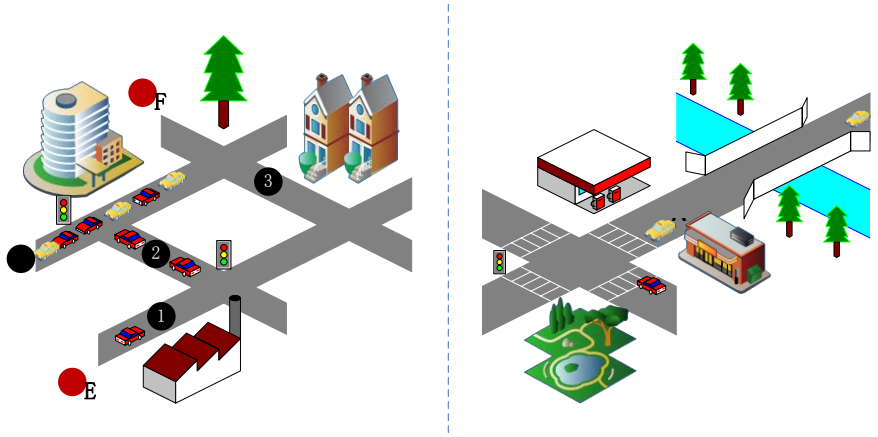

Figure 2. Spatial correlations could be affected.

In recent years, deep learning methods have been widely applied to various transportation tasks due to their excellent learning capabilities and have achieved remarkable results [7]. Existing deep learning methods can be divided into two categories: grid-based methods and graph-based methods. The former divides the study area into regular grids and uses convolutional neural networks (CNN) [8] and recurrent neural networks (RNN) [9] to capture temporal dependencies. This method ignores the topological structure of the road network, resulting in insufficient use of spatial relationships between different regions. The latter constructs a graph structure based on the road network topology to capture hidden spatial relationships. Spatial-Temporal Graph Neural Networks (ST-GNNs) have been extended to the field of traffic prediction to achieve more satisfactory performance [10]. Most existing ST-GNNs first construct a predefined graph structure and then study the constructed graph. For example, Li *et al.* [11] proposed to use gated recurrent units (GRU) instead of matrix generation operations in graph convolution. Yu *et al.* [12] extracted the spatial correlation of traffic conditions through graph convolutional neural networks (GCN) and captured temporal dependencies through causal convolution. Such graphs are mostly determined by the Euclidean distance between each pair of regions, but complex traffic is influenced by multiple hidden factors such as road functions and regional distribution, resulting in the problem of the same distance but different spatial relationships. **Figure 2(b)** briefly illustrates that although node A and node B have the closest Euclidean distance, the relationship between node A and node C is more relevant. Therefore, constructing a graph based on functional similarity or traffic connections cannot reflect all aspects of the spatial structure. Wu *et al.* [13] proposed an adaptive learning graph adjacency matrix for traffic prediction, Bai *et al.* [14] proposed an adaptive graph convolutional network, but ignored the fact that the graph structure changes over time. Ta *et al.* [15] proposed an adaptive graph structure learning component to capture spatial correlations, but ignored the issue that over-learning node attributes may affect model performance.

Traffic forecasting is a classic task in ITS [16], and early work focused mainly on statistical methods, such as autoregressive integrated moving average methods and Kalman filters [17]. Although these methods have strong interpretability, their linear and stationary assumptions limit their performance in traffic prediction. Machine learning-based methods, such as support vector regression and k-nearest neighbor models, have the ability to handle relatively complex dependencies, but manual feature engineering is more time-consuming. Deep learning [18] can automatically extract representations from networks, making it widely used in the field of transportation and achieving good performance. Due to the natural structural characteristics of transportation networks, many researchers consider directly modeling traffic data on graphs, which can be collectively referred to as STGNNs [19]. Graph-based research mainly captures spatial correlations and temporal dependencies through graph neural networks. For example, Li *et al.* modified GRU by replacing its matrix generation operation with graph convolution for traffic prediction. Zhao *et al.* [20] proposed a T-GCN model to learn the topological structure of the road network and dynamic changes in data. Bai *et al.* proposed AGCRN, which can capture time dependencies and spatial correlations. Most existing spatial-temporal graph neural networks construct predefined adjacency matrices based on spatial distance or functional similarity, but this approach ignores some relevant information in complex scenarios, resulting in a decline in model performance. For time modeling, existing methods are mainly divided into three categories: RNN-based methods, CNN-based methods, and attention-based methods. RNN-based methods usually use long-short-term memory (LSTM) or gated recurrent units (GRU) as the basic block for time modeling. RNNs used to model long sequences can lead to gradient disappearance problems, and their sequential nature makes parallelization impossible during training. In contrast, CNN-based methods are easy to parallelize. Graph WaveNet uses multiple stacked dilated 1D convolutions to exponentially expand the receptive field, but long-term correlations will be diluted and cannot be effectively utilized. Attention-based methods focus on each time position in parallel, making them more advantageous in long-term modeling.

Graph structure learning aims to learn optimized graph structures and their representations. Existing graph structure learning methods can be classified into three categories: measure learning methods, probabilistic modeling methods, and direct optimization methods. Measure learning methods define different metrics to measure node relationships in a graph and refine the graph structure by learning the metric functions. Probabilistic modeling methods sample graphs from certain distributions and model the sampling probabilities of edges with learnable parameters. Direct optimization methods treat the entire graph as a learnable parameter and use graph neural network parameters to optimize it. In the field of traffic prediction, recent methods have directly used learning parameters to construct graph structures, but neglected the role of node attributes. This makes it difficult for the model to optimize when the training data is sparse.

Experiments have shown that node attributes can better improve the functionality of the model.

In this paper, a traffic prediction method named multi-head attention spatial-temporal graph neural networks (MSTNet) is proposed, which effectively captures complex spatial-temporal relationships in traffic data by utilizing MST-Block in both temporal and spatial dimensions. Additionally, an adaptive graph structure learning component is designed to more effectively leverage node attributes in the data. Our contributions can be summarized as follows:

• Integrating attentional mechanisms into the constructed MSTNet enables effective capture of spatial-temporal heterogeneity in traffic prediction tasks.

• To obtain the optimal graph adjacency matrices that more effectively reflect the time-varying spatial relationships in the short term, an adaptive graph structure learning component is constructed, which incorporates multi-head attention mechanisms.

• The research combines the multi-head attention mechanism with bottleneck residual blocks and integrates them into the spatial-temporal module. This integration allows for a more accurate focus on valuable information and potential spatial-temporal correlations within the data during training.

• Experimental evaluations were conducted on real-world datasets and compared with various benchmark methods to assess the effectiveness and superiority of the proposed model. The results of the experiments demonstrate the effectiveness and superiority of the proposed model.

2. Methodology

2.1. Question Formulation

Table 1 displays the implication of the symbols used in this article. The task of traffic prediction is to anticipate the future traffic conditions of each area, based on the historical traffic records of N regions on the traffic network.

Table 1. Notions and description.

Notion	Description
N	Number of nodes
F	Dimension of node attributes
S	Window size of historical traffic conditions
T	Window size of future traffic conditions
x	Node attributes that record historical traffic conditions
y	Real future traffic conditions
$\hat{y}$	Predicted future traffic conditions
$G = (V, E, A)$	Graph defined by nodes, edges and adjacency matrix
$L(\cdot, \cdot)$	Graph structure learner
$P(\cdot, \cdot)$	Multi-step traffic condition predictor

According to the previous study, N regions and their paired connections are defined as weighted digraphs $G=(V,E,A)$ in our approach. Where V is a set of $|V|=N$ nodes, E is a set of edges, $A\in R^{N\times N}$ is a weighted adjacency matrix that represents the node proximity between any pair of nodes. The traffic history records at time t is represented as a graphic signal $X_{(t)}\in R^{N\times F}$, where F is the dimension of each node's attribute. Given the historical S traffic information $x\in\left[X_{(t-S+1)},\cdots,X_{(t)}\right]$ and the adjacency matrix $A\in R^{N\times N}$ of the traffic network, the traffic flow prediction problem is defined as a function f which predicts the graphical signal $\hat{y}\in\left[\hat{X}_{(t+1)},\cdots,\hat{X}_{(t+T)}\right]$ for the next T-step, as follows.

$$\hat{y}=f(x,A) \tag{1}$$

2.2. The Model

The architecture of the proposed MSTNet model is illustrated in Figure 3, which comprises an input module, stacked spatial-temporal attention modules, skip convolutional modules, and an output module. The main idea of the model is to learn complex spatial-temporal correlations by constructing MST-Block, each of which is composed of T-Layer, MAT-Layer, and S-Layer. The MAT-Layer utilizes a multi-head attention mechanism and residual blocks. The graph adjacency matrix used in the S-Layer is derived from the output of the adaptive graph structure learning module. To efficiently learn deep neural networks, the output of each MST-Block is passed through the skip convolutional module. Ultimately, the feature fusion mechanism is applied in the output module to achieve multi-step traffic forecasting. The learning process of the MSTNet model is formally defined as follows:

$$A^{*}=L(X,A). \tag{2}$$

$$\hat{y}=P\left(X,A^{*}\right). \tag{3}$$

where $L(X,A)$ is employed to optimize the graph adjacency matrix, taking node attributes X and pre-defined adjacency matrix A as inputs, and outputting the optimal adjacency matrix A^{*}. Meanwhile, $P\left(X,A^{*}\right)$ is applied to achieve traffic prediction, and the output represents the final prediction result.

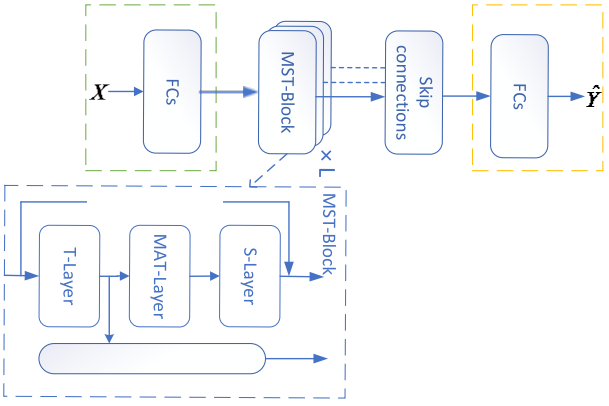

Figure 3. Framework of the proposed MSTNet.

2.3. Graph Structure Learning

As shown in **Figure 4(d)**, the adaptive graph structure learning component consists of two modules: macro-level graph structure learning and micro-level graph structure learning. These modules adaptively infer the adjacency matrices of the macro and micro-level graphs, respectively. Finally, the two adjacency matrices are fused together. This can be formally expressed as follows:

$$A^* = Norm\left(ReLU\left(A_{ma} + A_{mi}\right)\right). \tag{4}$$

where $ReLU$ as the activation function and $Norm$ is utilized as the normalization function.

The macro-level graph structure learning module complements descriptive information by capturing implicit factors that are difficult to capture through learning predefined rules. Denoting the trainable hidden relationships between nodes in the graph as Δ, and the predefined adjacency matrix as A. Residual connection is used to generate the macro-level adjacency matrix A_{ma}. This can be formally expressed as follows:

$$A_{ma} = A_{pre} + \Delta. \tag{5}$$

To implement Δ, a direct optimization method is employed, which involves designing a learnable graph adjacency matrix E. This can be formally expressed as follows:

$$\Delta = E. \tag{6}$$

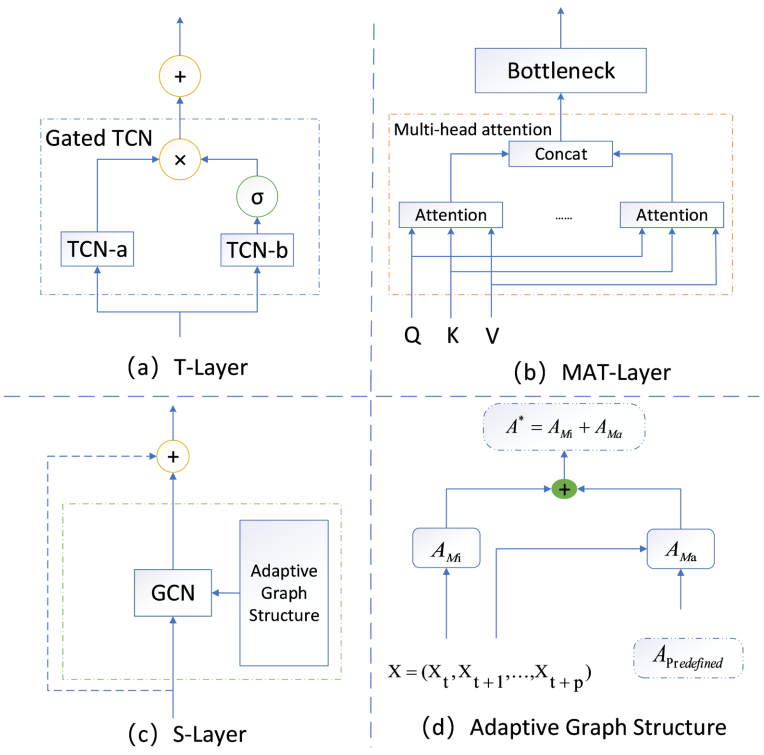

Figure 4. The architecture of MST-Block.

Looking at it from another perspective, variations in factors such as road construction, weather conditions, and traffic congestion may directly impact changes in the structure of a graph. The fluctuations are captured by mining relevant node attribute information. The initial attribute $X \in R^{S\times N\times F}$ is expanded to dimension D from F by applying a 1 × 1 convolution operation $C(\cdot)$, in order to effectively capture information about uncertain factors that may affect node spatial relationships. A multi-head attention mechanism $MHA(\cdot)$ is utilized to extract this information.

$$M = MHA(C(X)). \tag{7}$$

To complete the metric learning approach, the temporal dimension is reduced by cross-utilizing dilated convolutions. This method derives the relationships between nodes by learning a metric function $\varphi(\cdot,?)$ for dual-node representations, as shown below.

$$A_{mi}[i,j] = \varphi(M_i, M_j), \text{ for } 1 \le i, j \le N. \tag{8}$$

Where $A_{mi}[i,j]$ denotes the learned relationship between node i and node j. The proximity between nodes is represented using the dot product, and the metric learning function can be defined as follows:

$$A_{mi} = M \cdot M^{\mathrm{T}}. \tag{9}$$

2.4. MST Block

When dealing with long sequences, the problems of vanishing or exploding gradients are common for RNN models. To overcome these issues, a Temporal Convolution Network (TCN) with causal convolution can be used to capture the temporal dependencies of traffic conditions, as it has several advantages. Firstly, by increasing the dilation factor, TCN can enlarge the receptive field to handle longer sequences. Secondly, TCN can capture longer sequences with fewer layers than traditional RNN models, saving computing resources and time. To improve performance and accelerate model convergence, a Gated-TCN is constructed using the gated linear unit (GLU). Specifically, two TCN models, TCN-a and TCN-b, are built using dilated convolutional neural networks. TCN-b is then used to generate gating signals, which are dot-multiplied with TCN-a to form the Gated-TCN model. This approach enhances the model's representation capability and improves its ability to handle long sequences.

$$h = TCN_a(X) \odot \sigma(TCN_b(X)). \tag{10}$$

where σ represents the Sigmoid function and the Hadamard product. $h \in R^{T\times N\times D}$ is the output of the Gated-TCN.

To enhance the generalization ability of the attention mechanism and minimize errors across different prediction time steps, a multi-head attention layer was introduced between the time-dependent and spatially correlated layers. This layer can effectively model both historical and future time steps, capturing and combining different dependencies (e.g., short-term and long-term dependencies). Therefore, it is beneficial to use different subspaces of queries, keys, and

values in combination. Its input is the feature values of time nodes, *i.e.*, $\vec{h}=\{\vec{h}_1,\vec{h}_2,\cdots,\vec{h}_N\}$, $\vec{h}_i \in R^F$, where N is the number of nodes and F is the dimension of node features. The output dimension is F'. First, a weight matrix $W \in R^{F'\times F}$ is applied to each node, and then self-attention is used to calculate an attention coefficient for each node. The formal expression is as follows:

$$e_{ij}=a\left(W\vec{h}_i,W\vec{h}_j\right). \tag{11}$$

where e_{ij} is the influence coefficient of node i on node j. The non-linear activation function used in this paper is *LeakyReLU*, which can be defined by the following formula:

$$e_{ij}=LeakyReLU\left(\vec{a}^{\mathrm{T}}\left[W\vec{h}_i \,\|\, W\vec{h}_j\right]\right). \tag{12}$$

where, $\|$ denotes the concatenation operation. To better distribute weights between nodes, normalization is applied to the coefficients calculated for the target node and all its neighbor nodes. The formal expression is as follows:

$$\alpha_{ij}=softmax_j\left(e_{ij}\right)=\frac{exp\left(e_{ij}\right)}{\sum_{k\in N_i} exp\left(e_{ik}\right)}. \tag{13}$$

where k represents the neighboring nodes of node i. The normalized attention coefficients are linearly combined with their corresponding nodes to serve as the final output feature vector of each node. The formula is as follows:

$$\vec{h}_i'=\sigma\left(\sum_{j\in N_i}\alpha_{ij}W\vec{h}_j\right). \tag{14}$$

where K groups of single-head attention layers are used, which are mutually independent, and their results are concatenated.

Graph Convolutional Networks (GCN) were employed to capture the complex spatial relationships among nodes, as it is an effective method for extracting features from graph data. The graph adjacency matrix A^* used in our approach was obtained from the graph structure learning component mentioned earlier. Li *et al.* constructed a directed graph for the transportation network and modeled the spatial relationships between nodes using diffusion convolutional operations.

$$Z=X\star_{\mathcal{G}}\theta=\sum_{k=0}^{K}P^kX\theta_k. \tag{15}$$

In the context of a directed graph, where capturing the influence of upstream and downstream traffic flow is crucial, a bidirectional diffusion process is modeled to define the diffusion graph convolution, denoted by $\star_{\mathcal{G}}$. P represents the transition matrix, and k denotes the diffusion length. The diffusion graph convolution can be expressed as follows:

$$Z=X\star_{\mathcal{G}}\theta=\sum_{k=0}^{K}P_f^kX\theta_k+P_b^kX\theta_k. \tag{16}$$

where P_f denotes the forward transition matrix, and P_b denotes the backward transition matrix. A stack of graph convolutional layers is utilized in the network, combined with the graph adjacency matrix A^* obtained from the graph structure learning component described above. Formally, graph convolutional layer can be defined as:

$$\mathcal{Z} = X \star_{\mathcal{G}} \theta_{\mathcal{G}^*}. \tag{17}$$

where $\mathcal{G}^*$ stands for graph with optimal structure A^* and $\theta_{\mathcal{G}^*}$ are trainable parameters.

Residual network is adopted to enhance the model performance, effectively addressing the problem of network degradation. Specifically, the main path is added to the module with the residual edge after two convolutional operations, followed by *relu* activation function. Notably, no hyperparameters are used in this process. The formal definition is as follows:

$$x_{l+1} = x_l + \mathcal{F}\left(x_l, \mathcal{W}_l\right). \tag{18}$$

where $\mathcal{F}\left(x_l, \mathcal{W}_l\right)$ denotes the residual part consisting of two 3 × 3 convolutional operations. A residual connection is added in each MST-Block to improve model performance. Specifically, the output of the l^{th} S-Layer and T-Layer can be obtained by

$$\mathcal{X}^{(1)} = x_{l+1} + \mathcal{X}^{(l-1)}. \tag{19}$$

Then, the features from different S-Layer and T-Layer are fused together through skip connections.

$$z = \sum_l FC_{skip}^{(l)}\left(z^{(l)}\right). \tag{20}$$

where $FC_{skip}^{(l)}$ is a fully connected network at lth S-Layer and T-Layer.

The short-term spatial-temporal features and long-term spatial-temporal features captured by stacking multiple MST-Block are fused to achieve the final traffic condition prediction. A fully connected layer network is applied to directly predict the traffic conditions of all network nodes at T steps. Additionally, a residual connection is added in each ST-Layer to improve the model's performance.

$$\hat{y} = FC_{out}\left(\vec{h}_i' + z\right). \tag{21}$$

where FC_{out} represents the output of the fully connected network.

The training objective of MSTNet is Mean Absolute Error (MAE), and the loss function for multi-step traffic prediction is jointly optimized. The mathematical expression of the loss function for MSTNet is as follows:

$$\mathcal{L}\left(y, \hat{y}\right) = \frac{1}{T \times N \times D} \sum_{i=1}^{T} \sum_{j=1}^{N} \sum_{k=1}^{D} \left| y_{i,j,k} - \hat{y}_{i,j,k} \right|. \tag{22}$$

where $y_{i,:,:}$ is the ground truth, and $\hat{y}_{i,:,:}$ is the prediction of all nodes at time step i.

3. Experiments

3.1. Datasets

1) The METR-LA dataset contains traffic information collected from 207 loop detectors on the highways in Los Angeles County. The dataset records traffic speed statistics from March 1, 2012 to June 30, 2012.

2) The PEMS-BAY dataset contains traffic speed statistics collected from 325 sensors in the harbor area from January 1, 2017 to June 30, 2017, and was collected by the Performance Measurement System of the California Department of Transportation.

The speed data was aggregated into 5-minute windows and pairwise road network distances between sensors were calculated for both datasets. The data was split into 70% for training, 20% for testing, and the remaining 10% for validation. The detailed statistical information of the datasets is shown in **Table 2**.

3.2. Evaluation Metrics

The performance of our model was evaluated using three widely-used indicators: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). MAE and RMSE measure the absolute magnitude of the deviation of the true value from the predicted value, while MAPE measures the relative magnitude (*i.e.*, percentage) of the deviation. Compared to MSE/RMSE, which amplifies the prediction error by using the square of the error, MAE and MAPE are less susceptible to extreme values. However, MSE/RMSE can highlight error values that have a large impact and are more sensitive to outlier data.

- Mean Absolute Error (MAE):

$$\mathrm{MAE}\left(\hat{Y},Y\right)=\frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i-y_i\right|. \tag{23}$$

- Mean Absolute Percentage Error (MAPE):

$$\mathrm{MSE}\left(\hat{Y},Y\right)=\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i-y_i\right)^2. \tag{24}$$

- Root Mean Squared Error (RMSE):

$$\mathrm{RMSE}\left(\hat{Y},Y\right)=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i-y_i\right)^2}. \tag{25}$$

3.3. Baselines

The study compared MSTNet with the following baseline methods: 1) Autoregressive integrated moving average (ARIMA). 2) Fully connected LSTM with hidden units (FC-LSTM). 3) Spatial-temporal graph convolutional networks (ST-GCN). 4) Diffusion convolutional recurrent neural network (DCRNN). 5) Convolutional network architecture (Graph WaveNet), which introduces adaptive graph structure and dilated convolutions to capture spatiotemporal correlations. 6) Adaptive spatial-temporal graph neural network for traffic forecasting (AdaSTNet). 7) Multi-Head self-Attention spatiotemporal graph convolutional network for traffic flow forecasting (MSASGCN). 8) Spatial-Temporal graph attention networks for traffic flow forecasting (STGAT). 9) A graph Multi-Attention network for traffic prediction (GMAN). Grid search was used to select the best hyperparameters for all neural network-based methods based on the validation set performance.

Table 2. The statistics of METR-LA and PEMS-BAY.

Dataset	Nodes	Edges	Time windows
METR-LA	207	1515	34,272
PEMS-BAY	325	2369	52,116

3.4. Experimental Setting and Analysis

All experiments use the traffic speed in the last hour to predict traffic speed for the following hour or more, *i.e.*, $S = T = 12$. We determine parameters through manual hyperparameter tuning. The number of SMT-Blocks was set to 6 to fully cover the input length. Each SMT-Block contains a S-Layer with an inflation factor of 2, a multi-head attention layer with head = 4, and a graph convolutional layer. Dropout with p = 0.5 is applied on the output of the graph convolutional layer. The number of filters for all layers in Gated-TCN and GCN is set to 32 on both datasets. The node dimension in the adaptive graph structure is set to 6, and p = 0.3 is applied on M. During training, the Adam optimizer is used with an initial learning rate of 0.001, which is fine-tuned with a learning rate of 0.00001. The batch size is set to 64, and MAE is used as the training loss. To avoid overfitting, the model is validated on the validation set after each epoch, and early stopping is adopted.

The study conducts 15-minute, 30-minute, and 60-minute ahead predictions on both datasets and compares the performance of MSTNet with baseline models. As shown in **Figure 5**, the proposed MSTNet achieves excellent performance in all prediction horizons and all metrics. **Table 3** reveals that: 1) deep learning models outperform traditional statistical models; 2) graph-based models, including DCRNN and ST-GCN, outperform FC-LSTM in graph-based deep learning methods applied to the traffic domain; 3) models using adaptive graph learning, such as Graph WaveNet, Ada-STNet, and MSTNet, outperform graph-only models; 4) MSTNet proposed in this paper outperforms Graph WaveNet and Ada-STNet models.

From **Figure 5(a)**, it can be observed that MSTNet is able to obtain smaller MAE error values compared to other models. From **Table 3**, it can be seen that the MAE and RMSE error values of the proposed model on the METR-LA dataset have decreased to 2.60 and 5.02, respectively. **Figure 5** demonstrates that the nonlinear modeling capability of deep learning neural networks is highly effective and that road network information has a significant impact on traffic prediction.

In order to confirm the distinctions between MSTNet and other models, the predicted and true values one hour ahead on the PEMS-BAY test set (including node 81) are plotted. As shown in **Figure 6**, the predicted values of the MSTNet model being closer to the true values indicate the robustness of the model. When there is a significant change in speed (e.g., at 9:00 and 12:00), Graph WaveNet cannot immediately capture the change and has a significant delay. At 11:00, a

large fluctuation in the predicted values of the DCRNN model is observed, which deviates from the true values.

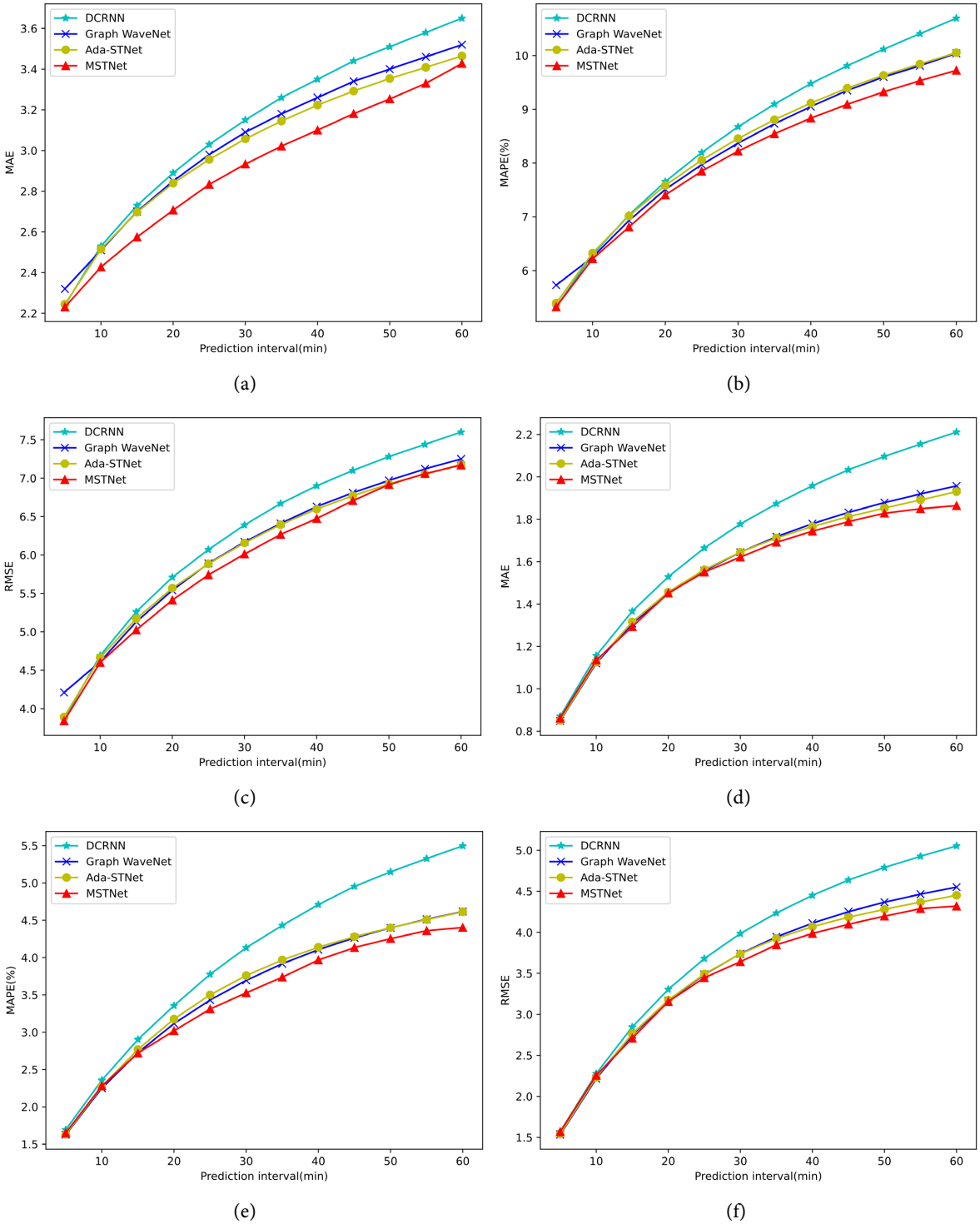

Figure 5. Comparison of stepwise predicted MAE, MAPE and RMSE results for different models on the MATR-LA and PEMS-BAY datasets. (a) Compare MAE results at MATR-LA; (b) Compare MAPE results at MATR-LA; (c) Compare RMSE results at PEMS-BAY; (d) Compare MAE results at PEMS-BAY; (e) Compare MAPE results at MATR-LA; (f) Compare RMSE results at PEMS-BAY.

Table 3. Performance comparison of multi-step traffic condition forecasting.

Dataset	Model name	15 min			30 min			60 min		
		MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
METR-LA	ARIMA	3.99	8.12	9.6	5.15	10.45	12.7	6.90	13.23	17.4
	FC-LSTM	3.44	6.30	9.6	3.77	7.23	10.9	4.37	8.69	13.2
	ST-GCN	2.88	5.74	7.6	3.47	7.24	9.6	4.59	9.40	12.7
	DCRNN	2.77	5.38	7.3	3.15	6.45	8.8	3.60	7.60	10.5
	Graph Wave-Net	2.69	5.15	6.9	3.07	6.22	8.4	3.53	7.37	10.0
	Ada-STNet	2.65	5.06	**6.8**	3.03	6.08	8.2	3.47	7.18	9.8
	MSASGCN	2.75	5.32	7.41	3.11	6.48	8.6	3.81	8.24	12.0
	STGAT	2.66	5.12	6.9	3.01	6.12	**8.1**	3.46	7.19	9.8
	GMAN	2.69	5.55	7.4	3.15	6.78	9.0	4.03	8.11	11.7
	MSTNet	**2.60**	**5.02**	**6.8**	**2.93**	**6.01**	8.2	**3.45**	**7.17**	**9.7**
PEMS-BAY	ARIMA	1.62	3.30	3.5	2.33	4.76	5.4	3.38	6.50	8.3
	FC-LSTM	2.05	4.19	4.8	2.20	4.55	5.2	2.37	4.96	5.7
	ST-GCN	1.36	2.96	2.9	1.81	4.27	4.2	2.49	5.69	5.8
	DCRNN	1.38	2.95	2.9	1.74	3.97	3.9	2.07	4.74	4.9
	Graph Wave-Net	**1.30**	2.74	**2.7**	1.63	3.70	3.7	1.95	4.52	4.6
	Ada-STNet	**1.30**	2.73	**2.7**	1.62	3.67	3.6	1.89	4.36	4.5
	MSASGCN	1.34	2.90	2.9	1.75	3.88	3.9	2.12	4.71	5.0
	STGAT	1.32	2.76	2.8	**1.61**	3.68	3.7	1.91	4.43	4.6
	GMAN	1.34	2.82	2.8	1.62	3.72	3.6	**1.86**	**4.32**	**4.3**
	MSTNet	**1.30**	**2.71**	**2.7**	1.62	**3.64**	**3.5**	**1.86**	**4.32**	4.4

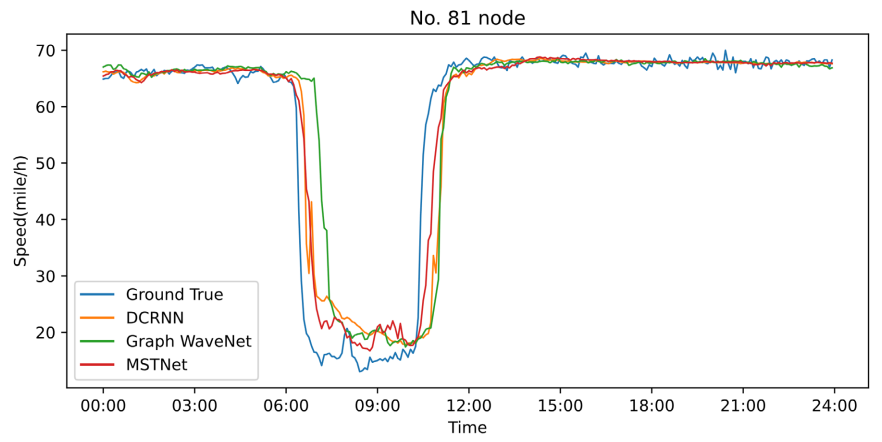

Figure 6. Comparison of prediction curves for one hour ahead prediction on a snapshot of the data of PEMS-BAY.

4. Conclusion

This paper proposes an adaptive spatial-temporal graph neural network model

based on the multi-head attention mechanism for traffic flow prediction. By designing an adaptive graph structure learning component, it accurately reflects the true dependency relationships between nodes and the dynamic correlations within the road network. By combining the multi-head attention mechanism with bottleneck residual blocks and embedding them between Gate-TCN and GCN, it effectively captures the spatial-temporal correlations in the traffic data. To further enhance performance, a "two-stage training algorithm" is integrated into the model training. Through extensive experiments on two datasets, the proposed method has been shown to outperform existing baseline models, particularly in medium to long-term prediction. Future research directions include handling imbalanced datasets and exploring multi-scale input approaches for traffic data to further improve prediction performance.

Conflicts of Interest

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

References

[1] Lv, Y., Duan, Y., Kang, W., Li, Z. and Wang, F.Y. (2014) Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, **16**, 865-873. https://doi.org/10.1109/TITS.2014.2345663

[2] Moorthy, C.K. and Ratcliffe, B.G. (1988) Short Term Traffic Forecasting Using Time Series Methods. *Transportation Planning and Technology*, **12**, 45-56. https://doi.org/10.1080/03081068808717359

[3] Van Der Voort, M., Dougherty, M. and Watson, S. (1996) Combining Kohonen Maps with ARIMA Time Series Models to Forecast Traffic Flow. *Transportation Research Part C: Emerging Technologies*, **4**, 307-318. https://doi.org/10.1016/S0968-090X(97)82903-8

[4] Lippi, M., Bertini, M. and Frasconi, P. (2013) Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning. *IEEE Transactions on Intelligent Transportation Systems*, **14**, 871-882. https://doi.org/10.1109/TITS.2013.2247040

[5] Zarei, N., Ghayour, M.A. and Hashemi, S. (2013) Road Traffic Prediction Using Context-Aware Random Forest Based on Volatility Nature of Traffic Flows. *Intelligent Information and Database Systems*: 5*th Asian Conference*, *ACIIDS* 2013, Kuala Lumpur, 18-20 March 2013, 196-205. https://doi.org/10.1007/978-3-642-36546-1_21

[6] Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C. and Sun, J. (2016) A Spatiotemporal Correlative k-Nearest Neighbor Model for Short-Term Traffic Multistep Forecasting. *Transportation Research Part C: Emerging Technologies*, **62**, 21-34. https://doi.org/10.1016/j.trc.2015.11.002

[7] Reza, S., Oliveira, H.S., Machado, J.J. and Tavares, J.M.R. (2021) Urban Safety: An Image-Processing and Deep-Learning-Based Intelligent Traffic Management and Control System. *Sensors*, **21**, Article No. 7705. https://doi.org/10.3390/s21227705

[8] Guo, S., Lin, Y., Feng, N., Song, C. and Wan, H. (2019) Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. *Pro-*

ceedings of the AAAI Conference on Artificial Intelligence, **1**, 922-929. https://doi.org/10.1609/aaai.v33i01.3301922

[9] Chen, C., Li, K., Teo, S.G., Zou, X., Wang, K., Wang, J. and Zeng, Z. (2019) Gated Residual Recurrent Graph Neural Networks for Traffic Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, **1**, 485-492. https://doi.org/10.1609/aaai.v33i01.3301485

[10] Yan, H., Ma, X. and Pu, Z. (2021) Learning Dynamic and Hierarchical Traffic Spatiotemporal Features with Transformer. *IEEE Transactions on Intelligent Transportation Systems*, **23**, 22386-22399. https://doi.org/10.1109/TITS.2021.3102983

[11] Li, Y., Yu, R., Shahabi, C. and Liu, Y. (2017) Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting.

[12] Yu, B., Yin, H. and Zhu, Z. (2017) Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *Proceedings of the* 27*th International Joint Conference on Artificial Intelligence*, Stockholm, 13-19 July 2018, 3634-3640. https://doi.org/10.24963/ijcai.2018/505

[13] Wu, Z., Pan, S., Long, G., Jiang, J. and Zhang, C. (2019) Graph Wavenet for Deep Spatial-Temporal Graph Modeling. *Proceedings of the* 28*th International Joint Conference on Artificial Intelligence Main Track*, Macao, 10-16 August 2019, 1907-1913. https://doi.org/10.24963/ijcai.2019/264

[14] Bai, L., Yao, L., Li, C., Wang, X. and Wang, C. (2020) Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. *NeurIPS* 2020, 6-12 December 2020, 17804-17815.

[15] Ta, X., Liu, Z., Hu, X., Yu, L., Sun, L. and Du, B. (2022) Adaptive Spatio-Temporal Graph Neural Network for Traffic Forecasting. *Knowledge-Based Systems*, **242**, Article ID: 108199. https://doi.org/10.1016/j.knosys.2022.108199

[16] Zhang, J., Wang, F.Y., Wang, K., Lin, W.H., Xu, X. and Chen, C. (2011) Data-Driven Intelligent Transportation Systems: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, **12**, 1624-1639. https://doi.org/10.1109/TITS.2011.2158001

[17] Xie, Y., Zhang, Y. and Ye, Z. (2007) Short-Term Traffic Volume Forecasting Using Kalman Filter with Discrete Wavelet Decomposition. *Computer-Aided Civil and Infrastructure Engineering*, **22**, 326-334. https://doi.org/10.1111/j.1467-8667.2007.00489.x

[18] Goodfellow, I., Bengio, Y. and Courville, A. (2016) Deep Learning. MIT Press, Cambridge.

[19] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. and Philip, S.Y. (2020) A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, **32**, 4-24. https://doi.org/10.1109/TNNLS.2020.2978386

[20] Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T. and Li, H. (2019) T-gcn: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems*, **21**, 3848-3858. https://doi.org/10.1109/TITS.2019.2935152