Intelligent Decision Technologies -1 (2024) 1–18 1
~~DOI 10.3233/IDT-240690~~
IOS Press
UNCORRECTED PROOF

A spatiotemporal transfer learning framework with mixture of experts for traffic flow prediction

Junxiu Chen[a] and Weican Xie[b,*]
[a]*The Higher Educational Key Laboratory for Flexible Manufacturing Equipment Integration of Fujian Province (Xiamen Institute of Technology), Xiamen, Fujian, China*
[b]*Xiamen Planning Digital Technology Research Center, Xiamen, Fujian, China*

Received 9 May 2024
Accepted 19 August 2024

Abstract. For traffic management entities, the ability to forecast traffic patterns is crucial to their suite of advanced decision-making solutions. The inherent unpredictability of network traffic makes it challenging to develop a robust predictive model. For this reason, by leveraging a spatiotemporal graph transformer equipped with an array of specialized experts, ensuring more reliable and agile outcomes. In this method, utilizing Louvain algorithm alongside a temporal segmentation approach partition the overarching spatial graph structure of traffic networks into a series of localized spatio-temporal graph subgraphs. Then, multiple expert models are obtained by pre-training each subgraph data using a spatio-temporal synchronous graph transformer. Finally, each expert model is fused in a fine-tuning way to obtain the final predicted value, which ensures the reliability of its forecasts while reducing computational time, demonstrating superior predictive capabilities compared to other state-of-the-art models. Results from simulation experiments on real datasets from PeMS validate its enhanced performance metrics.

Keywords: Traffic flow prediction, intelligent decision technologies, louvain algorithm, expert models, fine-tuning

1. Introduction

Given its fundamental part in people's daily activities, transportation also exerts a substantial influence on environmental conditions [1]. As the count of cars and drivers has swelled, so too have the problems of traffic congestion and safety on our streets become increasingly severe. To solve this problem, many countries are committed to developing intelligent transportation systems (ITS) to achieve efficient traffic management [1]. Traffic control and guidance are the keys to the ITS, and traffic prediction is the prerequisite of scientific management and control [2]. However, traffic network data has strong temporal and spatial correlation and nonlinearity, which brings challenges to the establishment of accurate traffic prediction models.

With the deepening of research on traffic prediction algorithms, researchers have proposed plenty of high-performance prediction models, the algorithms of deep neural networks, which can mine complex nonlinear relationships between data from a large amount of historical data, thereby achieving higher prediction accuracy and stronger generalization ability [3,4]. For instance, Yu et al. [5] characterized the

*Corresponding author: Weican Xie, Xiamen Planning Digital Technology Research Center, Xiamen, Fujian, 361000, China. E-mail: Xieweican2000@126.com.

traffic and speed data of the traffic network into a static image, and then captures the spatio-temporal correlation through the spatio-temporal loop convolutional network, and verifies its superior performance on a traffic network in Beijing. Wu et al. [6] introduced an advanced predictive model for traffic flow that integrates various deep learning techniques. The model harnesses the power of Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs) to explore the intricate spatial and temporal dimensions of traffic data. It synthesizes historical traffic metrics, including velocities and traffic volumes, with the aid of attention mechanisms, effectively highlighting the DNN-BTF model's capacity to tackle predictive challenge. Yang et al. [7] put forth an advanced LSTM framework, crafted to elevate the performance of traffic flow forecast methodologie, which mines extremely long distance temporal correlations through attention, effectively improving the memory ability of the LSTM model. Yang et al. [8] introduced a ranking system based on ideal solution similarity to differentiate road segments into distinct categories. Following this, they employed convolutional LSTM networks for spatiotemporal data mining of pivotal road segments, which allows for the accurate prediction of their diverse states. Zhang et al. [9] have crafted a specialized CNN for anticipating short-term traffic patterns by conducting an analysis of the data's spatio-temporal progression. The system selects pertinent features through CNN-based mining, thereby boosting the predictive power of the forecasting model. Zhao et al. [10] employed hierarchical clustering to segment traffic flow data into distinct groups, followed by an analysis of spatial correlations among road networks and segments within these groups using the conventional Euclidean space framework. By pinpointing the top-k most relevant road segment data strongly associated with the segment of interest, the LSTM is fed features that boost its forecasting precision. X Zhang and Q Zhang. [11] fused the predictive capabilities of LSTM networks with the robustness of XBoost's ensemble learning to focus on estimating forthcoming traffic volumes, thereby circumventing the overfitting tendency inherent in LSTMs and bolstering the models' predictive performance across various scenarios. Cai et al. [12] have utilized the correlation entropy as a robust loss function for LSTM, aimed at mitigating the impact of non-Gaussian noise on short-term traffic flow predictions and improving the model's noise immunity. Xia et al. [13] combined distributed modeling frameworks with LSTM networks to solve the problem of difficulty in training and using models caused by large traffic data, improving the efficiency and usability of projecting near-future traffic patterns. Zhang and Jiao [14] implemented a gated convolutional module with an array of kernel sizes to unearth the temporal and spatial interdependencies in historical traffic datasets. They also crafted an attention mechanism that incrementally augments the model's width to assign importance to key hidden features, which maintains high accuracy with a relatively low computational cost. Fang et al. [15] enhanced their LSTM model for predicting short-term traffic flows by embedding an attention mechanism. This addition enables the model to discern and emphasize key informational inputs, leading to more accurate predictive outcomes

Standard algorithms for convolutional and recurrent neural networks are designed for data within Euclidean domains and are not suitable for the graph-based data from complex traffic networks that exist in non-Euclidean spaces. Graph Neural Networks [16,17], however, can adeptly process this type of data by leveraging various aggregation methods to discern the relationships between nodes and extract underlying features. Their ability to represent the spatial connections within traffic networks makes them well-suited for data mining tasks in non-Euclidean contexts. For example, Yu et al. [18] crafted an STGCN for the purpose of traffic forecasting, leveraging the model's ability to capture spatial and temporal dependencies within traffic data. It mined the spatiotemporal correlation of road network information through stacking gated convolutional network and graph convolutional network structure, and it outperformed the ensemble CNN-RNN model in terms of forecasting accuracy, reflecting its enhanced predictive capabilities. Guo et al. [19] introduced an attention mechanism into the ASTGCN for the initial time to perform traffic flow predictions. They dissected spatio-temporal correlations through three unique temporal branches and employed attention to weigh the significance of hidden features across each branch's layers, which resulted in higher prediction accuracy. Zhao et al. [20] presented a novel neural network for traffic prediction that synergizes GCN with GRU within the T-GCN framework, adeptly seizing the evolving dynamics within traffic datasets and outperforming other advanced models. Bai et al. [21] designed a module that adaptively learns each spatial node and applied it to a graph convolutional recursive network to generate an adaptively learning graph convolutional framework(AGCRN) designed for anticipating traffic patterns, allowing the model to automatically capture different fine-grained traffic spatio-temporal correlations. Zheng et al. [22] crafted the GMAN framework, which incorporating an encoder-decoder approach, the model projects the evolution of traffic patterns over differing time spans. The model fuses spatial and temporal attention with a gating technique to enhance the significance of spatiotemporal embeddings, demonstrating effectiveness in long-term predictive tasks through real-data trials. Song et al. [23] developed a groundbreaking framework known as the STSGCN, designed to address the complexities of spatial-temporal dynamics in traffic flow prediction through a synchronized graph convolutional approach, thereby markedly enhancing predictive precision over methods that analyze these correlations asynchronously. Wang et al. [24] Unveiled an innovative strategy employing a multi-graph adversarial neural network for the autonomous detection of spatial-temporal features in traffic data. This technique allows for the real-time extraction of these states and the subsequent generation of traffic forecasts constrained by the GAN framework. Yin et al. [25] introduced an innovative traffic forecasting framework known as the MASTGN. The model adopted encoder-decoder structure and mixed spatial attention. The three forms of attention, internal attention and temporal attention, integrate hidden features from different angles and achieve a very high accuracy. Zhang et al. [26] crafted a unique Spatiotemporal Graph Attention Network for forecasting traffic flow, capable of unearthing both global and local spatial interactions and incorporating various levels of temporal dynamics. Moreover, By tapping into the traffic data's semantic nuances, it secures remarkable outcomes in predictive analytics. Li et al. [27] have engineered a pioneering model for understanding the spatial-temporal patterns present in traffic data, adeptly visualizing the temporal and spatial features, fully harnessing the natural connections of time and space, and markedly improving the accuracy of traffic flow forecasts. Na et al. [28] developed an adaptive approach for computing adjacency matrices that, in conjunction with graph convolutional networks, adeptly uncovers the temporal variations in spatial relationships of road networks. It outperforms the conventional fixed-matrix methods for local hidden feature aggregation in terms of both accuracy and adaptability. Ni and Zhang [29] employed a multi-graph framework to depict the transportation network, then uses an interpretable spatiotemporal graph convolutional network (STGMN) for hidden feature information mining, and Elevated the network's depth by stacking additional layers within a residual framework, which prediction results have advantages compared to the advanced models previously proposed. Yin et al. [30] combined spatiotemporal graph neural network and transfer learning to mine spatiotemporal traffic patterns of specific nodes, and introduces clustering mechanism to elevate the predictive capabilities for the intended outcome. Jin et al. [31] designed a transformative traffic prediction model known as Trafformer, which combines spatial and temporal insights into a singular transformer model, adept at uncovering complex dependencies across space and time. Yu et al. [32] took into account the diverse spatiotemporal dynamics in traffic forecasting by employing a causally-driven spatiotemporal synchronous graph convolutional network to uncover spatial-temporal relationships, which led to superior predictive outcomes. Chen et al. [33] derived adjacency matrices from traffic flow data, leveraging the power of attention mechanisms, they constructed a transformer encoder in tandem with graph convolutional networks to act as a proficient feature extractor for traffic's spatial-temporal correlations, augmenting the model's forecasting efficacy. Liu [34] combines SAE, GCN, and BiLSTM to predict the passenger flow of urban rail transit, and evaluates it through real data at different granularities, proving its high accuracy and good robustness.

Despite the applicability of existing forecasting models to data from complex traffic networks, there remains a need to address issues related to increasing the accuracy of calculations and decreasing the duration of the computation process. These mainly contain three parts. 1) Creating a localized spatiotemporal graph allows for a more nuanced representation of the intricate spatial and temporal dynamics within traffic data, but the number of nodes in each local spatiotemporal graph has multiplied than the original graph, resulting in a significant increase in the calculation time. 2) Traffic monitoring sensors can detect and record various indicators of traffic conditions, encompassing flow, occupancy, and speed.. How to effectively use this information's spatiotemporal dependence to enhance the precision of the predictive model is of utmost importance. 3) When leveraging a graph neural network for the concurrent extraction of temporal and spatial correlations, it is essential to account for the ancillary data among nodes across time and space to accurately aggregate their latent representations. To solve these problems, the current research designs a spatiotemporal synchronization graph transformer with mixture of experts (MOE-STSGFomer) for anticipating traffic flow. The innovative points of this research include:

Firstly, by combining Louvain algorithm with local time sliding window, traffic network data set is divided into several local time-gap subgraph data sets. Then, each subset is pre-trained to obtain several expert models, and then these expert models are migrated and the expert gated network is fine-tuned to obtain the prediction model of the entire road network map, which can effectively reduce the prediction time while ensuring a high prediction accuracy.

Secondly, the graph Transformer network is used in each expert model, only encoder structure is used in the network, and the self-attention multi-head structure in the graph Transformer is replaced by trainable edge information, so that both node information and edge information are considered when extracting spatiotemporal correlation synchronously. The model can more fully and accurately express and Leverage the traffic network's dynamic interplay of space and time

Finally, the current research uses two real datasets on PeMS for simulation experiments, and the experimental outcomes unequivocally show that our model's forecasting capabilities surpass those of current state-of-the-art predictive models

2. Preliminary

Envisioning traffic flow forecasting as the anticipation of future sequences, each influenced by multiple variables. These data come from multiple traffic nodes on the road network. Under the assumption, X_t symbolizes the features of nodes at time t, and X_t^f stands for the collective traffic flow properties of the nodes at that instant. The objective of forecasting traffic flow is to learn a complex nonlinear formula through historical traffic data to estimate future traffic flow over a specified period, as follows:

$$\left(X_{t+1}^f, \ldots, X_{t+\tau_1}^f\right) = F\left[\left(X_{t-\tau_2+1}, \ldots, X_t\right)\right] \tag{1}$$

In addition, we have defined some of the concepts used in the method, as shown below. Traffic network data can be represented by an undirectedgraph $G = (V, E)$ structure, where $V \in R^N$ represents the set of nodes (all sensors) and E represents the set of edges (connecting edges between sensors). Whether there is a link edge between nodes is expressed by the critical matrix $A \in R^{N \times N}$ Setting $A_{i,j} = 1$ to 1 creates an edge between node i and node j; setting it to 0 eliminates any such link.

3. Methodology

To ensure high prediction accuracy and solve the problem that training the model presents considerable difficulties by using local space-time graph for feature extraction, this paper designed the MOE-STSGFormer method for short-term traffic forecasting tasks. Figure 1 illustrates that the technique is fundamentally made up of several stages: Construct local spatio-temporal subgraphs, Pre-training and

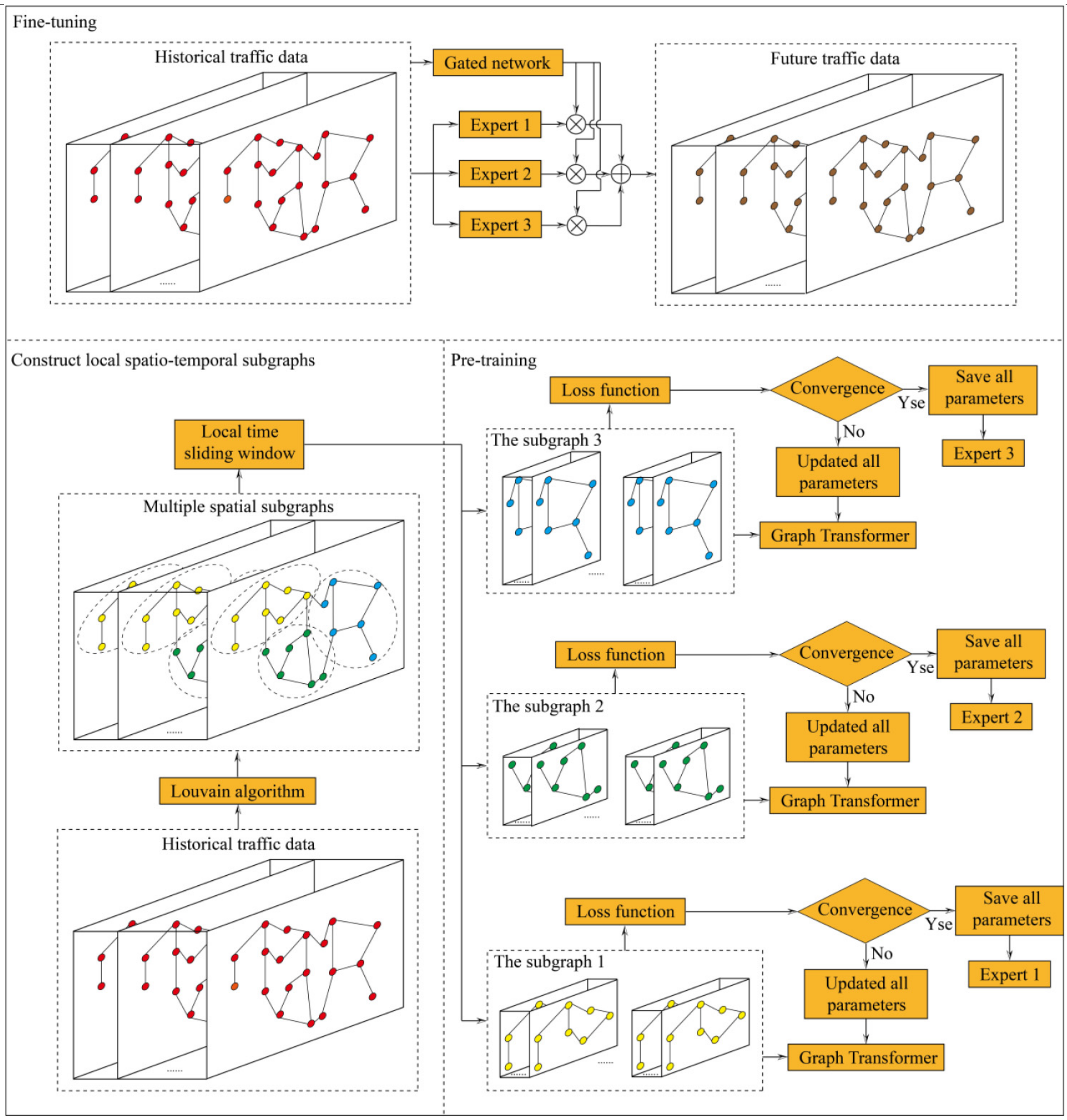


Fig. 1. The structure of MOE-STSGFormer.

Fine-tuning. Firstly, Louvain algorithm and local time sliding window are combined to reconstruct the historical input features into multiple local time-gap subgraphs. Then, the transformer network is used for pre-training and each model is saved and defined as an expert model. Finally, the final predicted value is obtained by combining all the fixed parameter expert models and fine-tuned gated network to train the historical input features. The framework of this model is described in detail below.

3.1. Construct local spatio-temporal subgraphs

To segment the optimal set of subgraph structures, this paper first quotes a general standard for evaluating the rationality of community segmentation: modularity. The principle is the difference between the module cohesion of certain segmentation results and the cohesion of random segmentation results. The calculation process is as follows:

$$Q = \sum_{C} \left[\frac{\sum in}{2m} - \gamma \left(\frac{\sum tot}{2m} \right)^2 \right] \tag{2}$$

where Q is modularity. C is the total number of segmented subgraphs. $\sum in$ and $\sum tot$ tot are the sums of weights of edges and edges connected to nodes in the subgraph, respectively. m is the sum of the weights of all edges. γ is the resolution. The higher it is, the more communities are segmented; the lower it is, the less communities are segmented.

Louvain algorithm [35] is an algorithm based on modularity to search for optimal community segmentation. The algorithm first sets the resolution, selects the interval $[0, \gamma_{\max}]$ and the sampling interval s (s can be divisible by $\gamma_{\max}$), then the set of modularity resolution that can be selected is, and then calculates the subgraph segmentation set of the maximum modularity under each resolution. The specific process is as follows:

1) Each node in the network is assigned a different number so that there are subgraphs with the same number of vertices in the initial subgraph segmentation.
2) Add node i to the subgraph c of its neighbor node j in turn to calculate the overall modularity gain. The community modularity after node joining is as follows:
$$Q^{c}_{add} = \frac{\sum in + k_{i,in}}{2m} - \gamma \left(\frac{\sum tot + k_i}{2m} \right)^2 \tag{3}$$
where $k_{i,in}$ is defined as the cumulative weight connected by node i to subgraph c and k_i is indicative of the degree of node i. There is only one node in subgraph c' before node i is moved, then the modularity of subgraph c' before node i is removed:
$$Q^{c'} = 0 - \gamma \left(\frac{k_i}{2m} \right)^2 \tag{4}$$
The modularity of community c' after node i moving out is:
$$Q^{c'}_{rem} = 0 \tag{5}$$
Then, the modularity gain obtained is:
$$\Delta Q = \left(Q^{c}_{add} - Q^{c}\right) + \left(Q^{c'}_{rem} - Q^{c'}\right) = \frac{k_{i,in}}{2m} - \gamma \frac{k_i \sum tot}{2m^2} \tag{6}$$
3) Add each node to the subgraph whose modularity gain is greater than 0 and has the maximum modularity gain. If the modularity gain calculated by the surrounding subgraphs is less than 0, the current node is not added to any subgraph.
4) The results obtained in the previous step are reconstructed. Each subgraph is merged again, and the original graph is converted into a new hypergraph. It can be considered that the new subgraph is a large node, and the edge weight between these two significant nodes is the cumulative weight of the edges that interconnect all nodes across both subgraphs. After constructing the new hypergraph, the modularity transformation is iteratively calculated again.
5) After repeating steps 2–4 repeatedly, stop the algorithm until the overall modularity no longer changes or the predefined iteration count is met.

Louvain algorithm decomposes the spatial graph structure of historical traffic data into multiple subgraph structures. Utilizing a local time sliding window, the subgraph configuration for every historical traffic dataset is reconstructed. Assume that the q-th subgraph G^q, has an input feature identified by

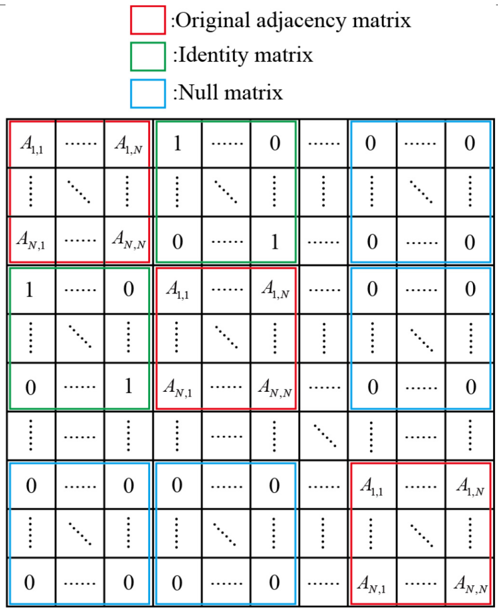


Fig. 2. The new adjacency matrix.

$\left[X^q_{t-\tau_2+1}, \ldots, X^q_t\right]$, and the number of time channels of the time sliding window used for feature reconstruction is τ_3, then the input feature after reconstruction is:

$$\left\{\begin{array}{c} \left[X^q_{t-\tau_2+1}\middle\|\cdots\middle\|X^q_{t-\tau_2+\tau_3}\right], \\ \left[X^q_{t-\tau_2}\middle\|\cdots\middle\|X^q_{t-\tau_2+\tau_3-1}\right], \\ \vdots \\ \left[X^q_{t-\tau_3+1}\middle\|\cdots\middle\|X^q_{t}\right] \end{array}\right\} \tag{7}$$

Considering N^q as the count of initial input feature nodes, the reconstructed model yields $\tau_3 N^q$ nodes. After reconstruction, the new adjacency matrix represents each channel's graph structure connection mode, as shown in Fig. 2. It can be seen that it is composed of the original adjacency matrix, the identity matrix, and the zero matrix, and its dimension is $\tau_3 N^q \times \tau_3 N^q$.

3.2. *Pre-training*

The graph transformer network uses a stacked graph self-attention network (GSA) for data mining. Figure 3 displays the structure of a one-layer graph self-attention network, which calculates the spatio-temporal dependence between any two locations through the linear transformation of the three branches and allows the model to more effectively seize the comprehensive details of historical data.

With H^l as the input feature for the node at the lth layer, it is a composite of the node's input feature and the position encoding in the first layer. Position encoding is usually in the form of trigonometric functions:

$$P^l_{i,j} = \begin{cases} \sin\left(\frac{j}{10000^{\frac{i}{n}}}\right), i \in odd \\ \cos\left(\frac{j}{10000^{\frac{i}{n}}}\right), i \in even \end{cases} \tag{8}$$

where $P^l_{i,j}$ is the position coder feature, i and j are the indexes of the reconstructed input feature nodes and time channels. The specific calculation process of Query, Key, and Value for self-attention is as follows:

$$\begin{cases} Q^l = \left(H^l + P^l\right) W^l_q \\ K^l = \left(H^l + P^l\right) W^l_k \\ V^l = \left(H^l + P^l\right) W^l_v \end{cases} \tag{9}$$

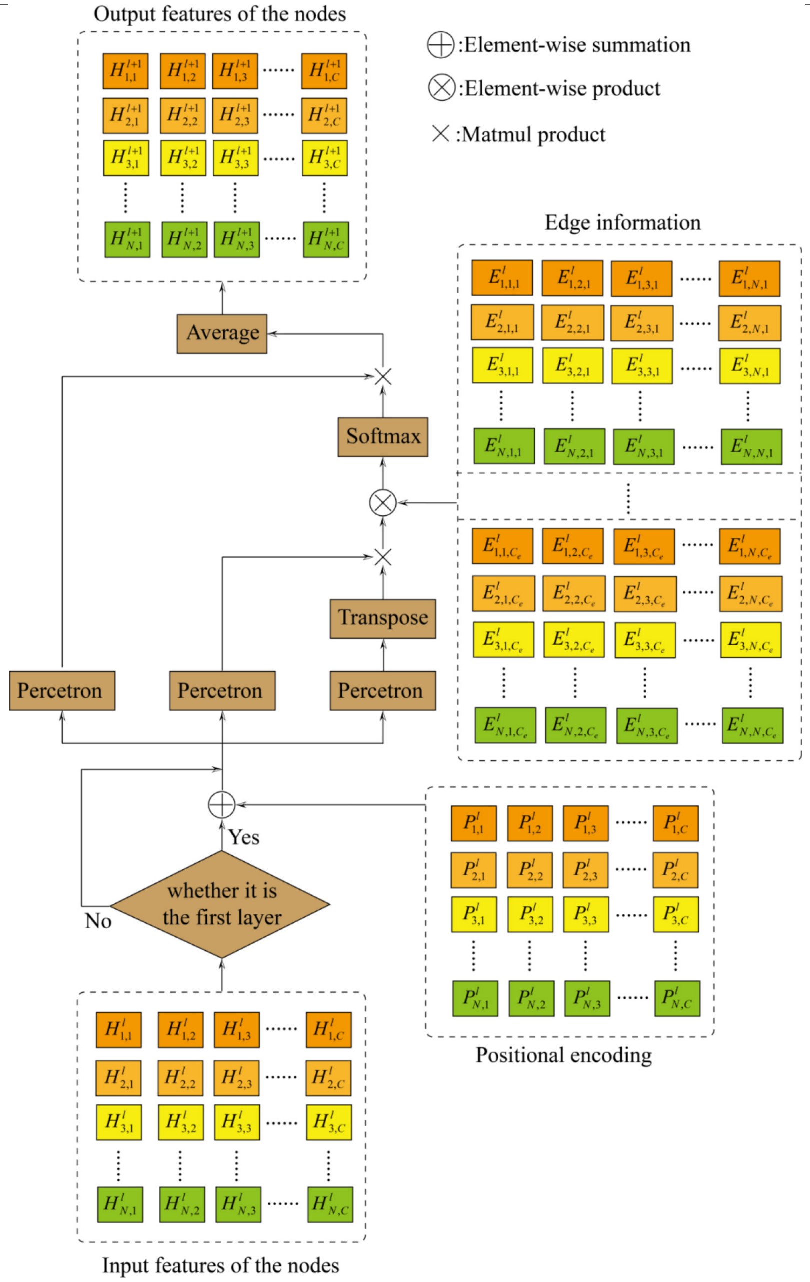


Fig. 3. The structure of GSA.

where Q^l, K^l and V^l are respectively the Query of the first layer, Key and Value, and W_q^l, W_k^l and W_v^l are respectively the weights of the three perceptrons of the first layer.

If it is not in the first layer, the input feature is only node input features. The specific calculation process of Query, Key, and Value of self-attention is as follows:

$$\begin{cases} Q^l = H^l W_q^l \\ K^l = H^l W_k^l \\ V^l = H^l W_v^l \end{cases} \tag{10}$$

The correlation Z^l between each vector is obtained by calculating the dot product of each vector in Query with each vector in Key:

$$Z^l = Q^l \times \left(K^l\right)^T \tag{11}$$

Then, correlation Z^l and edge information E^l are multiplied by corresponding positions to obtain a vector correlation matrix α^l with edge information, which Softmax normalizes to make its gradient stable during training:

$$\alpha^l = Soft\max\left(E^l \otimes Z^l\right) \tag{12}$$

where α^l is the normalized vector correlation matrix with edge information. $E^l \in R^{N\times N\times C_e}$ is the edge information. C_e is the channel number of edge information. The edge information of each layer is obtained by multiplying the trainable channel weight W^l with the adjacency matrix A^q of the local space-time graph:

$$E^l = W^l A^q \tag{13}$$

Finally, the vector features of all nodes in the next layer are obtained by producting of A^l and V^l for each channel:

$$H^{l+1} = A^l \times V^l \tag{14}$$

After the transfomer prediction model corresponding to the subgraph is created through the above process, the transfomer prediction model is trained using MSE as a loss function and Adam as a parametric updated optimization algorithm. The trained parameters are then saved. Each trained model will undergo subsequent transfer learning as an expert model.

3.3. Transfer learning and fine-tuning

Transfer learning puts entire historical traffic data as input features into each trained expert model, and then weights the output features of each expert model through a gated network. Training the gated network represents a fine-tuning process. Finally, all the weighted output features are summed to arrive at the ultimate forecasted outcome.

Within the gated network, there are two layers of full connectivity. The top layer reduces the number of temporal channels in the input features to unity by linear mapping. The bottom layer, in turn, decreases the node count of the input features to equate with the domain expert model count through another linear mapping. The exact calculation process is detailed hereafter:

$$H^G = \sigma\left(W_2 X W_1\right) \tag{15}$$

Where H^G is the output sequence of the gated network, W_1 and W_2 represent the weights of a dual-layer fully-connected network σ is the Softmax function.

4. Empirical evaluation

The complete simulation experiment was conducted utilizing a computer equipped with an RTX 2080Ti GPU and the model was crafted using the open-source PyTorch framework.

4.1. Data description

For the simulation aspects of this paper, we have employed two datasets that are publicly accessible through PeMS:

– The PeMSD4 dataset is derived from 307 traffic sensors along 29 Bay Area roads in San Francisco, recorded over a 59-day period from January 1, 2018, to February 28, 2018. The training data includes 52 days, extending to February 21, 2018, and the test data comprises the last seven days of this period, ending on February 28, 2018.
– The PeMSD8 dataset is derived from 170 traffic sensors along 8 San Bernardino Area roads, recorded over a 61-day period from July 1, 2016, to August 31 2016. The training data includes 54 days, extending to August 25, 2016, and the test data comprises the last seven days of this period, ending on August 31, 2016.
– This paper mainly uses k-Nearest Neighbor [35] to interpolate missing data.

4.2. Experimental parameter settings

Multiple training and verification tests were executed to pinpoint the most efficient parameters for the MOE-STSGFormer model, which are as follows: (1) The duration of the historical time window for input features is one hour, while the prediction horizon varies from 5 to 45 minutes. The time window for feature reconstruction is set at 15 minutes, with each temporal data point spaced 5 minutes apart, $\tau_2 = 12 \quad \tau_1 \in \{1, 2, \ldots, 9\}$ and $\tau_3 = 3$. (2) The channel number of edge information C_e is allocated the value of 2, (3) the batch size per sample is 32 during the iterative optimization cycle, with a learning rate of 1e-4.

4.3. Subgraphs segmentation result

Utilizing the dataset's original adjacency matrix as a foundation, Louvain algorithm is used to segment the whole graph structure, and samples are collected within the range of 0$\sim$1.5 with a sampling interval of 0.01. The optimal modularity value under different resolutions is shown in Fig. 4.

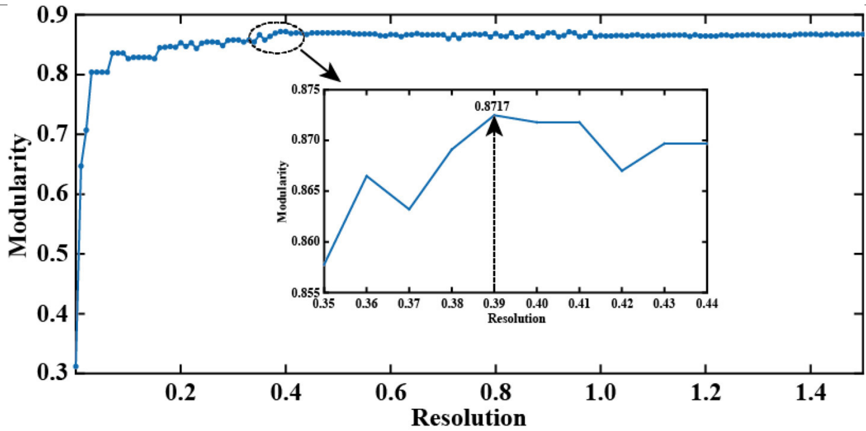
It can be seen that when the resolution is 0.39, the optimal modularity of PeMSD4 data set is obtained. In other words, at the 39th sampling, the optimal modularity value of the subgraph segmentation by Louvain algorithm is the largest, which is 0.8717. When the resolution is 0.61, the optimal modularity of PeMSD8 data is obtained, that is, at the 61th sampling, the optimal modularity value of the subgraph segmentation by Louvain algorithm is the largest, which is 0.7473. Through this process, 23 subgraphs can be generated from PeMSD4 data and 12 subgraphs can be generated from PeMSD8 data.

4.4. Baseline models

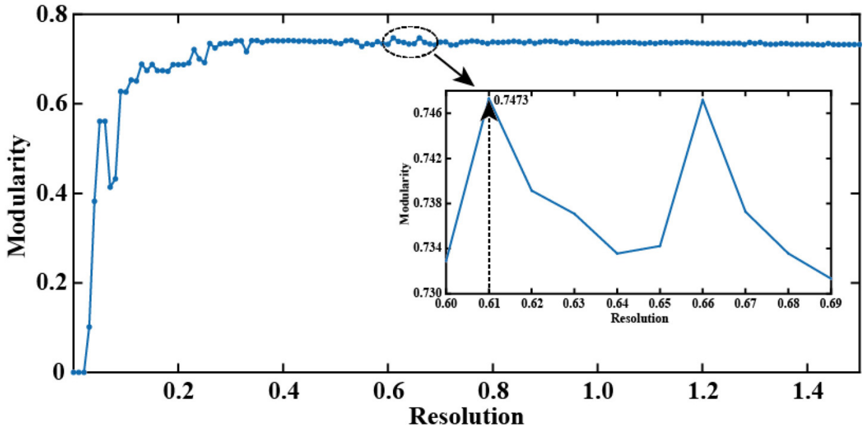
To establish the superiority of our model, we will benchmark it against seven advanced baseline models: LSTM, GCN, STGCN, ASTGCN, STSGCN, STGMN, and Trafformer. The LSTM model is designed with a 5-layer setup, and the GCN model shares an equivalent structure with the STGCN model. Other baseline models are configured according to the descriptions provided in the references.

4.5. Performance superiority analysis

To begin with, an assessment of the precision of each predictive model is undertaken. Error metrics including Mean Absolute Error (*MAE*), Root Mean Square Error (*RMSE*), and the Coefficient of



(a) The Optimal Modularity of PeMSD4



(b) The Optimal Modularity of PeMSD8

Fig. 4. The optimal modularity at different resolutions.

Determination (R^2) are applied:

$$MAE = \frac{1}{NT}\sum_{i=1}^{N}\sum_{j=1}^{T}\left|\hat{y}_{i,j} - y_{i,j}\right| \tag{16}$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{N}\sum_{j=1}^{T}\left(\hat{y}_{i,j} - y_{i,j}\right)^2} \tag{17}$$

$$R^2 = 1 - \frac{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{T}\left(\hat{y}_{i,j} - y_{i,j}\right)^2}{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{T}\left(\bar{y}_{i,j} - y_{i,j}\right)^2} \tag{18}$$

where T is the number of channels in the time dimension of the test set, $\hat{y}_{i,j}, y_{i,j}$ and $\bar{y}_{i,j}$ are the predicted values of the model, the true values of the samples and the average of the true values of the samples during testing. *MAE* and *RMSE* gauge model error, with lower figures suggesting enhanced accuracy. On the other hand, R^2 measures the model's predictive similarity, where higher values imply greater precision.

Table 1
Three evaluation metrics of different prediction models on two data sets

Model	PeMSD4			PeMSD8		
	MAE	*RMSE*	R^2	*MAE*	*RMSE*	R^2
LSTM	20.2125	30.7477	0.9633	15.9882	23.4225	0.9748
GCN	21.7381	33.1505	0.9573	16.7401	24.7421	0.9718
STGCN	20.1238	30.2878	0.9644	16.4426	23.9907	0.9735
ASTGCN	19.3602	29.2162	0.9669	14.4933	21.3635	0.9790
STSGCN	16.8577	24.5555	0.9766	13.0255	19.1288	0.9832
STGMN	16.7102	24.9426	0.9742	13.4974	19.7372	0.9815
Trafformer	14.3063	21.4115	0.9813	11.2123	17.2561	0.9877
Ours	11.0181	17.8011	0.9884	8.8089	14.6822	0.9910

Table 2
Calculation times of different prediction models

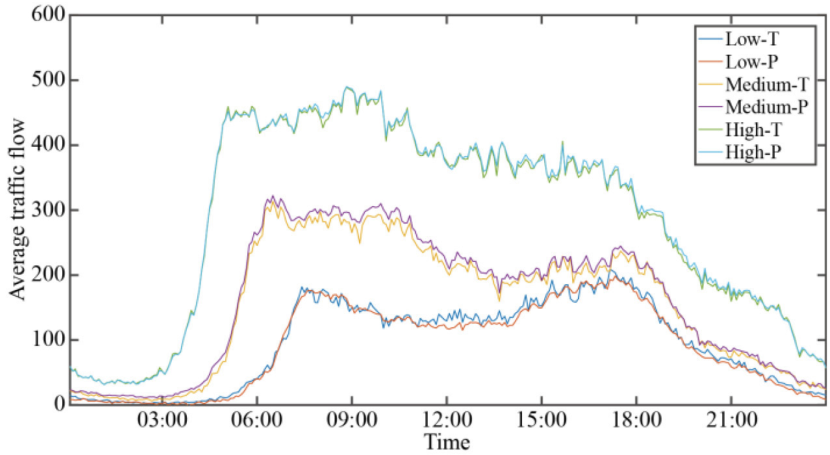
Model	PeMSD4		PeMSD8	
	$T_1\ (s/epoch)$	$T_2\ (s)$	$T_1\ (s/epoch)$	$T_2\ (s)$
LSTM	9.7906	0.9844	7.5634	0.6241
GCN	7.0781	0.7539	5.0342	0.5347
STGCN	9.6648	0.8627	7.7081	0.5365
ASTGCN	29.1571	1.5873	14.0454	0.9862
STSGCN	49.6465	4.6824	29.5872	2.1067
STGMN	22.3285	1.1249	11.7024	0.8746
Trafformer	47.4365	4.3337	24.2158	1.8735
Ours	20.4296	1.0224	9.9852	0.7674

Table 1 illustrates the performance of various models as measured by *MAE*, *RMSE*, and R^2 on the two datasets. The results are obtained when the prediction horizon time length is 5min. which can be found *MAE* and *RMSE* of LSTM and GCN are the highest and R^2 is the lowest. While LSTM focuses solely on the temporal relationships within historical data, GCN concentrates on spatial relationships, leading to diminished predictive precision. Trafformer and STSGCN can synchronously mine the spatiotemporal correlation of historical data, with lower *MAE* and *RMSE* and higher R^2 compared to other baseline models. This indicates that models that synchronously mine the spatiotemporal correlation of historical data have higher prediction accuracy than those that asynchronously mine the spatiotemporal correlation of historical data. The MOE-STSGFormer model designed by us has the lowest *MAE* and *RMSE* and the highest R^2, compared with the baseline model with the best effect, *MAE* and *RMSE* are reduced by 22.98% and 16.86%, and R^2 is increased by 0.71% in PeMS04 data set; compared with the best baseline model, *MAE* and *RMSE* were reduced by 21.44% and 14.92%, and R^2 was improved by 0.33% in the PeMS08 dataset. This approach yields superior predictive accuracy in comparison to alternative baseline models.

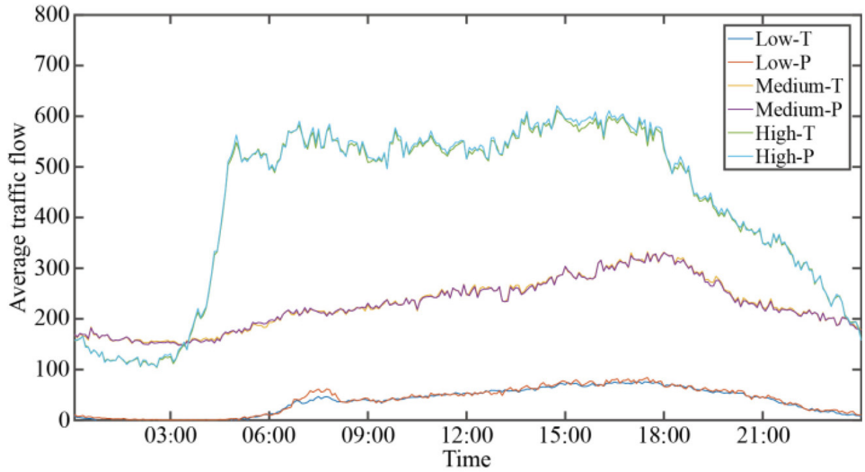
The time required for model training and testing is also a significant metric in evaluating the model's effectiveness. Table 2 shows the calculation time of our designed model and all baseline models, where T_1 is the time required for a single epoch to train the model, and T_2 is the total time required to test the model.

Referencing Table 4, it is evident that the time taken for our model to perform calculations is more than what is needed for LSTM, GCN, and STGCN models, because these three models are simple in structure and sacrifice the prediction accuracy. When pitted against the STSGCN and Trafformer models, our model boasts a lower time frame for processing predictions, which indicates that the model designed by us solves the problem of increasing the prediction time caused by constructing local spatiotemporal graph for synchronous spatiotemporal correlation mining.

A plethora of spatial nodes exists for traffic data, with the potential for heterogeneity among them. To verify that the prediction model designed by us can have higher prediction accuracy on different types of spatial nodes, the predicted value of high traffic flow, medium traffic flow and low traffic flow are selected



(a) The Visualization Results of PeMSD4



(b) The Visualization Results of PeMSD8

Fig. 5. Visualization of true and predicted traffic flow values in different traffic patterns.

to compare with the real value. The diagram in Fig. 5 visually represents how the MOE-STSGFormer model can adapt to traffic flow datasets with diverse traffic modes, ranging from high to low.

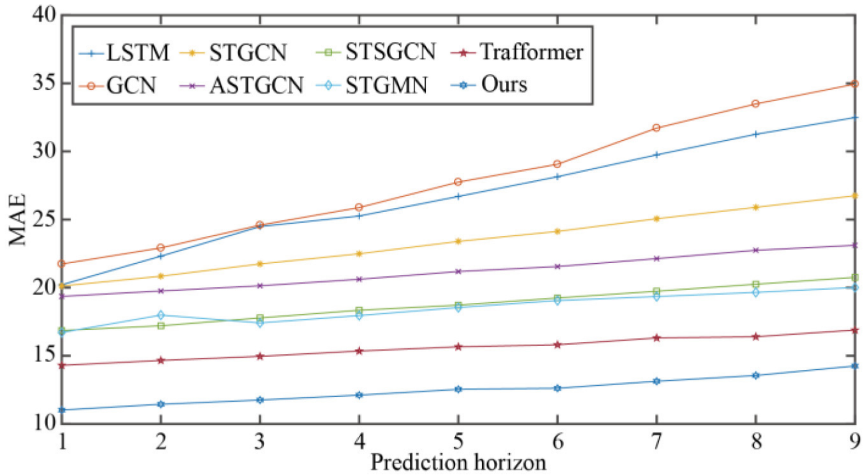
The prediction performance assessments mentioned previously were conducted under the condition that the prediction horizon equals 1. This paper verify that MOE-STSGFormer also has good prediction accuracy in other prediction horizons, the model was compared with other baseline *MAE* models in the two datasets when the prediction horizon is 1–9, which is 5–45 minutes. Figure 6 illustrates the outcomes of our MOE-STSGFormer model, which were observed with a prediction horizon extending from 1 to 9 across two different datasets. When juxtaposed with baseline models, our MOE-STSGFormer model shows the lowest performance metrics, highlighting its ability to sustain optimal prediction accuracy under diverse prediction horizons.

4.6. *Verification of edge information performance*

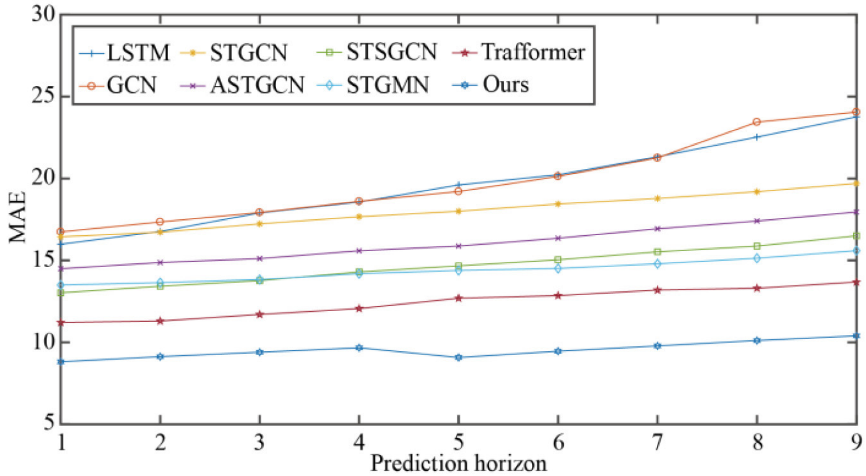
The variable C_e, indicating the quantity of edge information channels, is essential for the model's predictive accuracy. To select the optimal edge information channels of the model, By keeping other parameters

Table 3
Evaluation metrics of prediction models with different number of edge information channels

Model	PeMSD4			PeMSD8		
	MAE	*RMSE*	R^2	*MAE*	*RMSE*	R^2
$C_e = 1$	13.7162	20.9425	0.9837	10.9584	16.8416	0.9725
$C_e = 2$	11.0181	17.8011	0.9884	8.8089	14.6822	0.9910
$C_e = 3$	11.3342	18.0546	0.9856	9.5421	15.2158	0.9845



(a) The M AE of All Prediction Models on PeMSD4



(b) The M AE of All Prediction Models on PeMSD8

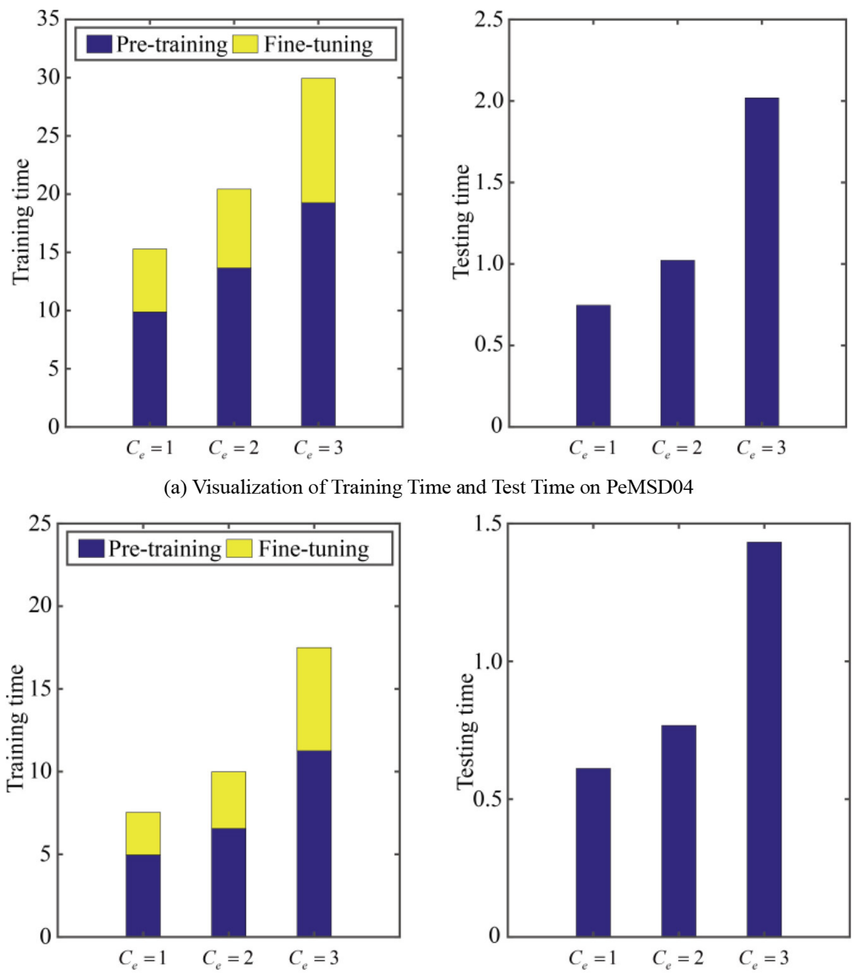
Fig. 6. The MAE of all prediction models in different prediction horizons.

stable and altering the edge information channels, we evaluated the model's prediction capabilities. The corresponding error indicators and processing times are detailed in Table 3 and depicted in Fig. 7.

As observed in Table 3, when C_e changes from 1 to 2, the errors in the two data sets will become smaller, that is, the prediction accuracy will increase, but when C_e changes to 3, the error will increase. It signifies that an overabundance of edge information channels could lead to overfitting, which consequently impairs the model's accuracy in forecasting. Figure 7 illustrates that an escalation in the count of edge information

Table 4
Five evaluation metrics of different prediction models on two data sets

Model	PeMSD4		PeMSD8	
	STSGFormer	MOE-STSGFormer	STSGFormer	MOE-STSGFormer
MAE	10.9954	11.0181	8.7542	8.8089
RMSE	17.9216	17.8011	14.5465	14.6822
R^2	0.9883	0.9884	0.9923	0.9910
$T_1\ (s/epoch)$	38.2519	20.4296	18.5796	9.9852
$T_2\ (s)$	3.5487	1.0224	1.8741	0.7674



(a) Visualization of Training Time and Test Time on PeMSD04

(b) Visualization of Training Time and Test Time on PeMSD08

Fig. 7. Visualization of training time and test time.

channels correlates with a progressive rise in the model's pre-training, fine-tuning, and testing durations. Hence, to strike a balance between predictive accuracy and computational efficiency, this study opts for two edge information channels.

4.7. *Verification of mixture expert models*

To verify that obtaining the final predictive model through pretraining multiple expert models and fine-tuning the gating system can solve the problem of difficult training of predictive models, this paper compares the predictive performance of the spatiotemporal synchronous graph transformer model (STSGFormer) trained on the entire spatial graph data with the original model (MOE-STSGFormer), the outcomes from both datasets are detailed in Table 4.

The performance of MOE-STSGFormer and STSGFormer in terms of prediction accuracy is comparable for both datasets; however, MOE-STSGFormer is notably faster in computation. To encapsulate, the approach of initially pre-training multiple expert models followed by fine-tuning the gating mechanism ensures high predictive accuracy, while simultaneously simplifying the model to expedite its computation time.

5. Conclusion

In this paper, a traffic flow prediction model based on MOE-STSGFormer is proposed to solve the problem of high computing time and high hardware requirement when there are too many nodes in the traffic network. MOE-STSGFormer uses Louvain algorithm based on optimal modularity to divide the spatial graph structure of the whole traffic network into multiple sub-graphs, and then reconstructs the data of each subgraph by using time sliding window. Then, multiple expert models are obtained through pre-training, and finally, multiple expert models are fused through fine-tuning to obtain the final predicted value. The simulation results show that the proposed method has a high prediction accuracy, reducing the error by 15%–20% compared with the best baseline model, and the calculation time is much lower than other models for synchronous mining of spatio-temporal correlation, and it is easier to train and test. Moreover, it is proved by experiments that selecting the optimal number of edge information channels is conducive to improving the prediction performance of the model. In addition, it is also verified by experiments that adding Mixture Expert Models to the model can ensure the constant prediction accuracy while reducing a large amount of calculation time and calculation cost.

Conflict of interest

The authors declare no conflicts of interest.

Data availability

The data used to support the findings of this study are included within the article.

Funding

This research was supported by 2022 Fujian province young and middle-aged Teacher Education Research Project (Science and Technology category) (No. JAT220470), 2022 Xiamen Institute of Technology School-level Research Fund for young and middle-aged projects (No. KYT2022004), College of Computer Science and Information Engineering 2021 Academic level Research Fund Project (No. EEKY2021003).

References

[1] Zhang J, Wang FY, Wang K, Lin WH, Xu X, Chen C. Data-driven intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems. 2011 Jul 21; 12(4): 1624-39.
[2] Wang K, Ma C, Qiao Y, Lu X, Hao W, Dong S. A hybrid deep learning model with 1DCNN-LSTM-Attention networks for short-term traffic flow prediction. Physica A: Statistical Mechanics and its Applications. 2021 Dec 1; 583: 126293.
[3] Liu H, Li X, Gong W. Research on detection and recognition of traffic signs based on convolutional neural networks. International Journal of Swarm Intelligence Research (IJSIR). 2022 Jan 1; 13(1): 1-9.

[4] Zhao W, Mu G, Zhu Y, Xu L, Zhang D, Huang H. Research on electric load forecasting and user benefit maximization under demand-side response. International Journal of Swarm Intelligence Research (IJSIR). 2023 Jan 1; 14(1): 1-20.
[5] Yu H, Wu Z, Wang S, Wang Y, Ma X. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors. 2017 Jun 26; 17(7): 1501.
[6] Wu Y, Tan H, Qin L, Ran B, Jiang Z. A hybrid deep learning based traffic flow prediction method and its understanding. Transportation Research Part C: Emerging Technologies. 2018 May 1; 90: 166-80.
[7] Yang B, Sun S, Li J, Lin X, Tian Y. Traffic flow prediction using LSTM with feature enhancement. Neurocomputing. 2019 Mar 7; 332: 320-7.
[8] Yang G, Wang Y, Yu H, Ren Y, Xie J. Short-term traffic state prediction based on the spatiotemporal features of critical road sections. Sensors. 2018 Jul 14; 18(7): 2287.
[9] Zhang W, Yu Y, Qi Y, Shu F, Wang Y. Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning. Transportmetrica A: Transport Science. 2019 Nov 29; 15(2): 1688-711.
[10] Zhao L, Wang Q, Jin B, Ye C. Short-term traffic flow intensity prediction based on CHS-LSTM. Arabian Journal for Science and Engineering. 2020 Dec; 45: 10845-57.
[11] Zhang X, Zhang Q. Short-term traffic flow prediction based on LSTM-XGBoost combination model. Computer Modeling in Engineering and Sciences. 2020 Oct 6; 125(1): 95-109.
[12] Cai L, Lei M, Zhang S, Yu Y, Zhou T, Qin J. A noise-immune LSTM network for short-term traffic flow forecasting. Chaos: An Interdisciplinary Journal of Nonlinear Science. 2020 Feb 1; 30(2).
[13] Xia D, Zhang M, Yan X, Bai Y, Zheng Y, Li Y, et al. A distributed WND-LSTM model on MapReduce for short-term traffic flow prediction. Neural Computing and Applications. 2021 Apr; 33: 2393-410.
[14] Zhang Z, Jiao X. A deep network with analogous self-attention for short-term traffic flow prediction. IET Intelligent Transport Systems. 2021 Jul; 15(7): 902-15.
[15] Fang W, Zhuo W, Yan J, Song Y, Jiang D, Zhou T. Attention meets long short-term memory: A deep learning network for traffic flow forecasting. Physica A: Statistical Mechanics and its Applications. 2022 Feb 1; 587: 126485.
[16] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. Advances in Neural Information Processing Systems. 2016; 29.
[17] Velickovic P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. Stat. 2017 Oct; 1050(20): 10-48550.
[18] Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arxiv preprint arxiv:1709.04875. 2017 Sep 14.
[19] Guo S, Lin Y, Feng N, Song C, Wan H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence. 2019 Jul 17; 33(1): 922-929.
[20] Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, et al. T-gcn: A temporal graph convolutional network for traffic prediction. IEEE Transactions on Intelligent Transportation Systems. 2019 Aug 22; 21(9): 3848-58.
[21] Bai L, Yao L, Li C, Wang X, Wang C. Adaptive graph convolutional recurrent network for traffic forecasting. Advances in Neural Information Processing Systems. 2020; 33: 17804-15.
[22] Zheng C, Fan X, Wang C, Qi J. Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence. 2020 Apr 3; 34(1): 1234-1241.
[23] Song C, Lin Y, Guo S, Wan H. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence. 2020 Apr 3; 34(1): 914-921.
[24] Wang J, Wang W, Liu X, Yu W, Li X, Sun P. Traffic prediction based on auto spatiotemporal Multi-graph Adversarial Neural Network. Physica A: Statistical Mechanics and its Applications. 2022 Mar 15; 590: 126736.
[25] Yin X, Wu G, Wei J, Shen Y, Qi H, Yin B. Multi-stage attention spatial-temporal graph networks for traffic prediction. Neurocomputing. 2021 Mar 7; 428: 42-53.
[26] Zhang X, Xu Y, Shao Y. Forecasting traffic flow with spatial–temporal convolutional graph attention networks. Neural Computing and Applications. 2022 Sep; 34(18): 15457-79.
[27] Li Y, Zhao W, Fan H. A spatio-temporal graph neural network approach for traffic flow prediction. Mathematics. 2022 May 21; 10(10): 1754.
[28] Hu N, Zhang D, Xie K, Liang W, Diao C, Li KC. Multi-range bidirectional mask graph convolution based GRU networks for traffic prediction. Journal of Systems Architecture. 2022 Dec 1; 133: 102775.
[29] Ni Q, Zhang M. STGMN: A gated multi-graph convolutional network framework for traffic flow prediction. Applied Intelligence. 2022 Oct; 52(13): 15026-39.
[30] Yin X, Li F, Shen Y, Qi H, Yin B. NodeTrans: A Graph Transfer Learning Approach for Traffic Prediction. 2022 Jul 4. arXiv.2207.01301.
[31] Jin D, Shi J, Wang R, Li Y, Huang Y, Yang YB. Trafformer: Unify time and space in traffic prediction. InProceedings of the AAAI Conference on Artificial Intelligence. 2023 Jun 26; 37(7): 8114-8122.
[32] Yu X, Bao YX, Shi Q. STHSGCN: Spatial-temporal heterogeneous and synchronous graph convolution network for traffic flow prediction. Heliyon. 2023 Sep 1; 9(9).
[33] Chen J, Zheng L, Hu Y, Wang W, Zhang H, Hu X. Traffic flow matrix-based graph neural network with attention mechanism for traffic flow prediction. Information Fusion. 2024 Apr 1; 104: 102146.

[34] Liu F. A passenger flow prediction method using SAE-GCN-BiLSTM for Urban Rail Transit. International Journal of Swarm Intelligence Research (IJSIR). 2024 Jan 1; 15(1): 1-21.
[35] Blondel VD, Guillaume JL, Lambiotte R, et al. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment. 2008; 1-6.
[36] Manyol M, Eke S, Massoma A, Biboum A, Mouangue R. Preprocessing approach for power transformer maintenance data mining based on k-Nearest neighbor completion and principal component analysis. International Transactions on Electrical Energy Systems. 2022 Oct 03; 4: 10.