International Journal of Innovative Computing, Information and Control
Volume **19**, Number **4**, August **2023**
 ISSN 1349-4198
pp. 1251–1264

MCL-STGAT: TAXI DEMAND FORECASTING USING SPATIO-TEMPORAL GRAPH ATTENTION NETWORK WITH MARKOV CLUSTER ALGORITHM

Taoyi Zhang[1,2], Yubo Wang[1,2] and Zhicheng Wei[1,2,*]

[1]College of Computer and Cyber Security
[2]Key Lab of Network and Information Security of Hebei Province
Hebei Normal University
No. 20, South Second Ring Road East, Yuhua District, Shijiazhuang 050024, P. R. China
{ zty; wangyubo }@stu.hebtu.edu.cn; *Corresponding author: weizhicheng@hebtu.edu.cn

Received October 2022; revised February 2023

Abstract. *Spatial-temporal data prediction of taxi demand is a challenging task because of complicated spatial dependencies and dynamical trends of temporal pattern between different regions. However, limited representations of given spatial graph structure with incomplete adjacent connections may restrict effective spatial-temporal dependencies learning of model. This study proposes a taxi demand prediction model that combines node2vec algorithm with a graph attention network and Markov cluster algorithm with convolution operation to capture spatial dependency and adopts Long Short-Term Memory (LSTM) to obtain temporal correlation. The core of model proposed in this paper is to construct spatial block. First, we propose a novel method for constructing traffic graph based on Markov cluster algorithm, and name semantic correlation graph. Then, considering complexity of traffic data, we employ graph attention layer with mask graph operation and use modified convolutional operation to obtain deep spatial dependency. Finally, experimental results on New York City (NYC) Taxi dataset and Chengdu online taxi dataset demonstrate that our method achieves state-of-the-art performance consistently compared with other baselines.*
Keywords: Taxi demand prediction, Graph attention network, Markov cluster algorithm, Node2vec algorithm

1. **Introduction.** People in urban areas typically use taxis or online taxi for travel, which provide convenience and comfort. According to the New York Taxi and Limousine Commission (TLC), there are currently 13,587 taxis operating in New York City, which generate billions of orders each year. There are some problems [1] with the supply and demand of taxis and passengers, for example, in one area, taxis may be empty for long periods while passengers in another area keep on waiting. Predicting future demand improves efficiency of taxi use and reduces waiting time. However, there are still numerous challenges in forecasting. Firstly, taxi demand is variable, and mainly depends on passenger demand. Secondly, taxi demand during weekdays and weekends is different. Thirdly, traffic data is non-Euclidean in structure, complex spatial correlation, and dynamic time dependence, making prediction challenging. In addition, complex spatial and temporal dependencies between regions can affect taxi demand forecasting. It is worth saying that there is a spatio-temporal pattern underlying a large amount of taxi data, and capturing this pattern helps to improve prediction results.

Among early solutions for traffic prediction issue, statistical methods and machine learning methods, for example, ARIMA and SVR, have been used [1]. These methods

DOI: 10.24507/ijicic.19.04.1251

ignore the nature of traffic patterns which had complex spatial dependence and highly dynamic temporal correlations.

In the last decade, as deep learning shines, prediction results improve dramatically. Some scholars used Convolutional Neural Networks (CNN) to capture spatial dependence and RNN to obtain temporal correlation [2]. Zhang et al. [3] proposed ST-ResNet to predict pedestrian flow. Traditional convolution operation is only applicable to Euclidean data, while traffic data is non-Euclidean structured. To handle this problem, recent studies [4, 5, 6] have adopted Graph Convolution Neural Networks (GCN) to extract spatial correlations. GCN considers structural information of traffic graph and aggregate information of neighboring nodes in non-Euclidean space for feature extraction. However, a single road network structure often cannot adequately capture spatial characteristics of data. Actually, more semantic correlations between urban roads have been exploited from various aspects [7]. Geng et al. [8] encoded relationships between regions into multiple semantic graphs to forecast demand from different perspectives. Zhu et al. [9] considered external factors and semantic relevance of traffic information to design knowledge graph for traffic speed forecasting. Wang et al. [10] proposed dynamic hypergraph convolution networks for traffic flow forecasting. All three are based on the GCN framework. However, GCN has two major limitations: one is the inability to perform inductive task, i.e., to handle dynamic graph problem, the other is the bottleneck of dealing with directed graphs, and it is not easy to implement the assignment of different learning weights to different neighbors.

More recently, Generative Adversarial Networks (GAN) [11] have been used in the field of transportation. Zhang et al. [12] proposed a TrafficGAN model for traffic flow forecasting which adopted LSTM [13] and CNN to capture the spatiotemporal dependencies within GAN framework. TFGAN proposed by [14] is a deep learning model based on multiple GCNs which are trained by generating adversarial networks for traffic forecasting. Zhang et al. [15] proposed a GCGAN model for traffic speed forecasting which combined adversarial training and graph CNN. Zhang et al. proposed another TrafficGAN in [16] which used Bi-directional LSTM and CNN with a deformable convolution kernel to predict traffic flow and speed within GAN framework.

In the latest study, Transformer [17] shines brightly. Its success is due to the attention mechanism, which gives more attention to the part of the input information that is useful for solving a task. Xu et al. [18] proposed an STTN model which adopted spatial and temporal transformers to forecast traffic speed. Chen et al. [19] proposed a Bi-STAT model for traffic flow forecasting that used spatial-adaptive transformer and temporal-adaptive transformer under an encoder-decoder architecture. Transformer applies a self-attentive mechanism to modeling global contextual information. However, this pixel-to-pixel pair-based modeling approach has computationally intensive and more requirements for hardware.

Some of the above methods use temporal extraction and spatial extraction module to predict traffic problems, which improves prediction accuracy. It is worth noting that capturing spatial dependence should consider not only the influence of other regions at the same time but also the changes on demand in other regions at different times. Focusing only on the topology of traffic network cannot obtain global dynamic spatial and temporal dependence information. To solve these problems, this work proposes a spatio-temporal graph attention network with the Markov Cluster Algorithm (MCL) [20]. In addition, this paper considers traffic semantic information and adopts a self-attention mechanism to assign different weights for different parts. The main efforts of this paper are as follows.

1) We obtain a stable and global traffic state map based on the idea of the Markov cluster algorithm, which performs random walks on a graph to find potential connected and non-connected regions in it.

2) We propose a Spatio-Temporal Graph Attention Network-based Markov Cluster Algorithm model (MCL-STGAT) that adopts Graph Attention Network (GAT) [21], CNN, and LSTM to capture spatial and temporal dependencies.

The rest of this paper is mixed as follows. Section 2 gives a detailed description of concepts related to taxi demand prediction problem and required theoretical knowledge. In Section 3, we describe our proposed approach in detail. In Section 4, we compare our method with other traffic forecasting methods and discuss the experimental results. Finally, we have a conclusion in Section 5.

2. Problem Statement and Preliminaries.

In this section, we first formulate taxi demand problem definitions, and then expound the theory of Markov cluster algorithm, graph attention layer and LSTM.

2.1. Problem formulation.

Definition 2.1. *Spatial region. We regard an administrative region as a spatial region and number these regions to obtain a set S in which r_i ($i \in [1, \ldots, N]$, $|S| = N$) is our target unit for taxi demand forecasting.*

Definition 2.2. *Taxi demand tensor. We represent citywide taxi demand distributions across regions during past T time slots as a tensor: $X \in R^{N \times T}$, where each entry x_i^t denotes generated taxi demand at region r_i in the t-th time slot.*

Definition 2.3. *Region graph. We regard a region as a node in graph, and use demand between nodes as weight of edge. Whole study area is denoted as $G(V, W, A)$, where v_i represents a region, each weight w_{ij} represents correlation strength between v_i and v_j. A is correlation matrix and its each element $a_{ij} = w_{ij}$. Larger weight means that two regions have higher correlation.*

Definition 2.4. *Taxi demand forecasting formulation. Taxi demand forecasting is a specific problem of time series analysis. Given a specific time interval, data can be divided into continuous time series. For example, setting the interval to 20 minutes, data is defined as $\ldots, X_t, \ldots$, $-\infty < t < +\infty$. Then taxi demand can be formulated as follows:*

$$(Y_{t+1}, \ldots, Y_{t+P}) = F(X_{t-M}, \ldots, X_t)\ (M \geq 1,\ P \geq 1) \tag{1}$$

where $(X_{t-M}, \ldots, X_t)$ is historical data sample. $(Y_{t+1}, \ldots, Y_{t+P})$ is future predicted data sample. F is a forecasting function which can be constructed by deep neural networks as mentioned in above part.

2.2. Markov Cluster algorithm (MCL algorithm).

Unlike feature clustering, graph clustering is difficult to observe. For a certain node, we use graph clustering to aggregate it with closely connected nodes to obtain intrinsic correlation. MCL is a graph clustering algorithm. The heart of MCL lies the idea to simulate flow within a graph, to promote flow where the current is strong, and to demote flow where the current is weak [20]. That is to say, if you start from a point and reach one of the neighboring points, then you are much more likely to be in this cluster than to leave this current cluster to go to a new cluster [22]. MCL does not require the number of clusters to be artificially specified in advance, but can be determined by parameters. Besides, it can run on weighted or unweighted graphs. The specific process of MCL algorithm is given in Algorithm 1. The power parameter e and inflation parameter r in this algorithm are set to 2. In Section 4,

Algorithm 1. MCL algorithm

Input: a voidfree graph G, power parameter e, identity matrix I and inflation parameter r

Output: Matrix M

1: According to G, create the associated matrix M
2: (Optional) Add self loops to each node, $M = M + I$
3: Normalize the matrix M, $M_{pq} = \frac{M_{pq}}{\sum_{i=1}^{k} M_{iq}}$
4: Expand by taking the e-th power of the matrix M, $M = M^e$
5: Inflate by taking inflation of the resulting matrix with parameter r, $M_{pq} = \frac{M_{pq}^r}{\sum_{i=1}^{k} M_{iq}^r}$
6: Repeat step 4 and step 5 until a steady state is reached
7: **return** M

this paper sets the power parameter e and inflation parameter r according to experimental results.

2.3. **Graph attention layer.** Graph attention layer is the base component of GAT, which is used to learn attention coefficients between node pairs and update hidden feature of each node [23]. The process of calculating attention coefficient between node pairs is shown in Figure 1, and Equation (2) to Equation (3), where x_i and x_j are vector representation of nodes, W is weights of graph attention layer, e_{ij} is similarity coefficient, N_i is the set of neighboring nodes of node i and α_{ij} is attention coefficient.

$$e_{ij} = a\left(\left[Wx_i \parallel Wx_j\right]\right) \tag{2}$$

$$\alpha_{ij} = \frac{\exp\left(LeakyReLU\left(e_{ij}\right)\right)}{\sum_{k \in N_i} \exp\left(LeakyReLU\left(e_{ij}\right)\right)} \tag{3}$$

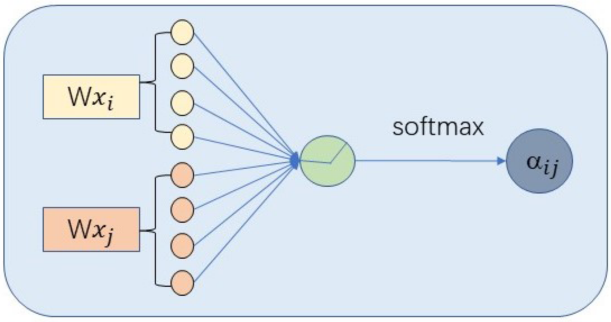


FIGURE 1. Attention coefficient

3. **Proposed Model.** In this part, we give an overview of the proposed model which contains a temporal block, a spatial block and a prediction layer. In Figure 2, the temporal block adopts an LSTM model to extract temporal correlation among input time series. The spatial block employs a graph attention layer with node2vec algorithm [24] and two convolutional layers based on MCL algorithm to capture spatial dependence among regions. The prediction layer uses a fully connected layer to obtain result.

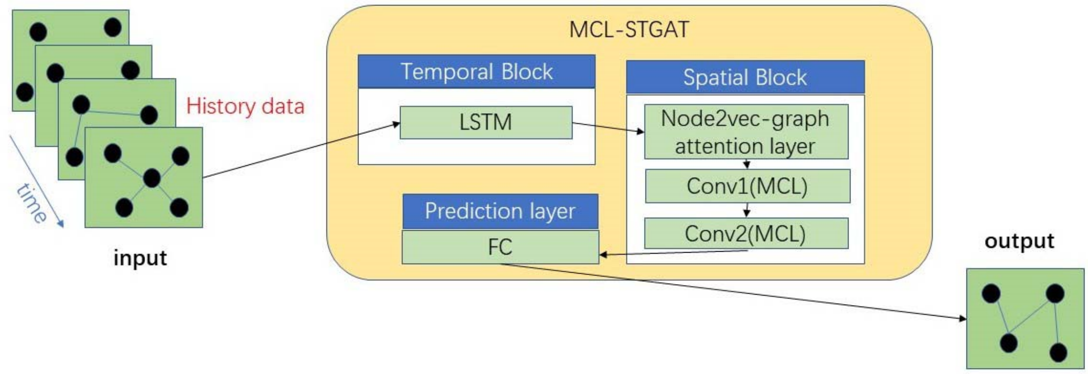


FIGURE 2. MCL-STGAT

3.1. **Temporal block.** Demand of a region is influenced not only by other regions at the same moment, but also by its own previous time. And local demand for taxis will increase as a result of previous trips by people from other areas. We reshape the demand from all regions into a time series type tensor as input to LSTM layer. LSTM has three types of gates that control cell state, which are input gate, forget gate and output gate. The structure of the LSTM unit is illustrated in Figure 3.

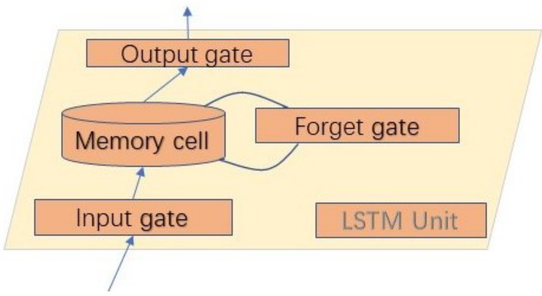


FIGURE 3. The unit of LSTM

Given an input data of $[h_{t-1}, x_t]$, the pass of an LSTM unit with gates is shown below:
Forget Gate f_t:

$$f_t = \sigma_g\big(W_f[h_{t-1}, x_t] + b_f\big) \tag{4}$$

Input Gate i_t:

$$i_t = \sigma_g\left(W_i[h_{t-1}, x_t] + b_i\right) \tag{5}$$

$$\check{C}_t = \tanh\left(W_c[h_{t-1}, x_t] + b_c\right) \tag{6}$$

Output Gate o_t:

$$o_t = \sigma_g\left(W_o[h_{t-1}, x_t] + b_o\right) \tag{7}$$

Cell State C_t:

$$C_t = f_t * C_{t-1} + i_t\check{C}_t \tag{8}$$

Output:

$$h_t = o_t * \tanh(C_t) \tag{9}$$

where h_{t-1}, x_t are output of the previous moment and input of this moment, respectively. W, b are the weights and biases in gates. C and $\check{C}$ present cell states, and $*$ presents element-wise multiplication here. $\sigma()$ and $\tanh()$ are activation functions.

3.2. Spatial block.

3.2.1. *Node2vec-graph attention layer.* The output of LSTM module is fed into spatial block to capture spatial dependency. To fully extract features, we proposed node2vec-graph attention layer and two convolution layers based on MCL algorithm, and the node2vec-graph attention layer is shown as Figure 4.

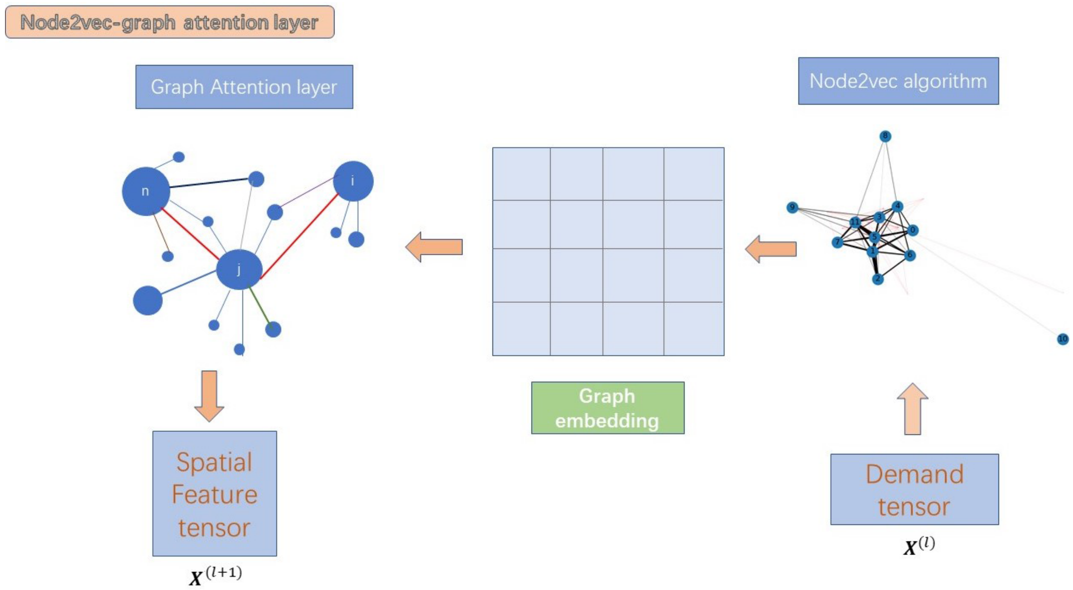


FIGURE 4. The process of node2vec-graph attention layer

Due to dynamic nature of traffic data, we cannot get an intrinsic spatio-temporal correlation between regions based on a certain state. Changes on demand in a region may be similar to geographically adjacent regions or to geographically distant regions. As Figure 5 shows, region 4 in Manhattan and region 95 in Queens are geographically distant regions, but they have a similar pattern as Figure 6 shows. Region 236 and region 237

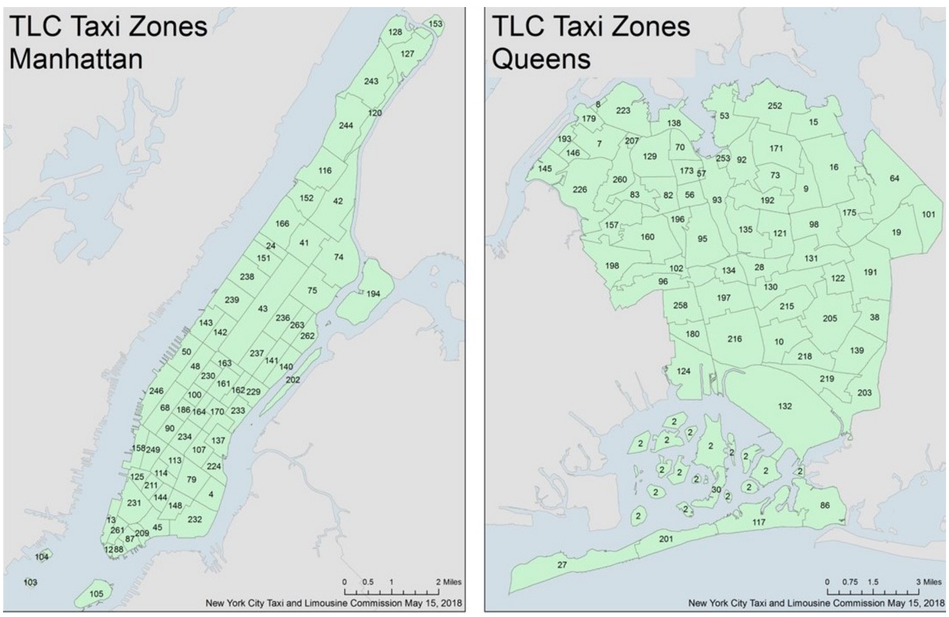


FIGURE 5. TLC taxi zones

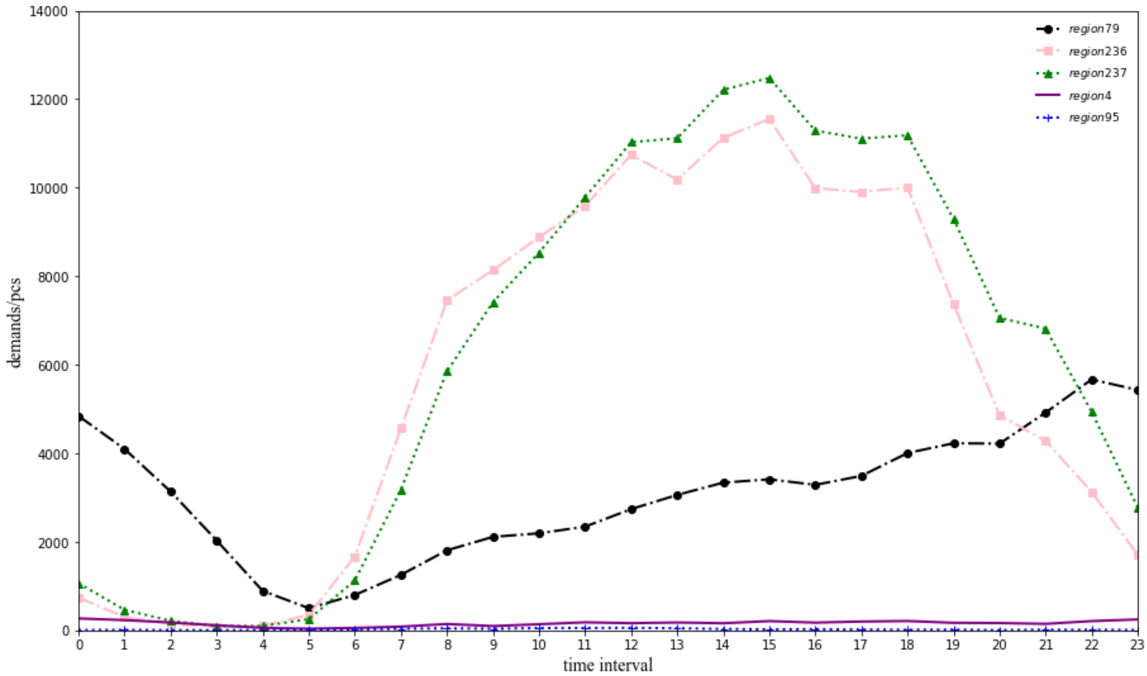


Figure 6. Changes on demand in different regions

are geographically adjacent regions in Manhattan, and they have a similar pattern. Region 79 and region 4 are also geographically adjacent regions in Manhattan, but they have different patterns. Only considering geographic location cannot capture deep spatial correlations. In this paper, we adopt attention mechanism to obtain global spatial dependency.

Graph embedding A^s: To obtain deep spatial correlations in region graph, we perform random walk with bias on it according to the idea of node2vec algorithm and this result will be applied to graph attention layer. Node2vec is designed to learn a mapping of nodes from a high-dimensional space to a low-dimensional space while maximizing the feature representation of its neighboring nodes. Node2vec-graph attention layer aims to extract the joint features of all regions at each time interval. For node2vec-graph attention layer, we denote it as

$$X_t^{l+1} = f_l\left(X_t^l, A^s\right) \tag{10}$$

where $X_t^l \in R^{N\times F_l}$ is an input of the l-th layer at time interval t, A^s is node2vec algorithm result and f_l denotes graph attention operation of the l-th as follows.

Calculate attention coefficient α_{ij}:

$$\alpha_{ij}^t = softmax\left(e_{ij}^t\right) = \frac{\exp\left(e_{ij}^t\right)}{\sum_{v_k\in N_{v_i}^t}\exp\left(e_{ij}^t\right)} \tag{11}$$

$$e_{ij} = LeakyReLU\left(\left[Wx_i^t \parallel Wx_j^t\right]\right) \tag{12}$$

Aggregate:

$$x_i = \sigma\left(\sum_{v_k\in N_{v_i}^t}\alpha_{ij}^t W A^s x_j^t\right) \tag{13}$$

where $\parallel$ denotes concatenation operation, $\sigma()$ and $LeakyReLU()$ are activation functions and W is weights of the graph attention layer.

3.2.2. *Conv (MCL) module.* After graph attention operation, we apply two convolution layers with MCL algorithm to going for further feature fusion. Taking account of semantic correlation and global spatial dependency, we perform random walk on region graph according to MCL algorithm in above and denote the result as semantic correlation graph A^N, and the operation of convolution is modified as follows:

$$X_{l+1} = ReLU\left(W_l A^N X_l + b_l\right) \tag{14}$$

where W_l and b_l are weights and biases of the convolution layer, respectively, and $ReLU()$ is an activation function.

3.2.3. *Prediction layer.* After above temporal block and spatial block, we have captured the joint spatial-temporal feature from input tensor. And then, we employ a prediction layer to map the extracted feature to a prediction demand tensor. The prediction layer is a learnable fully connected layer and formulation of it can be denoted as follows:

$$\hat{y} = ReLU\big(W_{fc}X + b_{fc}\big) \tag{15}$$

where $\hat{y}$ presents demand prediction result, X is an output from spatial block, W_{fc} and b_{fc} are parameters of fully connected layer, respectively, and $ReLU()$ presents an activation function.

4. Experiment.

To evaluate performance of the proposed model, experiment is conducted on two real-world datasets collected from NYC OpenData and Datasource: DidiChuxingGAIAInitiative.

4.1. Datasets and data processing.

NYC Taxi dataset contains order records and geographic information of taxi in New York City. This dataset contains 734 million taxicab trip records from January to July and is collected from 265 administrative regions in New York. Each trip record contains attributes including time, region number of pick-up and drop-off events, fare amount and so on. This dataset we used covers a total of 113 days. The first 89 days of data are utilized as training data, while the remaining 24 days are used as a test. Traffic data is consolidated into 20-minute intervals from raw data, and we obtain 8136 samples. To consider time periodicity of data, we reshape a sample into four parts by taking previous two-time intervals, same time interval of previous day and same time interval of previous week. The dataset is normalized using Z-Score normalization.

Chengdu online taxi dataset used in this paper contains 1.8 million order records spanning from November 1 to November 30, 2016. Each trip record includes several attributes, such as time, the number of pick-up and drop-off events within the area, and fare amount. This paper selects the study area based on the range of longitude and latitude ranges for trajectories in dataset, which is from 104.042E to 104.130E and from 30.652N to 30.728N. Due to the scattered nature of pick-up and drop-off locations, a grid division method is applied to calculating demand in each grid. The study area is divided into 1km $*$ 1km grids, resulting in 81 grids numbered from 0 to 80. The traffic data is aggregated into 10-minute intervals, resulting in 4320 samples. The first 21 days of data are used for training, and the remaining 9 days are used for testing. To normalize dataset, maximum-minimum normalization is applied.

4.2. Experimental setting.

In experiment of this paper, the values of the expansion parameter e and dilation parameter r in Markov cluster algorithm are set based on the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). RMSE and MAE are defined as follows:

$$MAE = \frac{1}{N}\sum_{i}^{N}|\hat{y}_i - y_i| \tag{16}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}^{N}(\hat{y}_i - y_i)^2} \tag{17}$$

where $\hat{y}_i$ denotes forecast result, y_i is ground truth and N represents the number of samples.

The parameters settings for MCL algorithm are presented in Table 1. It can be observed from Table 1 that the smallest RMSE and MAE values are achieved when the power parameter e and inflation parameter r are set 2. We train our model using the Adam optimizer [25] and set parameter learning rate (lr) to 0.0001 initially, then lr is set to 0.0001 after 10 epochs, and the total number of training epochs is 30. Batch size is set to 8, and the dropout is used to prevent overfitting. In addition, we employ Xavier parameter initialization to stable learning process, and we adopt a single node2vec-graph attention layer.

Table 1. The parameters settings for MCL algorithm

Parameter	Loss	
	MAE	RMSE
$e = 1$, $r = 1$	0.36	0.45
$e = 1$, $r = 2$	0.35	0.46
$\boldsymbol{e = 2}$, $\boldsymbol{r = 2}$	**0.35**	**0.45**
$e = 2$, $r = 1$	0.38	0.52
$e = 2$, $r = 3$	0.39	0.56
$e = 3$, $r = 2$	0.37	0.49
$e = 3$, $r = 3$	0.39	0.56

4.3. **Baseline methods for comparison.** MCL-STGAT is compared with following methods: ARIMA, GAT, LSTM, STGCN, and ConvLSTM. We use Mean Absolute Error (MAE), Root Mean Square Error (RMSE) to evaluate them.

4.4. **Experimental result and analysis.** Table 2 and Table 3 show comparison between different models in NYC Taxi dataset and Chengdu online taxi dataset, separately. The smallest error value is bolded. Additionally, Figure 7 shows the variation of RMSE with training epochs for different models.

Table 2. NYC Taxi dataset – Comparison between different models

Method	Loss	
	MAE	RMSE
MCL-STGAT (our)	**0.35**	**0.45**
ARIMA	1.47	1.12
GAT	0.54	1.01
STGCN	0.36	0.66
LSTM	0.44	0.76
ConvLSTM	0.41	0.75

TABLE 3. Chengdu online taxi dataset – Comparison between different models

Method	Loss	
	MAE	RMSE
MCL-STGAT (our)	**0.028**	**0.048**
ARIMA	0.104	0.083
GAT	0.079	0.126
STGCN	0.040	0.057
LSTM	0.044	0.059
ConvLSTM	0.031	0.067

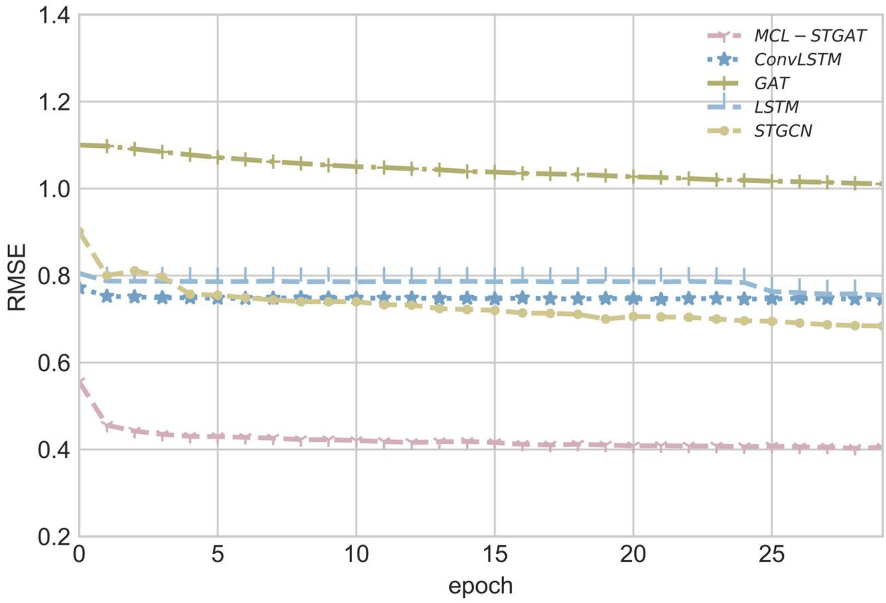


FIGURE 7. RMSE of different methods

In Table 2, comparison between ARIMA and MCL-STGAT demonstrates that RMSE of MCL-STGAT is 59.82% lower than that of ARIMA. ARIMA, as a traditional method, may have limited ability to effectively extract spatio-temporal features, which could result in reduced model performance. The deep learning-based approach is capable of effectively extracting deep spatio-temporal features, leading to improved prediction accuracy. The comparison among GAT, LSTM, and MCL-STGAT reveals that MCL-STGAT outperforms LSTM and GAT, with a reduction of 40.79% and 55.45% in RMSE, respectively. This statement suggests that considering only spatial or temporal dependence alone may not be sufficient to optimize traffic demand prediction. However, taking account of spatial-temporal dependence can lead to improved prediction accuracy. Comparing the performance of ConvLSTM and MCL-STGAT models proposed in this study, it can be concluded that RMSE of MCL-STGAT predictions is 40% lower than that of ConvLSTM. This suggests that traditional convolutional operation may be insufficient for spatial feature extraction of traffic data, and that node2vec-graph attention layer and Conv (MLC) module used in MCL-STGAT are better suited for extracting deeper spatial features, which improves the prediction accuracy of the model to some extent. The RMSE of MCL-STGAT prediction is 31.82% lower than that of STGCN. STGCN applies graph convolutional neural networks to extracting spatial features, while MCL-STGAT applies an attention mechanism to assigning different weights for different neighboring nodes. From the comparison of STGCN and MCL-STGAT, it can be concluded that the use of attention mechanism improves the prediction accuracy of the model to some extent.

The experiments conducted in the paper show that the proposed model has better prediction accuracy compared to other state-of-the-art models on two real datasets, NYC Taxi and Chengdu online taxi, which demonstrates the effectiveness and generalization ability of the proposed model.

To obtain a better understanding on prediction performance of MCL-STGAT model, we visualize ground-truth and forecasting results in New York City within next time interval, as shown in Figure 8. From this figure, we can see that our method accurately predicts demand generated in most regions. This is probably because MCL-STGAT can well capture spatial-temporal features of traffic demand.

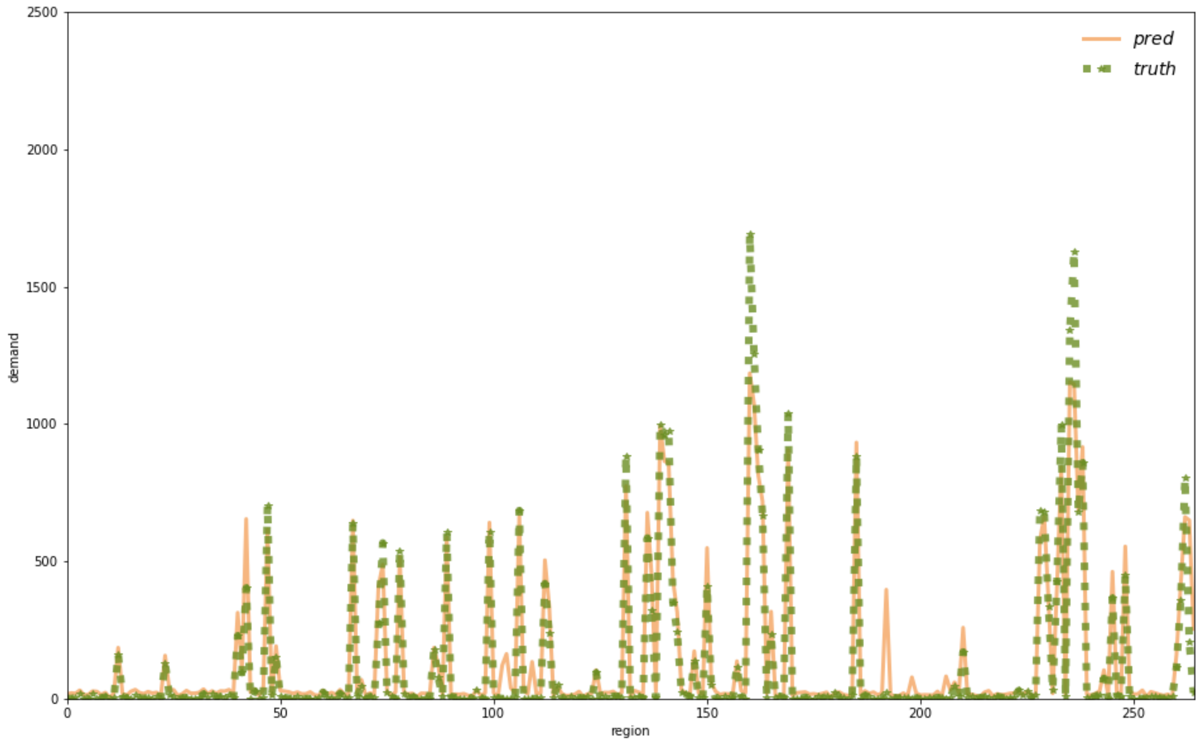


FIGURE 8. Ground-truth and predicted demands of 265 regions in New York City

4.5. **Effect of each part.** From Table 4, we can clearly observe that the MCL-STGAT model outperforms all the variants with lower MAE and RMSE values. The removal of any of the individual components, such as the LSTM layer (MCL-STGAT-NLSTM), node2vec-graph attention layer (MCL-STGAT-NNode2vec-graph-layer), or the convolutional operations with MCL algorithm (MCL-STGAT-NConv1, MCL-STGAT-NConv2), leads to a significant decrease in performance. This demonstrates that each component plays an important role in capturing the complex spatio-temporal correlations in traffic data, and the combination of these components in the MCL-STGAT model leads to improved prediction accuracy.

5. **Conclusion.** In this paper, we propose a novel spatial-temporal framework MCL-STGAT for taxi demand forecasting. Specifically, we employ LSTM, graph attention layer with node2vec algorithm and convolution with MCL algorithm to model complex

TABLE 4. Ablation experiment based on the NYC Taxi dataset

Method	Loss	
	MAE	RMSE
MCL-STGAT (our)	**0.35**	**0.45**
MCL-STGAT-NMCL	0.36	0.46
MCL-STGAT-NLSTM	0.36	**0.45**
MCL-STGAT-NNode2vec-graph-layer	0.38	0.61
MCL-STGAT-NConv1	0.37	0.49
MCL-STGAT-NConv2	0.37	0.48

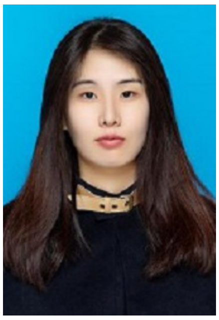
spatio-temporal correlations. Experiments on NYC Taxi dataset and Chengdu online taxi dataset show that MCL-STGAT achieves state-of-the-art results. In future, we will consider external events, such as considering weather factors to improve prediction accuracy and select stations for taxis based on demand predicting result [26]. Besides, we plan to conduct more experiments using MCL-STGAT with different spatio-temporal datasets, such as traffic flow data.

REFERENCES

[1] H. Yuan and G. Li, A survey of traffic prediction: From spatio-temporal data to intelligent transportation, *Data Science and Engineering*, vol.6, no.1, pp.63-85, 2021.
[2] J. Ye, J. Zhao, K. Ye et al., 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.
[3] J. Zhang, Y. Zheng and D. Qi, Deep spatio-temporal residual networks for citywide crowd flow prediction, *Proc. of the AAAI Conference on Artificial Intelligence*, vol.31, no.1, pp.1655-1661, 2016.
[4] J. Zhang, Y. Zheng and D. Qi, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, *Proc. of the 27th International Joint Conference on Artificial Intelligence Main Track*, pp.3634-3640, 2018.
[5] Y. Li, R. Yu, C. Shahabi et al., Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, *International Conference on Learning Representations*, 2018.
[6] L. Zhao, Y. Song, C. Zhang et al., T-GCN: A temporal graph convolutional network for traffic prediction, *IEEE Transactions on Intelligent Transportation Systems*, vol.21, no.9, pp.3848-3858, 2019.
[7] M. Lv, Z. Hong, L. Chen et al., Temporal multi-graph convolutional network for traffic flow prediction, *IEEE Transactions on Intelligent Transportation Systems*, vol.22, no.6, pp.3337-3348, 2021.
[8] X. Geng, Y. Li, L. Wang et al., Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting, *Proc. of the AAAI Conference on Artificial Intelligence*, vol.33, no.1, pp.3656-3663, 2019.
[9] J. Zhu, X. Han, H. Deng et al., KST-GCN: A knowledge-driven spatial-temporal graph convolutional network for traffic forecasting, *IEEE Transactions on Intelligent Transportation Systems*, vol.23, no.9, pp.15055-15065, 2022.
[10] J. Wang, Y. Zhang, Y. Wei et al., Metro passenger flow prediction via dynamic hypergraph convolution networks, *IEEE Transactions on Intelligent Transportation Systems*, vol.22, no.12, pp.7891-7903, 2021.
[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., Generative adversarial networks, *Communications of the ACM*, vol.63, no.11, pp.139-144, 2020.
[12] Y. Zhang, Y. Li, X. Zhou et al., TrafficGAN: Off-deployment traffic estimation with traffic generative adversarial networks, *2019 IEEE International Conference on Data Mining (ICDM)*, pp.1474-1479, 2019.
[13] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Computation*, vol.9, no.8, pp.1735-1780, 1997.
[14] A. Khaled, A. M. T. Elsir and Y. Shen, TFGAN: Traffic forecasting using generative adversarial network with multi-graph convolutional network, *Knowledge-Based Systems*, vol.249, 108990, 2022.

[15] Y. Zhang, S. Wang, B. Chen et al., GCGAN: Generative adversarial nets with graph CNN for network-scale traffic prediction, *2019 International Joint Conference on Neural Networks (IJCNN)*, pp.1-8, 2019.
[16] Y. Zhang, S. Wang, B. Chen et al., TrafficGAN: Network-scale deep traffic prediction with generative adversarial nets, *IEEE Transactions on Intelligent Transportation Systems*, vol.22, no.1, pp.219-230, 2021.
[17] A. Vaswani, N. Shazeer, N. Parmar et al., Attention is all you need, *arXiv Prints*, arXiv: 1706.03762, 2017.
[18] M. Xu, W. Dai, C. Liu et al., Spatial-temporal transformer networks for traffic flow forecasting, *arXiv Preprint*, arXiv: 2001.02908, 2020.
[19] C. Chen, Y. Liu, L. Chen et al., Bidirectional spatial-temporal adaptive transformer for urban traffic flow forecasting, *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
[20] S. M. Van Dongen, Graph clustering by flow simulation, *Standardization and Knowledge Transfer*, 2000.
[21] P. Veličković, G. Cucurull, A. Casanova et al., Graph attention networks, *International Conference on Learning Representations*, 2018.
[22] T. T. Zin, P. Tin and H. Hama, A special type of Markov branching process model for the novel coronavirus (COVID-19) outbreak, *International Journal of Innovative Computing, Information and Control*, vol.18, no.4, pp.1339-1346, 2022.
[23] W. Pian, Y. Wu, X. Qu et al., Spatial-temporal dynamic graph attention networks for ride-hailing demand prediction, *arXiv Preprint*, arXiv: 2006.05905, 2020.
[24] A. Grover and J. Leskovec, Node2vec: Scalable feature learning for networks, *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol.2016, pp.855-864, 2016.
[25] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv Preprint*, arXiv: 1412.6980, 2014.
[26] D. Tian, J. Lu and Z. Wei, A siting urban taxi stations model based on spatial-temporal origin-destination data, *International Journal of Innovative Computing, Information and Control*, vol.18, no.2, pp.477-495, 2022.

Author Biography



Taoyi Zhang received the B.Eng. degree from Software College of Hebei Normal University, China in 2020. She is currently a postgraduate student at the College of Computer and Cyber Security, Hebei Normal University, China. Her main research interests include the graph neural networks and computational social science.

Yubo Wang received the B.Eng. degree from Software College of Hebei Normal University, China in 2019. She is currently a postgraduate student at the College of Computer and Cyber Security, Hebei Normal University, China. Her main research interests include big data analysis and reinforcement learning.

Zhicheng Wei obtained his Ph.D. degree in Electronic and Information Engineering College from Tianjin University, China in 2007; he workd at the Department of Electrical and Computer Engineering, Ryerson Univeristy of Canada as the postdoctoral from May 2009 to May 2010. Professor Wei is currently a full-time professor at the College of Computer and Cyber Security, Hebei Normal University, China. He has published over 30 papers in journals and conferences. His main research interests include the computational social science, complex networks and graph networks.