Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Graph Neural Networks Empowered Origin-Destination Learning for Urban Traffic Prediction

Chuanting Zhang[1] | **Guoqing Ma**[2] | **Liang Zhang**[2] | **Basem Shihada**[2]

[1]School of Software, Shandong University, Jinan, China

[2]Computer, Electrical and Mathematical Science and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia

Correspondence
Basem Shihada
Email: basem.shihada@kaust.edu.sa

Funding Information
This research was supported by the Shandong Province Excellent Youth Science Fund Project (Overseas) under Grant No. 2024HWYQ-028, and by The Fundamental Research Funds of Shandong University under Grant No. 11480061330000.

Abstract

Urban traffic prediction with high precision is always the unremitting pursuit of intelligent transportation systems and is instrumental in bringing smart cities into reality. The fundamental challenges for traffic prediction lie in the accurate modeling of spatial and temporal traffic dynamics. Existing approaches mainly focus on modeling the traffic data itself, but do not explore the traffic correlations implicit in Origin-Destination (OD) data. In this paper, we propose STOD-Net, a dynamic Spatial Temporal OD feature enhanced deep Network, to simultaneously predict the in-traffic and out-traffic for each and every region of a city. We model the OD data as dynamic graphs and adopt graph neural networks in STOD-Net to learn a low-dimensional representation for each region. As per the region feature, we design a gating mechanism and operate it on the traffic feature learning to explicitly capture spatial correlations. To further capture the complicated spatial and temporal dependencies among different regions, we propose a novel joint feature learning block in STOD-Net and transfer the hybrid OD features to each block to make the learning process spatiotemporal-aware. We evaluate the effectiveness of STOD-Net on two benchmark datasets and experimental results demonstrate that it outperforms state-of-the-art by approximately 5% in terms of prediction accuracy and considerably improves prediction stability up to 80% in terms of standard deviation.

KEYWORDS
Traffic prediction, spatial-temporal modeling, deep neural networks, origin-destination learning

1 | INTRODUCTION

Smart transportation systems shape the blood vessels of a city and promote the fast development of social society. Along this line, highly accurate urban traffic prediction plays an essential part in intelligent transportation systems and facilitates the realization of smart cities [1,2]. As per the knowledge of traffic prediction, intelligent traffic control can be achieved to enhance travel efficiency, reduce traffic congestion, and improve citizens' quality of life. Traffic prediction can also strengthen public safety by predicting the traffic volume for each and every region of a city [3]. In addition, a significant number of vehicular applications, such as route planning, navigation, and travel time estimation, rely heavily on traffic condition evaluation [4,5,6,7,8]. Consequently, research on traffic prediction problems [9,10,11,12] has been active for decades and has received massive attention from both academia and industry.

Traffic prediction refers to the problem of predicting future traffic statuses such as volume, speed, and congestion, by modeling historical traffic data. The prediction horizon can be either short-term, for example, fifteen or thirty minutes, or long-term, for example, several hours later. Depending on the application scenarios and data structures, the traffic prediction can be applied for different road sensors/segments [13], or any region of a city in arbitrary granularities [14]. In this study, we focus on short-term traffic volume prediction for all regions of a city.

Although it has huge importance in intelligent transportation systems, traffic prediction with high precision still retains an extremely challenging task, as traffic status evolves in a highly complicated and nonlinear way, especially when the status transits between free flow, unstable, congestion, and recovery [15]. Thus the traffic volume series is of great nonlinear, and there exist both spatial and temporal dependencies among traffic series of different locations.

Lots of work has been proposed to solve the traffic prediction problem. Some initial works focus on using statistical models, such as Kalman filtering [16] and Auto-Regressive Integrated Moving Average (ARIMA) [17]. These models have clear mathematical definitions, leading to high interpretability. However, it is essential to recognize that they have few parameters and come with low predictability limitations. In particular, they become inefficient when traffic series are nonlinear and have spatial dependencies.

The advancement of machine learning techniques [18] and the increasing availability of big traffic data have made data-driven models strong competitors to statistical models for traffic prediction. Support vector machines, random forests, and neural networks have been explored for traffic prediction [19]. In particular, deep neural networks have drawn the most significant attention as their strong representation ability in an end-to-end way [20,21]. Long-short term memory networks (LSTM) and convolutional neural networks (CNN) are widely investigated in current literature. LSTM can model temporal dependence as it has feedback connections between different neurons. Together with the inputs of other traffic series, the spatial dependence can also be captured by LSTM. Similarly, CNN is able to capture both spatial and temporal dependencies when the input has multiple sequences, as convolution operation is a natural way to fuse information [22]. In particularly, a deep CNN framework was proposed to solve the spatial-temporal modeling of traffic prediction [23]. Deep residual learning was also introduced into traffic prediction for collectively forecasting the in-traffic and out-traffic of a region simultaneously [24]. Apart from CNN-oriented frameworks, the growing prominence of graph neural networks (GNN) [25,26,27,28,29] has spurred interest in designing graph-based learning methods with the ability to naturally support spatial modeling of traffic prediction. Along this line, spatial-temporal graph convolutional networks were designed to extend traffic prediction from grid structures to general domains [30,31,32].

Most of the above work models spatial-temporal dependencies of different city regions using historical traffic only. However, relying solely on historical traffic data may not capture the heterogeneity of spatial dependencies well. The reason is that almost all traffic sequences are time series with autocorrelation and periodicity characteristics, resulting in non-negligible spatial correlations for any two different traffic series. The prediction model faces considerable challenges when learning from this kind of data and is unable to distinguish which traffic series is relevant and which is not but maybe noise, especially for urban traffic prediction in city-wise [33]. Besides, both spatial and temporal traffic dependencies are not static. Instead, they are highly time-varying, depending on the current traffic situation and road network. Take for example the places that are penetrated by highways, there exist more areas with higher spatial correlation because the drivers drive fast and the distance they move is long. On the contrary, for the city center, the speed of cars is relatively low and the areas with higher spatial correlations are less.

To cope with the above challenges and enhance spatial-temporal dependencies modeling, we resort to introducing origin-destination (OD) data into traffic prediction for two reasons. First, the OD data records the traffic interactions among regions of a city and provides detailed statistics that generate the traffic data of each region. Thus, the OD data can reflect the spatial correlation more directly. Second, OD data, together with static road network data, can be regarded as auxiliary information for regional dependence modeling from the perspective of data fusion. They provide different dimensions for understanding spatial-temporal dependencies of regions. Thus, based on the historical traffic data, the OD data and the static road network information, we propose a novel deep learning framework called STOD-Net, that is, Spatial-Temporal OD feature enhanced deep Network, for urban traffic prediction. To the best of our knowledge, we are the first to predict grid-based traffic volume by incorporating OD data and real-world road network data. We leverage CNN to model the historical traffic data as they are sequences of matrices. Moreover, since the OD data are inherently in a graph structure, leading to the ineffectiveness of traditional CNN, we use GNNs to model them to obtain hidden representations. GNNs are novel neural networks designed especially for non-Euclidean data and gain increasing popularity in the deep learning field.

STOD-Net consists of four components. The first component is traffic feature learning, from which we obtain the hidden representations of different regions. In the second component, we perform OD feature learning, followed by a gating mechanism that functions as an enhanced spatial dependence modeling. Then in the third component, we perform joint feature learning for the historical traffic data and the OD data using convolutional layers with introduced dense connectivity. To enhance the modeling of spatial-temporal dependencies, we further transfer the learned OD features to the subsequent learning blocks and this yields the fourth component of STOD-Net. In summary, our main contributions are listed as follows.

- We propose STOD-Net, a spatial-temporal deep learning framework, for urban traffic prediction. STOD-Net accepts not only historical traffic data but also the OD data and static road network, and can model these two kinds of data simultaneously.
- We introduce GNN to model the OD data, as they retain the traffic interactions among regions. On the basis of the OD representations, we further propose a gating mechanism that operates on the traffic representations and acts as a guide to enhance spatial-temporal dependencies modeling.

We present a joint modeling scheme to both dynamic and static OD data, making STOD-Net more robust in capturing spatial dependence.

- We validate the effectiveness of our STOD-Net on two real-world datasets, and the results demonstrate that STOD-Net improves the prediction performance greatly.

The rest of this paper is organized as follows. Section II devotes to the related work on traffic prediction. Section III gives the problem formulation and preliminaries on graph neural networks. The detail of our proposed STOD-Net is introduced in Section IV. We report experimental results in Section V and conclude our paper in Section VI.

2 | RELATED WORKS

Current studies on traffic prediction can be classified into statistical model-based methods, traditional machine learning-based methods, deep learning-based methods, and OD-related traffic prediction. This section devotes to a brief review of related works on these three categories.

2.1 | Statistical model based traffic prediction

Methods that fall into this category mainly center on ARIMA and its variants. One of the earliest works that used ARIMA model for traffic prediction was completed in [34] by Ahmed and Cook, in which they discovered that ARIMA model can be more accurate than moving average and exponential smoothing in representing traffic volume data. Afterward, researchers explored other versions of ARIMA for traffic prediction including ARIMAX [35] and subset ARIMA [36]. Notwithstanding the popularity of ARIMA-based models in traffic prediction, they face great limitations on predictive ability. The reason is that they are simply linear models, which assume the traffic is stationary. Therefore, they usually fail when dealing with highly complicated nonlinear traffic data [37]. Despite a few nonlinear models being proposed like heteroskedasticity-based models, the analytical study of nonlinear models is still in its infancy compared to linear models.

2.2 | Traditional machine learning based traffic prediction

Due to the inefficacy of statistical models in solving the traffic prediction problem, researchers resort to machine learning models, which are flexible, data-driven, and can adjust the parameters automatically. Wu *et al.* applied support vector regression to traffic prediction [38] and achieved better prediction performances than statistical models since its strong generalization ability and guaranteed global minimum. *k*-nearest neighbor models were also explored for traffic prediction [39,40] and found that they can achieve better prediction performance than support vector regression in terms of mean squared error. Besides, random forests and artificial neural networks were also adopted for traffic prediction [41]. Though machine learning models have stronger predictive abilities than statistical models, they were more challenging to train ten years ago, particularly for neural networks. Besides, traditional machine learning models have few parameters, leading to limited performance gains for traffic prediction.

2.3 | Deep learning based traffic prediction

Deep learning models have been the mainstream for traffic prediction since the rise of deep neural networks. Different types of neural networks can be adopted for traffic prediction based on the data on hand. According to a recent survey [2], point-based data mainly adopt LSTM and GNN, and trajectory-based data use CNN more.

For point-based data, LSTM [42], bidirectional LSTM [43], and LSTM with feature enhancement [44] are investigated in current literature, respectively. These models are good at capturing the temporal dependency of traffic volume. To further model the spatial dependency, GNN is introduced into traffic prediction [45,46], as it can model the spatial relationships of different locations. Afterward, many works on GNN-based traffic prediction methods were proposed, such as spatial-temporal GNN [47], GNN with attention scheme [13] and multi-attention scheme [20].

For trajectory-based data, most prediction methods are built on top of CNN such as DeepST [23] and ST-ResNet [24]. The key difference between these two models is that ST-ResNet introduces residual learning strategy into traffic prediction, leading to a great performance improvement. CNN is also the basic learning block of several other frameworks based on meta-learning [14], context-aware learning [48], and multi-view learning [9,31]. These frameworks can capture both spatial and temporal dependencies of different regions of city-wise but from different perspectives.

2.4 | OD-realted traffic prediction

Recently, some works on adopting OD data for traffic prediction of a location or predicting OD matrix were proposed. For example, OD traffic was predicted by a hybrid spatial-temporal network[49] and a dynamic graph convolutional recurrent network was proposed[50] to model the dynamic characteristics of correlations among locations. Besides, the ideas of multi-task learning and generative adversarial networks were also explored in current literature. In[51], a multi-task adversarial spatial-temporal network model was proposed to simultaneously predict the traffic flow and OD flow. An encoder-decoder structure was designed to capture the spatial-temporal dependencies of different regions, and a discriminative loss on task classification and an adversarial loss on shared feature extraction are incorporated to reduce information redundancy. The discriminative loss was also adopted in[52] for traffic flow prediction, yet under the framework of Generative Adversarial Nets (GAN) and the proposed model is TrafficGAN. One advantage of TrafficGAN is that a deformable convolution kernel for CNN is adopted to handle the input road network data better. More recently, Wang *et al.* proposed MC-STGCN[32], a Multivariate Correlation-aware Spatio-temporal Graph Convolutional Networks for multi-scale traffic prediction. In MC-STGCN, auxiliary information such as traffic speed and traffic occupancy rate was introduced into traffic prediction for correlation measurement and accuracy improvements.

The fundamental difference between our work and the above works lies in that we use the OD data and static road network information to enhance spatial dependency modeling, instead of the static distance between locations. We further propose a gating mechanism that can effectively fuse the historical traffic data and the OD data.

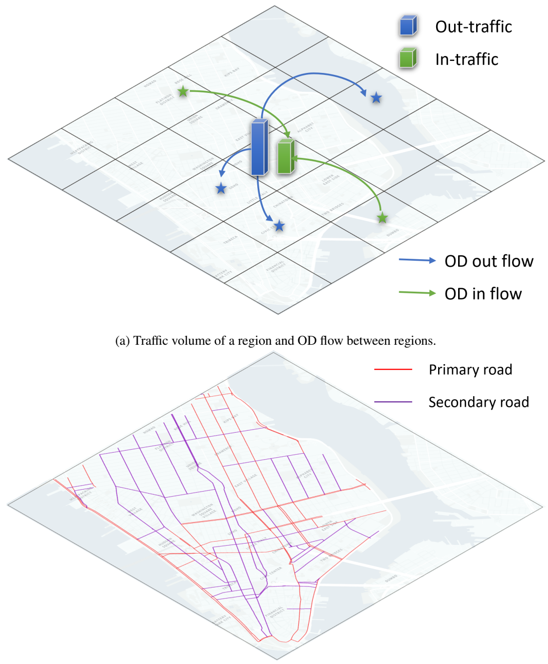
3 | PROBLEM FORMULATION AND PRELIMINARIES

This section devotes to the problem formulation of urban traffic prediction and the introduction of graph convolutional networks.

3.1 | Problem formulation

Definition 1 (Region)**.** Many feasible ways exist to define a region of a city and in this paper, we consider the following scenario: The city is evenly partitioned into an $I \times J$ grid map based on the longitude and latitude and a grid (i,j) denotes a region.

Definition 2 (In-Traffic/Out-Traffic)**.** For each region n, we consider two types of traffic volume, i.e., in-traffic and out-traffic. The in-traffic, $x^{(in,i,j)}$, indicates how many vehicles enter region n from neighboring regions. Similarly, the out-traffic, $x^{(out,i,j)}$, indicates how many vehicles drive out of this region.



(a) Traffic volume of a region and OD flow between regions.

(b) Static road network.

FIGURE 1 Example of the in- and out-traffic, OD in- and out-flow, and the static road network of New York City.

We denote the traffic volume as $\mathbf{X}_t \in \mathbb{R}^{2\times I\times J}$ for the grid map at time slot t , where $t \in [1, 2, 3, \cdots, T]$.

Definition 3 (OD Data)**.** At t-th time slot, some vehicles are driving from region (i,j) to region (i',j') and others may drive from (i',j') to (i,j). These records are called origin-destination flow data and we denote the traffic from (i,j) to (i',j') as $d^{\{(i,j),(i',j')\}}$. Thus, we use $\mathbf{D}_t \in \mathbb{R}^{N\times N}$ to denote the OD flow data at the t-th time slot, where $N = I \times J$ represents the total number of regions of the city.

Definition 4 (Static Road Network)**.** All the regions are connected by physical roads, based on which we obtain an adjacency matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$, denoting the static road network.

To make the understanding of these definitions easy, a toy example of the defined region, in- and out-traffic, OD flow, and the corresponding static road network is displayed in Fig. 1.

Definition 5 (Definition)**.** Given a series of historical traffic data $\{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_t\}$, the corresponding OD flow data $\{\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_t\}$, and static road network $\mathbf{S}$, predict $\mathbf{X}_{t+1}$.

Mathematically, our problem can be formally defined as

$$w^* = \arg\min_{w} \mathcal{L}(f(\mathcal{X}_t, \mathcal{D}_t, \mathbf{S}|w), \mathbf{X}_{t+1}), \tag{1}$$

† Time slot t denotes a time interval $(t, t+\Delta t]$, where Δt is the temporal granularity of the data. Δt could be 15 minutes or one hour based on the data on-hand.

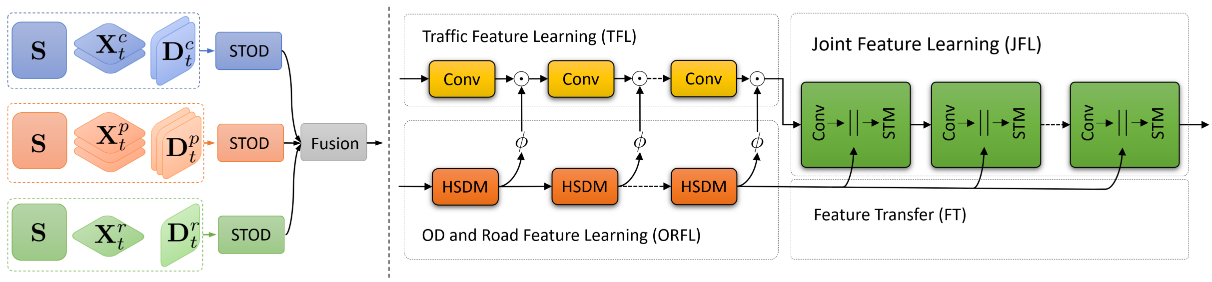


FIGURE 2 STOD-Net framework and the corresponding key component STOD: Convolutional (Conv), hybrid spatial dependence modeling (HSDM), and spatial temporal modeling (STM). $\odot$ and $\|$ denote the Hadamard product and concatenation operation. ϕ is a mapping function that implements the gating mechanism.

where w denotes all the parameters of our model $f(\cdot)$. $\mathcal{X}_t$ and $\mathcal{D}_t$ are the input features extracted from traffic volume data and OD data, respectively, and their construction will be detailed in Section IV. Moreover, $\mathcal{L}(\cdot,\cdot)$ is a loss function used to measure the goodness of our prediction model.

3.2 | Graph neural networks

Graph neural networks (GNNs) refer to frameworks for representation learning on graphs and their goal is to automatically learn a low-dimensional representation for the nodes, edges, or the whole graph, in an end-to-end fashion. Most GNN frameworks have a unified architecture, which can be described as

$$\mathbf{H}^{(l+1)} = g(\mathbf{H}^{(l)}, \mathcal{A}), \tag{2}$$

where l indexes the layers, and $\mathbf{H}^{(l)}$ denotes the hidden representation in layer l. Note that $\mathbf{H}^{(0)} = \mathbf{X}$ represents the input feature matrix. $\mathcal{A}$ is the adjacency matrix corresponding to the graph. g is a nonlinear function and different GNN frameworks have different choices of g.

One of the most popular GNN frameworks is Graph Convolutional Networks (GCNs) and the update rule is denoted as

$$g(\mathbf{H}^{(l)}, \mathcal{A}) = \sigma(\bar{\mathcal{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{3}$$

where $\bar{\mathcal{A}}$ is the symmetrically normalized graph Laplacian of $\mathcal{A}$ with self-loops, $\mathbf{W}^{(l)}$ is a parameter matrix of layer l, and σ is the rectified linear unit (ReLU) function. Eq. (3) indicates that when updating the representation for a node in the graph, GCN performs a weighted average of all neighbor's representations of that node.

The attention scheme can also be considered in g and this yields to graph attention networks[25], or GAT for short. The update rule in GAT can be described as

$$g(\mathbf{H}^{(l)}, \mathcal{A}) = \sigma(\mathbf{A}^{(l)}\mathbf{H}^{(l)}\mathbf{W}^{(l)}). \tag{4}$$

Eq. (4) is similar with Eq. (3) except that $\mathbf{A}^{(l)}$ is the attention weight matrix for layer l instead of graph Laplacian of $\mathcal{A}$.

In this study, we use GATs to learn representations for each region, based on which we propose a gating mechanism to enhance characterizing the hidden spatial dependencies.

4 | STOD-NET ARCHITECTURE

In this section, we first give a general introduction to our proposed traffic prediction framework. Then we detail each of its components to demonstrate how we model the spatial and temporal dependencies of urban traffic.

4.1 | Overall architecture

To solve Eq. (1), we propose a novel traffic prediction framework, i.e., STOD-Net, which can effectively fuse the information from both the traffic data and the OD data, thereby capturing the spatial and temporal correlations among different regions simultaneously. Fig. 2 illustrates the detail of STOD-Net, in which its fundamental component, i.e., spatial-temporal origin-destination learning (STOD), is also included.

As current literature has shown that traffic volume's temporal correlations can be effectively characterized by closeness dependence, periodicity dependence, and trend dependence[24]. Thus in this paper, we embrace this point of view and resort to three separate subnetworks to learn the hidden representations for these temporal dependencies to capture their different impacts on future traffic volume prediction. To reduce complexity, these three subnetworks are designed with shared architecture. As shown in Fig. 2, there are three types of inputs, i.e., the traffic volume data $\mathcal{X}_t = \{\mathbf{X}_t^c, \mathbf{X}_t^p, \mathbf{X}_t^r\}$, the OD data $\mathcal{D}_t = \{\mathbf{D}_t^c, \mathbf{D}_t^p, \mathbf{D}_t^r\}$ and the static road network data denoted by adjacency matrix $\mathbf{S}$. Specifically, $\{\mathbf{X}_t^c, \mathbf{D}_t^c, \mathbf{S}\}$, $\{\mathbf{X}_t^p, \mathbf{D}_t^p, \mathbf{S}\}$, and $\{\mathbf{X}_t^r, \mathbf{D}_t^r, \mathbf{S}\}$ are used to model the closeness dependence, periodicity dependence, and trend dependence, respectively. The constructions of $\mathbf{X}_t^c$ and $\mathbf{D}_t^c$ are as follows.

$$\begin{aligned}\mathbf{X}_t^c &= \mathbf{X}_{t-1} \,\|\, \mathbf{X}_{t-2} \,\|\, \cdots \,\|\, \mathbf{X}_{t-L_c},\\ \mathbf{D}_t^c &= \mathbf{D}_{t-1} \,\|\, \mathbf{D}_{t-2} \,\|\, \cdots \,\|\, \mathbf{D}_{t-L_c},\end{aligned} \tag{5}$$

where L_c is the length of closeness sequence and $\|$ denotes the concatenation operator. Similarly, $\mathbf{X}_t^p$, $\mathbf{D}_t^p$, $\mathbf{X}_t^r$, and $\mathbf{D}_t^r$ can be obtained as follows.

$$\begin{aligned}\mathbf{X}_t^p &= \mathbf{X}_{t-P} \,\|\, \mathbf{X}_{t-2P} \,\|\, \cdots \,\|\, \mathbf{X}_{t-L_pP},\\ \mathbf{D}_t^p &= \mathbf{D}_{t-P} \,\|\, \mathbf{D}_{t-2P} \,\|\, \cdots \,\|\, \mathbf{D}_{t-L_pP},\\ \mathbf{X}_t^r &= \mathbf{X}_{t-R} \,\|\, \mathbf{X}_{t-2R} \,\|\, \cdots \,\|\, \mathbf{X}_{t-L_rR},\\ \mathbf{D}_t^r &= \mathbf{D}_{t-R} \,\|\, \mathbf{D}_{t-2R} \,\|\, \cdots \,\|\, \mathbf{D}_{t-L_rR},\end{aligned} \tag{6}$$

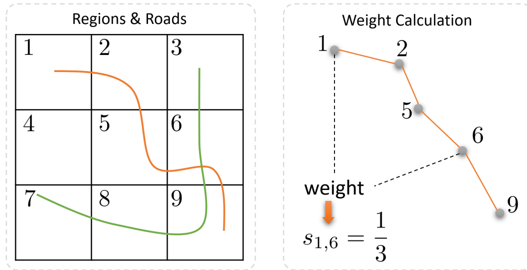
where L_p and L_r are the lengths of the periodicity sequence and trend sequence, P and R are the period and trend span corresponding to one day and one week, respectively. The adjacency matrix $\mathbf{S}$ is constructed from physical roads and the details will be given in the following subsection.

For STOD, as shown in the right of Fig. 2, it consists of four components, i.e., traffic feature learning (TFL), OD and road feature learning (ORFL), feature transfer (FT), and joint feature learning (JFL). The TFL's role is to learn initial representations for the traffic data. Like the TFL, the ORFL learns representations for the OD data and the static road network and then fuses them with the traffic representations. FT means we transfer the learned OD and road feature to subsequent layers for spatial and temporal modeling. As for the JFL, its function is to perform joint feature learning for the fused representations, including the traffic information and the OD information. Once we obtain the final representations for the closeness, periodicity, and trend data, we fuse them together, as shown in the left of Fig. 2, followed by a non-linear activation, and we get the predicted traffic volume $\hat{\mathbf{X}}_{t+1}$ for time slot $t+1$.

In the next, we first introduce how we construct the adjacency matrix for the static road network, then we give technical details for each of the components of our proposed STOD. For brevity, we use $\{\mathbf{X}_t, \mathbf{D}_t, \mathbf{S}\}$ to denote any of the data sequences of $\{\mathbf{X}_t^c, \mathbf{D}_t^c, \mathbf{S}\}$, $\{\mathbf{X}_t^p, \mathbf{D}_t^p, \mathbf{S}\}$, and $\{\mathbf{X}_t^r, \mathbf{D}_t^r, \mathbf{S}\}$ and omit the subscription. This is because the three sub-networks share the same architecture, which indicates the inputs receive exactly the same operations, only the outputs are different.

4.2 | Static road network construction

We adopt an adjacency matrix to represent the static road network, in which nodes denote regions and edges denote connection weights between regions. The weights are constructed from physical roads that are crawled from OpenStreetMap. The detail of weight construction is as follows.



Induced Adjacency Matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0.50 & 0.33 & 0 & 0 & 0.25\\ 1 & 0 & 0 & 0 & 1 & 0.50 & 0 & 0 & 0.33\\ 0 & 0 & 0 & 0 & 0 & 1 & 0.25 & 0.33 & 0.50\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0.50 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0.50\\ 0.33 & 0.50 & 1 & 0 & 1 & 0 & 0.33 & 0.50 & 2\\ 0 & 0 & 0.25 & 0 & 0 & 0.33 & 0 & 1 & 0.50\\ 0 & 0 & 0.33 & 0 & 0 & 0.50 & 1 & 0 & 1\\ 0.25 & 0.33 & 0.50 & 0 & 0.50 & 2 & 0.50 & 1 & 0 \end{bmatrix}$$

FIGURE 3 A toy example for constructing the adjacency matrix, i.e., $\mathbf{S}$, representing the static road network.

- We extract the information on all kinds of road types and classified them into four categories, i.e., primary, secondary, tertiary, and others;
- For each road of these four categories, we extract the region set using a depth-first search strategy. Then for any given region pair u and v, the connection weight is defined as

$$s_{u,v}^z = \frac{1}{d_{u,v}^z}, \tag{7}$$

  where $z \in \{\text{primary}, \text{secondary}, \text{tertiary}, \text{others}\}$ denotes the road type and $d_{u,v}^z$ the shortest path between u and v for the case of road type z.
- We assign a weight ρ_z for road type z since different roads have different capacities. For example, primary roads normally have more lanes than secondary roads, thus it has a greater impact on traffic. For region pair u and v, we summarize the weights of different road types and yield the final weight

$$s_{u,v} = \sum_i \rho_z s_{u,v}^z. \tag{8}$$

Fig. 3 shows a toy example for the construction of the adjacency matrix of the static road network. In this example, two roads belong to different types, connecting nine regions. The weights of different road types are set to 1 for simplicity.

4.3 | Traffic feature learning

For city-wise urban traffic prediction, convolutional networks have shown superior performance in capturing the spatial correlation of traffic volume of different regions. Thus, we adopt

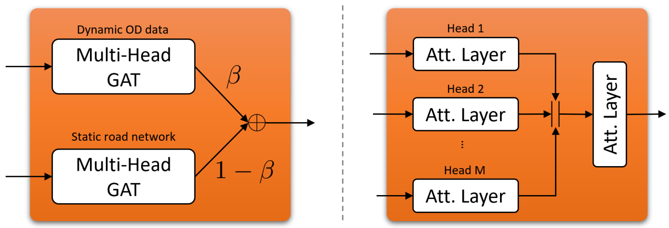


FIGURE 4 Design details of HSDM (left) and multi-head GAT (right).

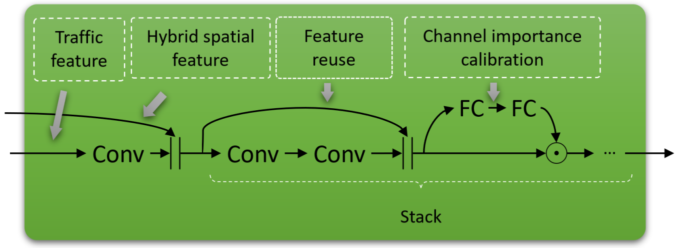


FIGURE 5 Design of joint feature learning module.

convolutional networks to learn the hidden representations of traffic volume data. In TFL, the obtained representations at layer l are defined as

$$\mathbf{X}_t^{(l)} = \sigma(\mathbf{W}_{\mathrm{TFL}}^{(l)} * \mathbf{X}_t^{(l-1)}), \tag{9}$$

where $*$ is the convolution operation, and σ is the ReLU function. In TFL, there are in total L_{TFL} layers for traffic feature learning and the final representation is denoted as $\mathbf{X}_t^{(L_{\mathrm{TFL}})}$.

4.4 | OD and road feature learning

The OD data records the traffic volume interactions from one region to another and explicitly reflects the connectivity of different regions. In addition, the traffic dynamics are largely influenced and constrained by the static road network. Thus, we use the dynamic OD data and the static road network to enhance the spatial dependence modeling of traffic volume. The challenge herein is how to design an approach that can effectively model these two types of data and fuse them with the traffic volume representation.

To solve this challenge, we propose a hybrid spatial dependence modeling (HSDM) component by simultaneously learning the dynamic OD data and the static road network. For the dynamic graph, we use $\mathbf{D}_t$ to denote its adjacency matrix and $\mathbf{H}_t$ its input feature. In this study, we use the in-traffic and the out-traffic of each node as the features. Therefore, $\mathbf{H}_t \in \mathbb{R}^{N\times 2}$ is obtained from $\mathbf{X}_t \in \mathbb{R}^{2\times I\times J}$ and they have the same elements, excluding the dimensions. For the static graph, the adjacency matrix is $\mathbf{S}$ and the input feature is $\mathbf{H} = \sum_t \mathbf{H}_t$.

As shown in Fig. 2, we use L_{ORFL} HSDM components to perform representation learning for the dynamic graph and the static graph. Specifically, each HSDM component includes two GAT models, in which the multi-head attention scheme is introduced to stabilize the learning process. Fig. 4 illustrates the implementation details of our HSDM component and the multi-head attention scheme. For the l-th component, the output can be expressed as

$$\mathbf{H}_t^{(l)} = \big\Vert_{m=1}^{M} \sigma(\mathbf{A}_{d,m}^{(l)} \mathbf{H}_t^{(l-1)} \mathbf{W}_{d,m}^{(l)}), \tag{10}$$

where $\mathbf{A}_{d,m}^{(l)}$ and $\mathbf{W}_{d,m}^{(l)}$ denote the masked self-attention weights and learnable parameters of the m-th head in the l-th HSDM, respectively. For each node pair u and v in $\mathbf{A}_{d,m}^{(l)}$, the attention weight $a_{u,v}^{(l)}$ is calculated by

$$\begin{aligned} a_{u,v}^{(l)} &= \frac{\exp(e_{u,v}^{(l)})}{\sum_{k\in\mathcal{N}_{(u)}} \exp(e_{u,k}^{(l)})}, \\ e_{u,v}^{(l)} &= \sigma\{(\mathbf{h}_u^{(l)}\mathbf{W}_{d,m}^{(l)} \Vert \mathbf{h}_v^{(l)}\mathbf{W}_{d,m}^{(l)})\mathbf{a}^{(l)}\}, \end{aligned} \tag{11}$$

where $\mathbf{a}^{(l)}$ denotes a learnable parameter vector and $\Vert$ is the concatenation operator. The core idea here is to compute the hidden representations of each node in the graph by attending to its neighbors, following a self-attention strategy.

The representation learning for the static road network can be obtained similarly and we describe it as

$$\mathbf{H}^{(l)} = \big\Vert_{m=1}^{M} \sigma(\mathbf{A}_{s,m}^{(l)} \mathbf{H}^{(l-1)} \mathbf{W}_{s,m}^{(l)}). \tag{12}$$

where $\mathbf{A}_{s,m}^{(l)}$ and $\mathbf{W}_{s,m}^{(l)}$ are the masked self-attention weights and learnable parameters of the m-th head in the l-th component of HSDM, respectively.

After we obtain the representations of the dynamic graph and the static graph, we design a gating mechanism by using a mapping function $\phi : \mathbb{R}^{N\times F} \rightarrow \mathbb{R}^{I\times J\times F}$ and apply it to the traffic volume representations, and then these two types of representations are fused, resulting in the following feature representation

$$\mathbf{G}_t^{(l)} = \beta\phi(\mathbf{H}_t^{(l)}) + (1-\beta)\phi(\mathbf{H}^{(l)}), \tag{13}$$

where β is a predefined parameter that balances the different effects of the dynamic and static spatial dependences on the final representations. $\mathbf{G}_t^{(l)}$ denotes the learned representations for each region and reflects the spatial dependencies of traffic volume among different regions. Then $\mathbf{G}_t^{(l)}$ is modulated with the traffic feature and the learned representation of Eq. (9) can be updated as follows

$$\mathbf{X}_t^{(l)} = \sigma(\mathbf{W}_{\mathrm{TFL}}^{(l)} * \mathbf{X}_t^{(l-1)}) \odot \mathbf{G}_t^{(l)}, \tag{14}$$

where $\odot$ is the Hadamard product.

4.5 | Joint feature learning

The spatial and temporal correlations of traffic volume among different regions are quite challenging to capture as they are affected by many factors, such as geographic distance and urban layout. Some regions that are far away from each other may have high spatial correlations when they are similar functional urban areas. Thus, a deeper network is preferred to learn the representations of traffic volume and capture the long-distance spatial correlations. We design a JFL component and perform joint feature learning for $\mathbf{X}_t^{(L_{\text{TFL}})}$. As shown in Fig. 2, we stack learning blocks sequentially in the JFL component and the design of each block is detailed in Fig. 5. The blocks are connected by a convolutional layer, whose purpose is to reduce the dimension of feature maps, and consequently to reduce the number of parameters in our STOD-Net.

For each block $b \in \{1, 2, \cdots, B\}$, the input to it can be written as

$$\mathbf{Y}_{t,b} = (\mathbf{Y}_{t,b-1} * \mathbf{W}_{\text{JFL},b}) \parallel \mathbf{G}_t^{(L_{\text{TFL}})}, \quad (15)$$

where $\mathbf{W}_{\text{JFL},b}$ denotes the parameters in the convolutional layer of block b in the JFL component and $\mathbf{Y}_{t,0} = \mathbf{X}_t^{(L_{\text{TFL}})}$. Note that in Eq. (15) we transfer (FL component) the hybrid spatial representations from dynamic OD data and static road network to the JFL module by adding shortcut connections between the last HSDM block and the JFL component. This is because shortcut connections among layers of neural networks are beneficial for the optimization of deeper models[53]. They can alleviate gradient vanishing or exploding problems, and strengthen feature propagation, thereby improving the prediction performance of neural networks. We use concatenation instead of summation because concatenation can increase the variation in the input of subsequent layers and improve efficiency[54].

We then forward $\mathbf{Y}_{t,b}$ to the STM module for spatial temporal modeling. STM consists of L_{JFL} layers and each layer performs

$$\mathbf{Y}_{t,b}^{(l)} = \mathcal{H}_b^{(l)}(\mathbf{Y}_{t,b}^{(l-1)}; \mathbf{W}_{\text{JFL},b}^{(l)}), \quad (16)$$

where $\mathcal{H}_b^{(l)}$ represents the composite function of layer l in block b and it implements six consecutive operations: BN-ReLU-Conv(1×1)-BN-ReLU-Conv(3×3). Note that $\mathbf{Y}_{t,b}^{(0)} = \mathbf{Y}_{t,b}$ represents the initial input to block b.

To further capture channel-wise importance and make the learning process flexible, we employ a gating mechanism with a sigmoid activation on $\mathbf{Y}_{t,b}^{(l)}$. Particularly, we parameterize the gating mechanism by forming a bottleneck with two fully-connected (FC) layers:

$$\mathbf{C}_{t,b}^{(l)} = \text{sigmoid}(\mathbf{W}_{\text{FC},2}^{(l)}\, \sigma(\mathbf{W}_{\text{FC},1}^{(l)} \mathbf{Y}_{t,b}^{(l)})). \quad (17)$$

Then the output $\mathbf{Y}_{t,b}^{(l)}$ is updated as $\mathbf{Y}_{t,b}^{(l)} = \mathbf{Y}_{t,b}^{(l)} \odot \mathbf{C}_{t,b}^{(l)}$. After B blocks' representation learning, the final output of the JFL component is $\mathbf{Y}_{t,B}$.

4.6 | Final fusion

The above subsections summarize the detailed implementations of our STOD. As shown in the left of Fig. 2, there exists three inputs, i.e., $\{\mathbf{X}_t^c, \mathbf{D}_t^c\}$, $\{\mathbf{X}_t^p, \mathbf{D}_t^p\}$, and $\{\mathbf{X}_t^r, \mathbf{D}_t^r\}$. Thus three outputs can be obtained: $\mathbf{Y}_{t,B}^{(c)}$, $\mathbf{Y}_{t,B}^{(p)}$, and $\mathbf{Y}_{t,3}^{(r)}$. They denote the learned spatiotemporal representations for the closeness, periodicity, and trend dependencies, respectively. The prediction can be obtained after performing fusion on the three kinds of outputs

$$\hat{\mathbf{X}}_{t+1} = \text{sigmoid}(\mathbf{Y}_{t,B}^{(c)} + \mathbf{Y}_{t,B}^{(p)} + \mathbf{Y}_{t,B}^{(r)}), \quad (18)$$

where sigmoid$(\cdot)$ stands for the sigmoid activation function. We adopt mean squared error (MSE) as our objective function, thus the loss function in Eq. (1) can be detailed as

$$\mathcal{L} = \|\hat{\mathbf{X}}_{t+1} - \mathbf{X}_{t+1}\|_2^2, \quad (19)$$

where $\|\cdot\|_2^2$ denotes the square of Frobenius norm. The parameters of our STOD-Net, i.e., w, can be obtained by optimizing Eq. (19) over all the training dataset.

5 | EXPERIMENTAL RESULTS

In this section, we introduce two real-world datasets used in the experiments, evaluation metrics, and several baseline methods we compared to. We also explain the detailed experimental settings, followed by thorough experiment results and the corresponding analysis.

5.1 | Datasets and preprocessing

The two real-world datasets come from New York City (NYC), which are publicly available[55]. They are taxi and bike trip records and we denote them as NYC-Taxi and NYC-Bike, respectively. NYC-Taxi includes approximately 22.3 million trip records from Jan. 1st, 2015 to Mar. 1st, 2015. While for the NYC-Bike dataset, it includes about 2.2 million trip records from Jul. 1st, 2016 to Aug. 29, 2016.

The NYC is split into 10 × 20 regions, and each region's area is approximately 1 km^2. We set the time interval as 30 minutes and calculate the traffic volume in each region and the OD data from one region to another. We use the first 50 days

TABLE 1 Prediction performance of different methods on the NYC-Taxi and NYC-Bike datasets.

Data	Method	In Traffic			Out Traffic		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE
NYC-Taxi	HA	71.02	41.10	38.06%	59.90	32.55	36.23%
	Naive	36.96	22.72	22.94%	31.78	18.32	22.96%
	ARIMA	34.92	21.97	24.85%	29.99	18.12	25.26%
	LR	30.55	18.93	19.83%	25.66	15.12	19.66%
	MLP	30.09 ± 0.21	18.57 ± 0.13	$19.97 \pm 0.18\%$	24.69 ± 0.24	14.31 ± 0.12	$19.06 \pm 0.16\%$
	ST-ResNet	23.89 ± 0.16	15.25 ± 0.07	$17.16 \pm 0.07\%$	19.47 ± 0.09	12.14 ± 0.06	$16.67 \pm 0.07\%$
	STGCN	22.78 ± 0.20	14.29 ± 0.15	$16.67 \pm 0.31\%$	18.52 ± 0.15	11.54 ± 0.13	$16.56 \pm 0.31\%$
	STDN	22.32 ± 0.22	14.09 ± 0.18	$16.15 \pm 0.62\%$	18.08 ± 0.27	11.38 ± 0.20	$16.13 \pm 0.52\%$
	STOD-Net	$\mathbf{21.44 \pm 0.08}$	$\mathbf{13.37 \pm 0.04}$	$\mathbf{15.28 \pm 0.09\%}$	$\mathbf{17.61 \pm 0.08}$	$\mathbf{10.87 \pm 0.05}$	$\mathbf{15.33 \pm 0.10\%}$
NYC-Bike	HA	17.46	11.02	37.31%	16.72	10.69	35.54%
	Naive	14.03	9.48	31.25%	13.43	9.28	30.62%
	ARIMA	12.92	8.81	28.59%	12.38	8.60	27.84%
	LR	11.89	8.07	26.76%	11.21	7.74	25.69%
	MLP	9.41 ± 0.04	6.54 ± 0.02	$23.05 \pm 0.10\%$	8.54 ± 0.06	6.12 ± 0.03	$21.71 \pm 0.15\%$
	ST-ResNet	8.96 ± 0.03	6.46 ± 0.02	$22.72 \pm 0.06\%$	8.19 ± 0.04	6.08 ± 0.03	$21.49 \pm 0.09\%$
	STGCN	8.83 ± 0.18	6.35 ± 0.12	$22.42 \pm 0.45\%$	7.89 ± 0.14	5.86 ± 0.10	$20.76 \pm 0.37\%$
	STDN	8.61 ± 0.18	6.14 ± 0.13	$21.42 \pm 0.22\%$	7.78 ± 0.18	5.73 ± 0.12	$20.15 \pm 0.31\%$
	STOD-Net	$\mathbf{8.18 \pm 0.03}$	$\mathbf{5.87 \pm 0.02}$	$\mathbf{20.63 \pm 0.05\%}$	$\mathbf{7.39 \pm 0.03}$	$\mathbf{5.48 \pm 0.02}$	$\mathbf{19.39 \pm 0.03\%}$

of data to train our model and the last ten days of data to test the model's performance. Before training our model, both the traffic volume data and the OD data are scaled into $[0, 1]$ by using Min-Max normalization and rescaled back to their original scale during evaluation. Besides, traffic values that are less than a certain threshold are ignored when evaluating performance since low traffic is of little interest, and this assumption is also commonly adopted in industry and academia[31]. We follow the same settings in[9,31] and set the threshold to 10.

5.2 | Baselines and evaluation etrics

We compare our model, STOD-Net, with the following several baselines:

- Historical Average (HA) method. The next time slot's prediction is set to the average of all historical traffic values.
- Naive method. This method treats the last observation as the prediction of the next time slot.
- Auto-Regressive Integrated Moving Average (ARIMA) model. ARIMA is one of the most frequently used time series prediction approaches and can model the autocorrelations in traffic data.
- Linear Regression (LR). LR denotes a linear approach to traffic prediction.
- Multilayer Perceptron (MLP). MLP represents a class of feedforward artificial neural networks. MLP can be used to model the nonlinear relationships hidden in the traffic data.
- Spatio-Temporal Residual Network (ST-ResNet)[24]. ST-ResNet is a deep learning framework for solving city-wise traffic prediction and has become a seminal work since its publication. It can model the spatial and temporal dynamics of traffic volume simultaneously.
- Spatial-Temporal Graph Convolutional Networks (STGCN)[4]. Instead of applying regular convolutional units, STGCN formulates the traffic prediction on graphs and builds the model with complete convolutional structures, which enables a much faster training speed with fewer parameters.
- Spatio-Temporal Dynamic Network (STDN)[31]. STDN can model the different spatial dynamics among different locations and the temporal shifting in traffic data. It achieves state-of-the-art performance on both the NYC-Taxi and the NYC-Bike datasets.

We adopt three metrics to evaluate the prediction performance of different approaches. The metrics are root-mean-square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), respectively.

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{t=1}^{T'}\sum_{i=1}^{I}\sum_{j=1}^{J}(x_t^{(i,j)} - \hat{x}_t^{(i,j)})^2}, \tag{20}$$

$$\text{MAE} = \frac{1}{M}\sum_{t=1}^{T'}\sum_{i=1}^{I}\sum_{j=1}^{J}|x_t^{(i,j)} - \hat{x}_t^{(i,j)}|, \tag{21}$$

$$\text{MAPE} = \frac{1}{M}\sum_{t=1}^{T'}\sum_{i=1}^{I}\sum_{j=1}^{J}|\frac{x_t^{(i,j)} - \hat{x}_t^{(i,j)}}{x_t^{(i,j)}}| \tag{22}$$

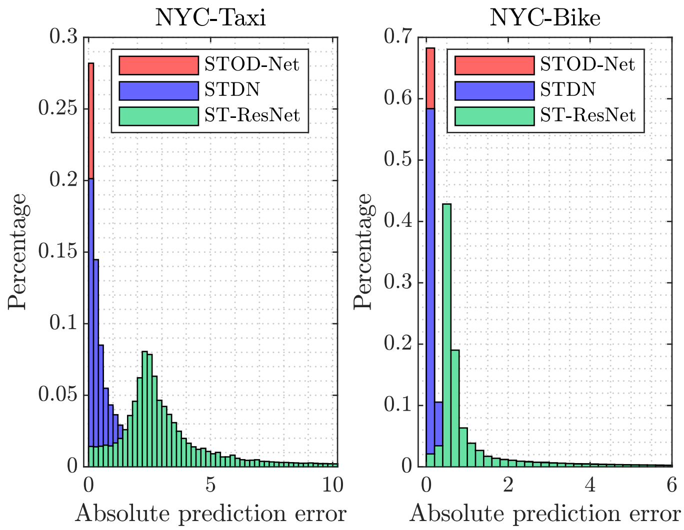


FIGURE 6 Histograms of absolute prediction errors.

where T' denotes the times slots in test dataset and $M = T' \times I \times J$. In addition, $x_t^{(i,j)}$ ($\hat{x}_t^{(i,j)}$) denotes the real (predicted) in/out traffic value at time slot t for region (i,j).

5.3 | Experimental settings

The experiment results are obtained with the following settings. For the lengths of closeness, periodicity, and trend sequences, we set them to 5, 4, 1, and these values are obtained based on a grid search over $L_c, L_p, L_r \in \{1,2,3,4,5\}$. When constructing the static road network, we set $\rho_z = \{0.4, 0.3, 0.2, 0.1\}$. In the TFL component, we use two convolutional layers for preliminary traffic representation learning, and the hidden representations in each layer are 24. In the ORFL component, there are two HSDM blocks, and each block has two M-heads GAT for the static and dynamic spatial dependencies modeling and M equals to 2. The parameter, β, that balances the different effects of the static and dynamic spatial dependences, is set to 0.5. In the JFL component, there are 3 blocks. Each block has $L_{\text{JFL}} = 8$ layers, and each layer outputs $K = 12$ hidden representations.

We use Adam to optimize our STOD-Net with a learning rate of 0.0001, and the learning rate is adjusted via the one-cycle policy. In addition, STOD-Net is trained using batch 128 for 500 epochs on the NYC-Taxi dataset and 100 epochs on the NYC-Bike dataset.

5.4 | City-wise prediction results

Table 1 summarizes the city-wise prediction performance comparisons of different methods on the NYC-Taxi and NYC-Bike datasets. To obtain Table 1, we carry out 10 independent experiments and report the mean and standard deviation ($\pm$) results of three evaluation metrics. The best results are marked in bold for clearness.

From this table we can clearly observe that 1) deep neural network-based models can normally achieve better prediction performance than linear models (LR and ARIMA), followed by simple forecasting methods (HA and Naive). For instance, the RMSE result on the in-traffic of the NYC-Taxi dataset improves from 36.96 (Naive) to 30.09 (MLP) to 22.32 (STDN), which validates the advantages of deep neural networks in modeling spatiotemporal dependencies of traffic volume. 2) Our proposed method, i.e., STOD-Net, achieves the best prediction performance in terms of RMSE, MAE, and MAPE, on both datasets. We take the MAPE metric on the NYC-Taxi dataset as an example to illustrate the effectiveness of STOD-Net. Specifically, the obtained MAPE results of STOD-Net for the in-traffic (out-traffic) are about 15.28% (15.33%), while the best baseline's (STDN) MAPE results are about 16.15% (16.13%). These numbers indicate approximately a 5.4% (5.0%) performance gain can be acquired by STOD-Net. For the other two metrics, that is, RMSE and MAE, similar performance gains can be obtained as well if we do the calculation, but we omit their specific values here for brevity.

Aside from lower prediction errors, STOD-Net remains stable for different experiments as it yields lower standard deviations than baselines on these three metrics. Consider STDN as an example, the standard deviations of MAPE on the NYC-Taxi dataset are 0.62 (in-traffic) and 0.52 (out-traffic), respectively. In comparison, STOD-Net's standard deviations of MAPE on the same dataset are approximately 0.09 (in-traffic) and 0.10 (out-traffic), respectively, which is much lower than STDN.

We also analyze the prediction errors in Fig. 6, which shows the histograms of the absolute prediction errors (APEs) of three methods, i.e., ST-ResNet, STDN, and STOD-Net. Fig. 6 left/right shows the results on the NYC-Taxi/NYC-Bike dataset. From this figure, we can see that for STOD-Net, a larger portion of prediction errors tend to be zero compared with ST-RseNet and STDN. The results of Fig. 6 demonstrate STOD-Net has lower prediction errors than its competitive counterparts.

The prediction results of Table 1 and Fig. 6 confirm the effectiveness of STOD-Net. Thus, we claim it yields better prediction performance than baselines. The reasons behind the success of STOD-Net can be attributed into twofold: 1) The introduced ORFL component models the hidden relationships of different regions, leading to an accurate characterization of their spatial dependencies. 2) The complex spatial and temporal dependencies among regions are simultaneously captured

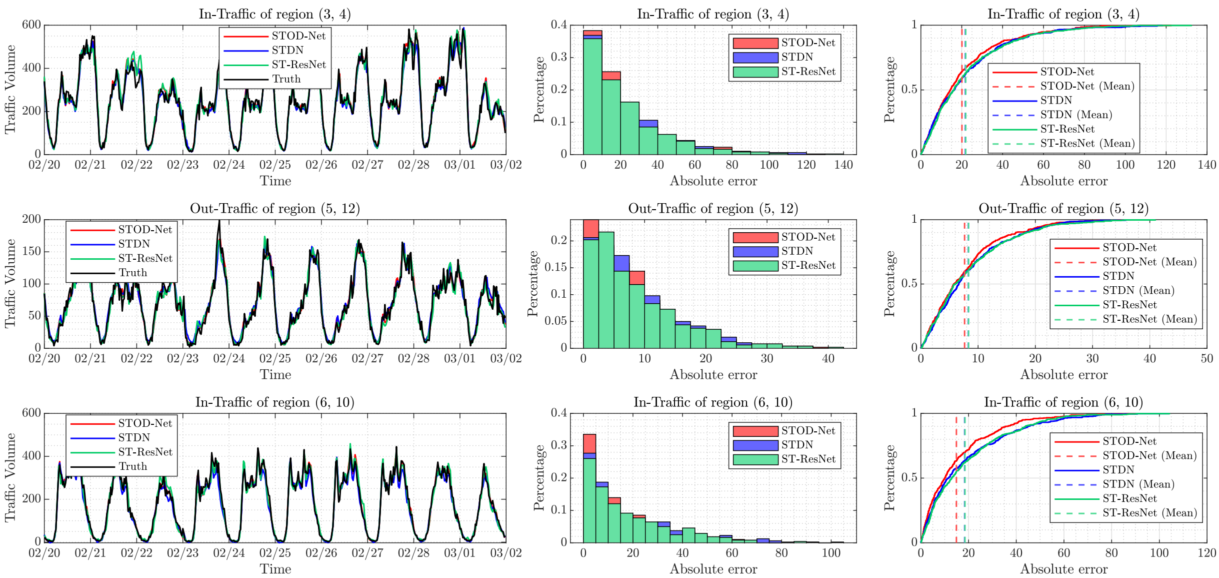


FIGURE 7 Predictions versus ground truth and the corresponding prediction error analysis for three randomly selected regions, i.e., $(3,4)$, $(5,12)$, and $(6,10)$, on the NYC-Taxi dataset.

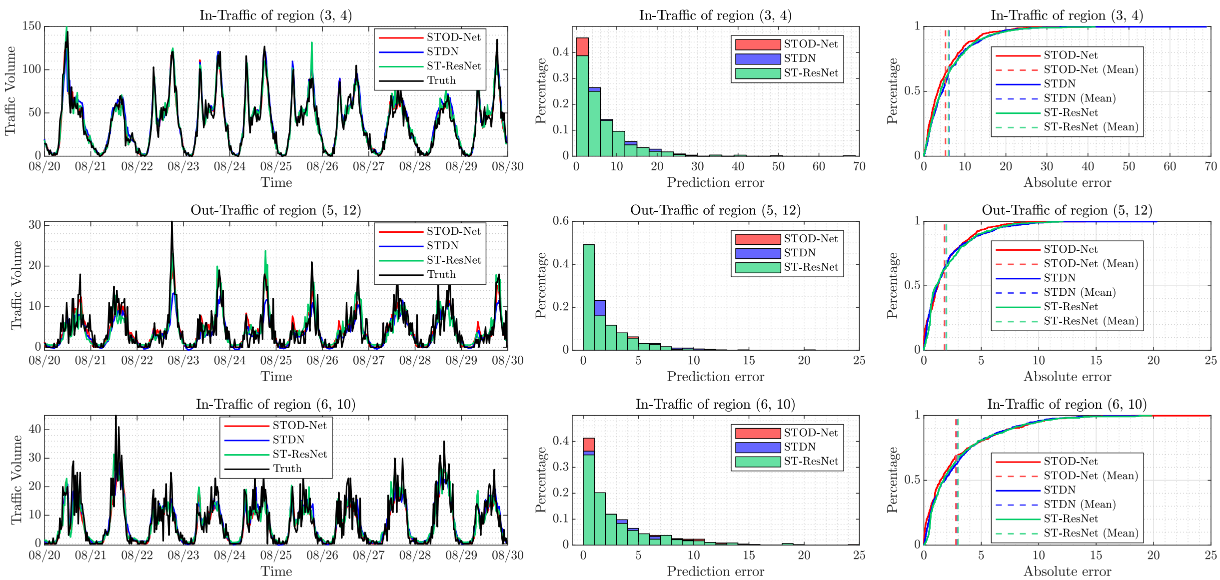


FIGURE 8 Predictions versus ground truth and the corresponding prediction error analysis for three randomly selected regions, i.e., $(3,4)$, $(5,12)$, and $(6,10)$, on the NYC-Bike dataset.

and learned by the JFL component, thus more effective representations can be obtained to enhance STOD-Net's prediction ability.

5.5 | Region-wise prediction results

The above subsection reports the overall quantitative prediction results for all regions. In this subsection, we compare the region-wise prediction performance of different methods. To be specific, we randomly select several regions and plot the ground truth values and the achieved predictions by different methods for these regions. In the following, we only compare STOD-Net with ST-ResNet and STDN as they generally obtain better performance than other baselines. Fig. 7 and Fig. 8 show the results of three randomly selected regions on the NYC-Taxi and NYC-Bike datasets, respectively. In Fig. 7 and Fig. 8, the left three subfigures compare the predictions versus ground truth values, the middle three subfigures show the

TABLE 2 Effectiveness of OD learning on prediction performance.

Dataset	Method	MSE	MAE	MAPE
NYC-Taxi	Basic	19.57	12.10	15.28%
	Basic+ORFL	19.33	11.92	15.09%
	Basic+ORFL+FT	19.22	11.90	15.08%
NYC-Bike	Basic	8.16	5.93	20.79%
	Basic+ORFL	8.01	5.83	20.48%
	Basic+ORFL+FT	7.88	5.73	19.97%

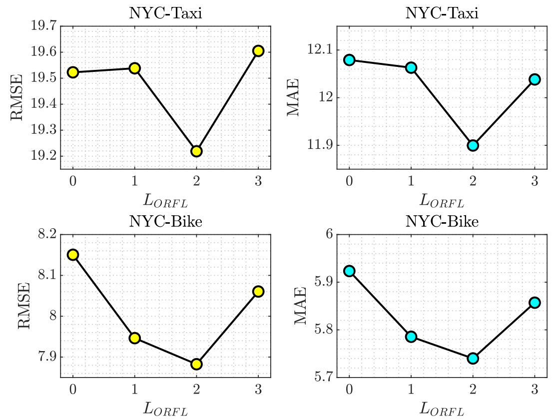


FIGURE 9 The number of HSDM blocks versus the prediction performance.

distributions of APEs, and the right three subfigures illustrate the cumulative distribution functions (CDFs) of APEs and the mean prediction errors.

From these two figures, we can observe that 1) The predictions of STOD-Net and STDN as well as ST-ResNet all match the ground truth values fairly well, though distinct traffic patterns exist in different regions or datasets. For instance, on the NYC-Taxi dataset, the traffic pattern of the region $(6,10)$ is different from that of the region $(5,12)$, whose peak traffic has a clear delay. Besides, region $(6,10)$ has multiple peak periods, while region $(5,12)$ tends to have a single peak period for most of the days.

Nevertheless, all three methods achieve competitive prediction results, and this validates the superiority of convolutional neural networks for urban traffic prediction. 2) STOD-Net method achieves better region-wise prediction results than its two counterparts, which can be verified by the histograms and CDFs of APEs. Take the region $(6,10)$ on the NYC-Taxi dataset as an example. From the histograms of APEs, we can perceive that STOD-Net has more prediction errors tending to zero compared with STDN and ST-ResNet. This can be more quantitatively reflected by the CDFs of APEs. More specifically, about 72% of APEs are less than 20 for the STOD-Net, while they are 66% and 64% for the STDN and ST-ResNet, respectively. The mean values of APEs for the STOD-Net, STDN, and ST-ResNet are $14.91, 18.35$, and 18.50, respectively.

Similar results can be obtained for the other regions on the NYC-Taxi and NYC-Bike datasets. Thus, we can conclude that our proposed STOD-Net achieves better region-wise prediction results than baselines.

5.6 | Impacts of OD and road feature learning

In this subsection, we go one step further and analyze the impacts of OD and road feature learning on the prediction performance of STOD-Net. In particular, we report the prediction results with and without OD and road feature learning to demonstrate the advantages of introducing OD feature learning for urban traffic prediction. Table 2 summarizes the obtained overall results on the NYC-Taxi and NYC-Bike dataset, without distinguishing the in-traffic and out-traffic. In this table, there are three versions of STOD-Net, i.e., Basic version, Basic+ORFL version, and Basic+ORFL+FT version. The 'Basic' denotes we only use the TFL and JFL components for traffic prediction without OD feature. The 'Basic+ORFL' means the ORFL component is further introduced into traffic prediction. The 'Basic+ORFL+FT' denotes both the ORFL and the FT components introduced into traffic prediction.

From Table 2 we know that the 'Basic' version of STOD-Net performs the worst among all the three versions. With the introduction of ORFL, the prediction errors on both datasets become lower. Besides, the ORFL component further improves prediction performance. The results in Table 2 indicate that the OD data can indeed provide additional performance gains and servers as a new direction to consider for urban traffic prediction.

5.7 | Impacts of hyper parameters

In this subsection, we explore the impacts of several related hyper-parameters on the prediction performance of STOD-Net. For example, the number of HSDM blocks and the parameter β that balances the different effects of the dynamic and static spatial dependencies.

5.7.1 | The number of HSDM blocks

We vary the number of HSDM blocks $L_{\text{ORFL}} \in \{0,1,2,3\}$ and present the obtained RMSE and MAE results on the NYC-Taxi and NYC-Bike datasets in Fig. 9. Note that when $L_{\text{ORFL}} = 0$,

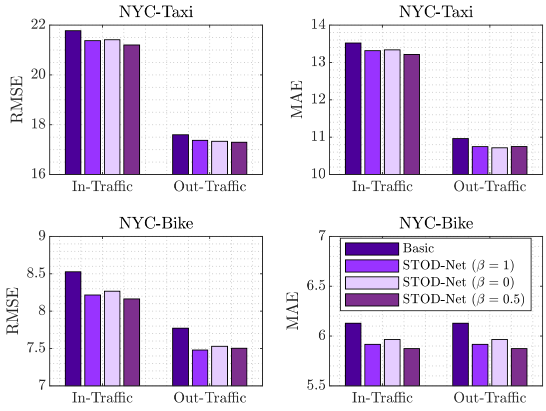


FIGURE 10 The influence of gate mechanism on prediction performance.

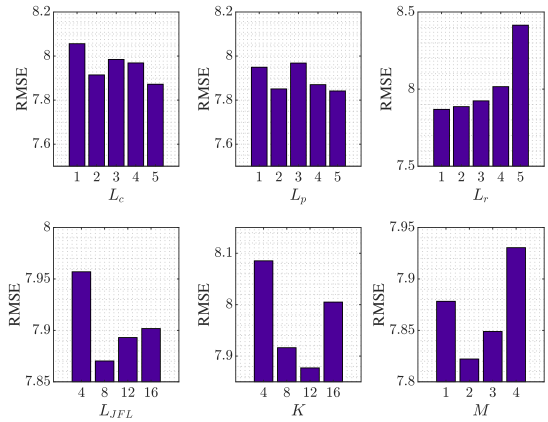


FIGURE 11 The influence of hyper-parameters on prediction performance.

STOD-Net degenerates into the 'Basic' version of it, which is explained in the last subsection. We can observe from Fig. 9 that with the increase of L_{ORFL}, the performances regarding the RMSE and MAE improve gradually till to their corresponding minimums, after which the performances decrease. The results of Fig. 9 are consistent with the results of Table 2 and demonstrate again the usefulness of OD feature learning. Besides, based on the results we can conclude that the number of HSDM blocks should not be too large, as larger L_{ORFL} may break the TFL component's representation learning, thus leading to performance degradation.

5.7.2 | The parameter β

The parameter β in Eq. (13) is a predefined value, which affects the gating mechanism. Actually, it balances the contribution of the dynamic and static spatial dependences to the final representations. We set $\beta \in \{0, 0.5, 1\}$ and report the obtained results in Fig. 10. Be noted that when $\beta = 0$, it means we only consider the static spatial dependence. Similarly, when $\beta = 1$, it means we only consider the dynamic spatial dependence. When $\beta = 0.5$, we consider both the static and dynamic spatial dependencies, and they contribute to the final representations equally. We can see that the prediction performances vary when the value of β changes. Nonetheless, the gating mechanism is beneficial to predictions regardless of its values, as it achieves better performance than the 'Basic' version of STOD-Net. Additionally, we also observe that considering both the static and dynamic spatial dependencies, that is, $\beta = 0.5$, can generally obtain lower prediction errors than only considering the static or dynamic spatial dependence. Because the static and dynamic OD data contain complementary information for capturing spatial dependencies, thus modeling them together enhances the prediction performance.

5.7.3 | Others

There are several other hyper-parameters that will influence the prediction performance of STOD-Net, and we give the experiment results here. Six hyper-parameters are considered, that is, the length of closeness dependence L_c, the length of periodicity dependence L_p, the length of trend dependence L_r, the number of layers in each dense block L_{JFL}, the output representations in each convolutional layer of dense block K, and the number of heads in GAT M. Their influence on prediction results is illustrated in Fig. 11. Note that we only report the RMSE results on one dataset (NYC-Bike) as other metrics have similar results to Fig. 11. It can be observed from Fig. 11 that with the increase of L_c and L_p, the prediction performance tends to be improved because more data are used to model temporal dependence. But for L_r, as its increases, the performance degrades gradually. This is because the increase of L_r will greatly reduce the number of training samples, thus under-fitting may occur.

For the number of layers in each dense block L_{JFL}, STOD-Net has a large capacity as its increases, leading to performance improvement. But too large L_{JFL} will make our model too strong to over-fitting the data, thus performance becomes worse. This phenomenon holds for the hyper-parameters K and M. These hyper-parameters control the capacity of STOD-Net. Larger values indicate large capacity and strong representation ability. But too large values can easily cause over-fitting and reduce the generalization of STOD-Net. Thus the best hyper-parameters can be obtained based on a grid search strategy.

6 | CONCLUSION

In this paper, we proposed STOD-Net, a spatial temporal origin-destination feature enhanced deep neural network, to solve urban traffic prediction. Beyond modeling the historical traffic itself, we introduced OD data into the prediction and adopted graph neural networks to model them to capture the inter-regional spatial interaction patterns. We consider two types of OD data, namely, static and dynamic OD data, in STOD-Net and fused them in a weighted fashion to capture different regional spatial dependences. Extensive experiments were conducted on two real-world datasets, and the results demonstrated that STOD-Net achieves superior prediction performance than state-of-the-art methods. Possible future directions include considering more fine-grained road network information and exploring the trade-off between the number of parameters and the prediction performance.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

Research data are available upon request.

REFERENCES

1. Fan X, Xiang C, Chen C, et al. BuildSenSys: Reusing Building Sensing Data for Traffic Prediction With Cross-Domain Learning. *IEEE Transactions on Mobile Computing.* 2021;20(6):2154-2171.
2. Tedjopurnomo DA, Bao Z, Zheng B, Choudhury FM, Qin AK. A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges. *IEEE Transactions on Knowledge and Data Engineering.* 2022;34(4):1544-1561. doi: 10.1109/TKDE.2020.3001195
3. Zheng Y, Capra L, Wolfson O, Yang H. Urban Computing: Concepts, Methodologies, and Applications. *ACM Transactions on Intelligent Systems and Technology.* 2014;5(3):1–55.
4. Yu B, Yin H, Zhu Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *Proceedings of the 27th International Joint Conference on Artificial Intelligence.* 2018:3634–3640.
5. AlQerm I, Shihada B. Energy Efficient Traffic Offloading in Multi-Tier Heterogeneous 5G Networks Using Intuitive Online Reinforcement Learning. *IEEE Transactions on Green Communications and Networking.* 2019;3(3):691-702. doi: 10.1109/TGCN.2019.2916900
6. AlQerm I, Shihada B. A cooperative online learning scheme for resource allocation in 5G systems. *2016 IEEE International Conference on Communications (ICC).* 2016:1-7. doi: 10.1109/ICC.2016.7511617
7. Elwhishi A, Ho PH, Naik K, Shihada B. Self-Adaptive Contention Aware Routing Protocol for Intermittently Connected Mobile Networks. *IEEE Transactions on Parallel and Distributed Systems.* 2013;24(7):1422-1435. doi: 10.1109/TPDS.2012.23
8. Zhang L, Zhang C, Shihada B. Efficient Wireless Traffic Prediction at the Edge: A Federated Meta-Learning Approach. *IEEE Communications Letters.* 2022;26(7):1573-1577. doi: 10.1109/LCOMM.2022.3167813
9. Yao H, Wu F, Ke J, et al. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence.* 2018;32(1):2588–2595.
10. Zhang C, Dang S, Shihada B, Alouini MS. Dual Attention-Based Federated Learning for Wireless Traffic Prediction. *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications.* 2021:1-10. doi: 10.1109/INFOCOM42981.2021.9488883
11. Zhang D, Feng X. Dynamic Auto-structuring Graph Neural Network: A Joint Learning Framework for Origin-Destination Demand Prediction. *IEEE Transactions on Knowledge and Data Engineering.* 2023;35(4):3699-3711.
12. Qi T, Li G, Chen L, Xue Y. ADGCN: An Asynchronous Dilation Graph Convolutional Network for Traffic Flow Prediction. *IEEE Internet of Things Journal.* 2022;9(5):4001-4014.
13. Guo S, Lin Y, Feng N, Song C, Wan H. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence.* 2019;33(01):922–929.
14. Pan Z, Liang Y, Wang W, Yu Y, Zheng Y, Zhang J. Urban Traffic Prediction from Spatio-Temporal Data Using Deep Meta Learning. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2019:1720–1730.
15. Polson NG, Sokolov VO. Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies.* 2017;79:1–17.
16. Kumar SV. Traffic Flow Prediction using Kalman Filtering Technique. *Procedia Engineering.* 2017;187:582–587.
17. Zare Moayedi H, Masnadi-Shirazi MA. ARIMA model for network traffic prediction and anomaly detection. *2008 International Symposium on Information Technology.* 2008;4:1-6. doi: 10.1109/ITSIM.2008.4631947
18. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature.* 2015;521(7553):436–444.
19. Lv Y, Duan Y, Kang W, Li Z, Wang F. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems.* 2015;16(2):865–873.
20. Zheng C, Fan X, Wang C, Qi J. GMAN: A Graph Multi-Attention Network for Traffic Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence.* 2020;34(1):1234–1241.
21. Zhang J, Zheng Y, Sun J, Qi D. Flow Prediction in Spatio-Temporal Networks Based on Multitask Deep Learning. *IEEE Transactions on Knowledge and Data Engineering.* 2020;32(3):468-478. doi: 10.1109/TKDE.2019.2891537
22. Zhang C, Zhang H, Qiao J, Yuan D, Zhang M. Deep Transfer Learning for Intelligent Cellular Traffic Prediction Based on Cross-Domain Big Data. *IEEE Journal on Selected Areas in Communications.* 2019;37(6):1389–1401.
23. Zhang J, Zheng Y, Qi D, Li R, Yi X. DNN-Based Prediction Model for Spatio-Temporal Data. *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* 2016:1–4. doi: 10.1145/2996913.2997016
24. Zhang J, Zheng Y, Qi D. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence.* 2017:1655–1661.
25. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. *arXiv preprint arXiv:1710.10903.* 2017.
26. Chen J, Ma T, Xiao C. FastGCN: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247.* 2018.
27. Qi S, Wang W, Jia B, Shen J, Zhu SC. Learning Human-Object Interactions by Graph Parsing Neural Networks. *Proceedings of the European Conference on Computer Vision (ECCV).* 2018.
28. Wang W, Lu X, Shen J, Crandall DJ, Shao L. Zero-Shot Video Object Segmentation via Attentive Graph Neural Networks. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).* 2019.
29. Fan L, Wang W, Huang S, Tang X, Zhu SC. Understanding Human Gaze Communication by Spatio-Temporal Graph Reasoning. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).* 2019.
30. Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine.* 2017;34(4):18–42.
31. Yao H, Tang X, Wei H, Zheng G, Li Z. Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence.* 2019;33(01):5668–5675.

32. Wang S, Zhang M, Miao H, Peng Z, Yu PS. Multivariate Correlation-Aware Spatio-Temporal Graph Convolutional Networks for Multi-Scale Traffic Prediction. *ACM Transactions on Intelligent Systems and Technology.* 2022;13(3).
33. Cui Z, Henrickson K, Ke R, Wang Y. Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting. *IEEE Transactions on Intelligent Transportation Systems.* 2020;21(11):4883–4894.
34. Ahmed MS, Cook AR. Analysis of freeway traffic time-series data by using Box-Jenkins techniques. *Transportation Research Record.* 1979;722.
35. Williams BM. Multivariate Vehicular Traffic Flow Prediction: Evaluation of ARIMAX Modeling. *Transportation Research Record.* 2001;1776(1):194–200.
36. Lee S, Fambro DB. Application of Subset Autoregressive Integrated Moving Average Model for Short-Term Freeway Traffic Volume Forecasting. *Transportation Research Record.* 1999;1678(1):179–188.
37. Dai X, Fu R, Zhao E, et al. DeepTrend 2.0: A light-weighted multi-scale traffic prediction model using detrending. *Transportation Research Part C: Emerging Technologies.* 2019;103:142–157.
38. Wu CH, Ho JM, Lee D. Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems.* 2004;5(4):276-281. doi: 10.1109/TITS.2004.837813
39. Zhang L, Liu Q, Yang W, Wei N, Dong D. An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction. *Procedia - Social and Behavioral Sciences.* 2013;96:653–662.
40. Cai P, Wang Y, Lu G, Chen P, Ding C, Sun J. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transportation Research Part C: Emerging Technologies.* 2016;62:21–34.
41. Leshem G, Ritov Y. Traffic flow prediction using AdaBoost algorithm with random forests as a weak learner. *Proceedings of world academy of science, engineering and technology.* 2007;19:193–198.
42. Fu R, Zhang Z, Li L. Using LSTM and GRU neural network methods for traffic flow prediction. *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC).* 2016:324–328.
43. Cui Z, Ke R, Pu Z, Wang Y. Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. *arXiv preprint arXiv:1801.02143.* 2018:1-11.
44. Yang B, Sun S, Li J, Lin X, Tian Y. Traffic flow prediction using LSTM with feature enhancement. *Neurocomputing.* 2019;332:320–327.
45. Zhao L, Song Y, Zhang C, et al. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems.* 2020;21(9):3848-3858. doi: 10.1109/TITS.2019.2935152
46. Du B, Hu X, Sun L, Liu J, Qiao Y, Lv W. Traffic Demand Prediction Based on Dynamic Transition Convolutional Neural Network. *IEEE Transactions on Intelligent Transportation Systems.* 2021;22(2):1237-1247.
47. Yao X, Gao Y, Zhu D, Manley E, Wang J, Liu Y. Spatial Origin-Destination Flow Imputation Using Graph Convolutional Networks. *IEEE Transactions on Intelligent Transportation Systems.* 2021;22(12):7474-7484.
48. Lin Z, Feng J, Lu Z, Li Y, Jin D. DeepSTN+: Context-Aware Spatial-Temporal Neural Network for Crowd Flow Prediction in Metropolis. *Proceedings of the AAAI Conference on Artificial Intelligence.* 2019;33(01):1020–1027.
49. Chen T, Nie L, Pan J, Tu L, Zheng B, Bai X. Origin-Destination Traffic Prediction based on Hybrid Spatio-Temporal Network. *2022 IEEE International Conference on Data Mining (ICDM).* 2022:879-884. doi: 10.1109/ICDM54844.2022.00101
50. Li F, Feng J, Yan H, et al. Dynamic Graph Convolutional Recurrent Network for Traffic Prediction: Benchmark and Solution. *ACM Trans. Knowl. Discov. Data.* 2023;17(1). doi: 10.1145/3532611
51. Wang S, Miao H, Chen H, Huang Z. Multi-Task Adversarial Spatial-Temporal Networks for Crowd Flow Prediction. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.* 2020:1555–1564.
52. Zhang Y, Wang S, Chen B, Cao J, Huang Z. TrafficGAN: Network-Scale Deep Traffic Prediction With Generative Adversarial Nets. *IEEE Transactions on Intelligent Transportation Systems.* 2021;22(1):219-230.
53. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016:770–778. doi: 10.1109/CVPR.2016.90
54. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely Connected Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2017:2261–2269. doi: 10.1109/CVPR.2017.243
55. City of New York . NYC OpenData. 2020.