Dynamic and Multi-faceted Spatio-temporal Deep Learning for Traffic Speed Forecasting

Liangzhe Han[1], Bowen Du[1,2], Leilei Sun[1,2*], Yanjie Fu[3], Yisheng Lv[4], Hui Xiong[5]
[1]State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China
[2]Peng Cheng Laboratory, Shenzhen 518055, China
[3]Department of Computer Science, University of Central Florida, FL 32816, USA
[4]Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[5]Department of Management Science and Information Systems, Rutgers University, NJ 07102, USA
[1]{liangzhehan, dubowen, leileisun}@buaa.edu.cn, [3]yanjie.fu@ucf.edu, [4]yisheng.lv@ia.ac.cn, [5]hxiong@rutgers.edu

ABSTRACT

Dynamic Graph Neural Networks (DGNNs) have become one of the most promising methods for traffic speed forecasting. However, when adapting DGNNs for traffic speed forecasting, existing approaches are usually built on a static adjacency matrix (no matter predefined or self-learned) to learn spatial relationships among different road segments, even if the impact of two road segments can be changeable dynamically during a day. Moreover, the future traffic speed cannot only be related with the current traffic speed, but also be affected by other factors such as traffic volumes. To this end, in this paper, we aim to explore these dynamic and multi-faceted spatio-temporal characteristics inherent in traffic data for further unleashing the power of DGNNs for better traffic speed forecasting. Specifically, we design a dynamic graph construction method to learn the time-specific spatial dependencies of road segments. Then, a dynamic graph convolution module is proposed to aggregate hidden states of neighbor nodes to focal nodes by message passing on the dynamic adjacency matrices. Moreover, a multi-faceted fusion module is provided to incorporate the auxiliary hidden states learned from traffic volumes with the primary hidden states learned from traffic speeds. Finally, experimental results on real-world data demonstrate that our method can not only achieve the state-of-the-art prediction performances, but also obtain the explicit and interpretable dynamic spatial relationships of road segments.

CCS CONCEPTS

• **Information systems → Data mining**.

KEYWORDS

traffic speed forecasting, graph construction, graph convolution

ACM Reference Format:
Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, Hui Xiong. 2021. Dynamic and Multi-faceted Spatio-temporal Deep Learning for Traffic Speed Forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3447548.3467275

* Corresponding Author.

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467275

1 INTRODUCTION

In recent years, intelligent transportation systems have been largely promoted by deep learning techniques. The successful applications include but not limited to computer vision methods for traffic data collection [9] and regulations enforcement [2], reinforcement learning for automatic pilot [16, 18] and traffic signal control [6], big data analysis for human mobility modelling [10, 11], etc.

Among these techniques, deep spatio-temporal neural networks are the most promising methods due to its ability to capture spatial dependencies and temporal dynamics simultaneously, which have been widely used in traffic speed forecasting, transportation demand prediction and travel time estimation. In the early stage, with the distributions of traffic status or demands divided into grids on a map, CNNs were used to learn the spatial interactions of different grids, and RNNs were used to learn the evolutionary traffic dynamics [28, 30, 33]. Not long after that, researchers found that it is more reasonable to model the spatial interactions of traffic systems as graph due to the fact that most of the traffic indicators are given along road network (such as traffic speeds and volumes of road segments) or associated with fixed stations (pick-up demands of car-hailing stations or check-in amounts of subway stations). Most of the recent research has focused on geometric spatio-temporal learning [1, 14, 19, 26, 27, 31, 32, 35].

Traffic speed forecasting is a well-defined representative geometric spatio-temporal learning problem, which encodes each road segment as a node in a graph, and the edges between nodes correspond to spatial influences of road segments. A number of methods have been proposed for traffic speed forecasting, for example, Yu et al. proposed STGCN to integrate graph convolution and gated temporal convolution [32], Li et al. designed DCRNN to use bidirectional graph random walk to model spatial dependency and recurrent neural network to capture the temporal dynamics [19], Wu et al. proposed MTGNN to exploit the inherent dependency relationships among multiple segments [26].

Even though, two key issues have rarely been discussed. The first problem is how to model the spatial dependencies of road segments dynamically. In most of the existing research, the impacts of two road segments are viewed as static, and determined by a predefined

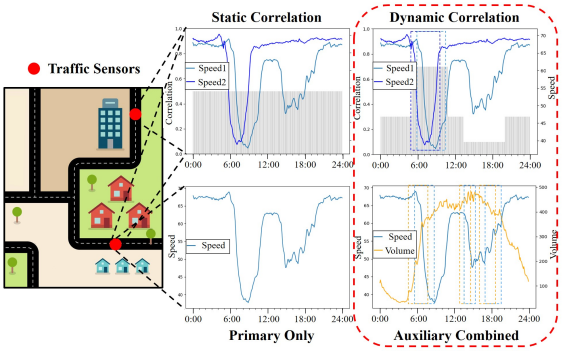


Figure 1: Example of dynamic spatial dependencies and the potential of auxiliary data to assist traffic speed forecasting.

or self-learned adjacency matrix. However, the interactions of two road segments at different time are supposed to be different. As traffic speed of two segments (top right) in Figure 1 shows, the speed pattern of them is similar in morning peak while completely different in evening peak. The second problem is how to incorporate the traffic speed forecasting problem with rich multi-faceted auxiliary data. Urban traffic networks are complex, dynamic and giant systems. Different types of traffic data record the observations of traffic systems from different perspectives. Therefore, it is promising to improve the performance of traffic speed forecasting by mining the latent patterns and dynamics underlying traffic systems deeply and sufficiently from multi-faceted data. As the combination of traffic speed and traffic volume (bottom right) in Figure 1 shows, traffic speed have a stiff drop after traffic volume increase rapidly in morning and evening peak, which reveals the potential of traffic volume to assist traffic speed forecasting.

This paper aims to address the above two issues. However, it is a nontrivial endeavor to design such a dynamic and multi-faceted traffic speed forecasting method due to the following challenges: First, traffic speed of multiple road segments follows a dynamic and implicit spatial pattern. Hand-designed spatial graph is easy to interpret but could not really reflect the interaction relationship. Second, generating a graph for each time slot through existing self-learned methods would introduce plenty of parameters making it hard to coverage. Third, it is difficult to model the effects of auxiliary information on traffic speed forecasting, as the impact may also have spatio-temporal characteristics.

In this paper, we propose a Dynamic and Multi-faceted Spatio-Temporal Graph Convolution Network (DMSTGCN) for traffic speed forecasting with multi-faceted auxiliary data. First, aiming to model dynamic spatial relationship between segments, a dynamic graph constructor inspired by tensor decomposition is designed to take periodicity and dynamic characteristics of traffic into consideration. Moreover, dynamic graph convolution is proposed to capture varying spatial dependencies of road segments. Then, to handle unique spatio-temporal dependencies in multi-faceted data, several parallel structures are employed, each of which consists of several dynamic graph convolutions and temporal convolution layers. For extra effects of auxiliary data on traffic speed, we design a framework powered by the multi-faceted fusion module to integrate auxiliary hidden states with traffic speed hidden states in a graph perspective. Finally, skip connections are added after each temporal convolution layer to several fully connected layers for final prediction. It is worth noting that the proposed method in this paper is a general framework to handle other problem with multi-faceted spatio-temporal graph sequences.

In summary, our contributions are three folds:

- A dynamic spatial dependencies learning method is proposed. Different from the existing methods, which are founded on a predefined or self-learned static adjacency matrix, our method propagates hidden states of nodes according to dynamic spatial relationships. We design a dynamic graph constructor and the dynamic graph convolution method to accomplish this.
- A multi-faceted fusion module is provided to incorporate the auxiliary hidden states with primary hidden states spatially and temporally. It is a generic framework to handle multi-faceted spatio-temporal data.
- We not only validate the effectiveness of the proposed methods by experiments on real-world datasets, but also uncover the explicit and discovered dynamic patterns of relationships among multi-faceted data.

2 PRELIMINARIES

In this section, we first introduce notations and preliminaries used in this paper. Then we formalize the problem of traffic speed forecasting with multi-faceted data.

2.1 Notations and Definitions

Definition 2.1 (Traffic Speed). Traffic speed is a crucial feature of road segment to describe the traffic condition. In this paper, we denote the set of segments with historical speed data as $\mathbb{V}_p = \{v_1, v_2, \cdots, v_{N_p}\}$, where N_p is the number of segments, and traffic speed of i-th segment at time slot t as $x^p_{i,t} \in \mathbb{R}$. Moreover, we use $\mathbf{X}^p_t \in \mathbb{R}^{N_p}$ to represent all traffic speed observations at time slot t.

Definition 2.2 (Auxiliary Feature). There exists latent correlations between different traffic features such as traffic speed and traffic volume. Although some features are not helpful in certain application directly, they can benefit prediction of traffic speed by some reasons. In this paper, the set of segments with historical auxiliary feature is denoted as $\mathbb{V}_a = \{s_1, s_2, \cdots, s_{N_a}\}$, where N_a is the number of segments, and auxiliary feature of i-th segment at time slot t as $x^a_{i,t} \in \mathbb{R}$. Similar to traffic speed, we use $\mathbf{X}^a_t \in \mathbb{R}^{N_a}$ as auxiliary feature of all segments at time slot t.

Definition 2.3 (Dynamic Graph). Correlations between different segments are dynamic which vary from dawn to dusk. This motivates us to build the dynamic graph, which contains one set of segments as its nodes but has different edges at each time. For time slot t, the traffic graph is denoted as $\mathcal{G}_t = \{\mathbb{V}, \mathbb{E}^t\}$, where $\mathbb{V}$ is the set of nodes and $\mathbb{E}^t$ is the set of edges. $e_{t,i,j} = (v_i, v_j, \mathbf{A}_{t,i,j}) \in \mathbb{E}^t$ means there is an edge pointing from v_j to v_i at time slot t with edge weight $\mathbf{A}_{t,i,j}$, the (t, i, j)-th entry of a 3-order adjacency tensor $\mathbf{A}$. Moreover, $\mathcal{G}_t$ can be represented in form of matrix $\mathbf{A}_t$, and the dynamic graph can be represented by $\mathbf{A}$.

2.2 Problem Formalization

Traffic speed forecasting aims to predict traffic speed for all segments with historical speed in a period of future time. In this paper, we further consider effects of historical auxiliary data. Formally, given P historical steps of traffic speed and P historical steps of the auxiliary feature, our goal is to learn a model f to predict future Q steps of traffic speed,

$$\hat{\mathbf{X}}^p_{t+1:t+Q} = f(\mathbf{X}^p_{t-P+1:t}, \mathbf{X}^a_{t-P+1:t}).$$

In this paper, we focus on traffic speed forecasting so we also call traffic speed as "primary feature" in the last of this paper.

3 METHODOLOGY

In this section, we will introduce our proposed model in detail. The overall framework of our proposed model is shown in Figure 2 which can be roughly divided into four parts: the dynamic graph constructor, the primary part, the auxiliary part and the multi-faceted fusion module.

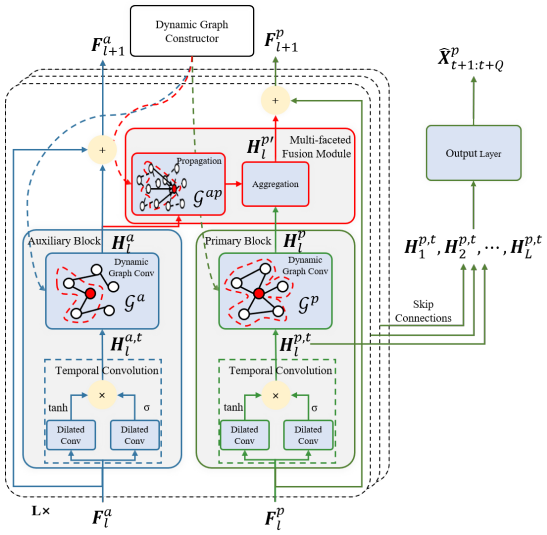


Figure 2: Architecture of DMSTGCN. The framework mainly consists of two parallel parts: one for primary feature and one for auxiliary feature. The dynamic graph constructor generates dynamic graphs for graph convolution layers. The primary part includes L primary blocks each of which consists of a temporal convolution layer and a dynamic graph convolution. The auxiliary part is similar to primary part. The multi-faceted fusion module first takes output of the auxiliary block into propagation and aggregates it with output of primary block for input of the next layer. Residual links are added for each block and hidden states after each temporal convolution of the primary part are connected to the output layer for final prediction.

3.1 Dynamic Graph Construction

As spatial dependencies between segments are implicit and changing constantly due to dynamic characteristic of traffic, it is necessary to properly design a graph learning module to address this. In previous work, dependencies between nodes were either defined by human knowledge like distance and functional similarity or generated as a static graph which omits the dynamic characteristic of traffic spatial dependencies. In this paper, to capture varying relationships between nodes, we propose the dynamic graph constructor which produces dynamic learnable graphs used in the dynamic graph convolution and the multi-faceted fusion module.

However, it is nontrivial to adaptively model relationships between each pair of nodes at each time. The simplest way is to directly assign a learnable parameter tensor to represent dynamic relationships, but the complexity of this method is $O(TNN)$, where T is the number of total time slots and N is the number of nodes, making it hard to compute and coverage. Considering the periodicity of traffic status, we assume that traffic at the same time in a day could share a same graph. Thus, our goal is transferred to an adjacency tensor $\mathbf{A} \in \mathbb{R}^{N_t \times N \times N}$ where N_t is the number of time slots in a day. And the graph at time slot t is $\mathbf{A}_{\phi(t)}$ where $\phi(t)$ is a function to get time in a day. But the complexity of this solution is still $O(N_t NN)$ which grows quadratically as the graph becomes larger. In reality, some structures of dynamic graph could be shared across time and space. For example, two adjacent segments could be correlated across the day, and people will leave from different resident areas to different work areas in the morning. So we propose a method inspired by Tucker decomposition [24] to form the adjacency tensor as shown in Figure 3. In traffic speed forecasting, instead of decomposing an known tensor, we aim to compose an unknown tensor with learnable parameters for spatial dependencies. This procedure performs reversely as original Tucker decomposition and updates dynamic graph by backpropagation during training.

Specifically, we assign three learnable matrices and one learnable core tensor including embedding of time slots $\mathbf{E}^t \in \mathbb{R}^{N_t \times d}$, embedding of source nodes $\mathbf{E}^s \in \mathbb{R}^{N_s \times d}$, embedding of target nodes $\mathbf{E}^e \in \mathbb{R}^{N_e \times d}$ and a core tensor $\mathbf{E}^k \in \mathbb{R}^{d \times d \times d}$, where N_t, N_s, N_e, d represent the number of time slots, number of original nodes, number of target nodes, and embedding dimension respectively. And the adjacency tensor is calculated as:

$$\mathbf{A}'_{t,i,j} = \sum_{o=1}^{d}\sum_{q=1}^{d}\sum_{r=1}^{d} \mathbf{E}^k_{o,q,r}\mathbf{E}^t_{t,o}\mathbf{E}^e_{i,q}\mathbf{E}^s_{j,r}, \tag{1}$$

$$\mathbf{A}''_{t,i,j} = \max(0, \mathbf{A}'_{t,i,j}), \tag{2}$$

$$\mathbf{A}_{t,i,j} = \frac{e^{\mathbf{A}''_{t,i,j}}}{\sum_{n=1}^{N_s} e^{\mathbf{A}''_{t,i,n}}}. \tag{3}$$

In this work, three tensors $\mathbf{A}^p \in \mathbb{R}^{N_t \times N_p \times N_p}$, $\mathbf{A}^a \in \mathbb{R}^{N_t \times N_a \times N_a}$ and $\mathbf{A}^{ap} \in \mathbb{R}^{N_t \times N_p \times N_a}$ are generated to represent dynamic graphs: graph among primary nodes $\mathcal{G}^p$, graph among auxiliary nodes $\mathcal{G}^a$, graph between primary nodes and auxiliary nodes $\mathcal{G}^{ap}$ respectively.

3.2 Dynamic Graph Convolution

Spatial dependencies of the traffic speeds of different road segments could be used to improve the traffic speed prediction performance. Though the graph convolution could aggregate neighbor hidden states to handle the spatial dependencies between segments, majority of existing methods utilized it on a static graph. In this paper, to

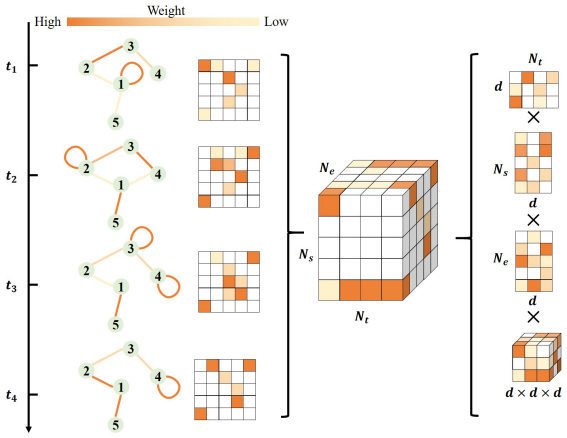


Figure 3: Construction of dynamic graph. Graph at single time slot could be represented as a matrix and graph at all time slots can be represented as a tensor with extra time dimension. And idea from tensor decomposition could be used to construct this tensor for simplicity of the model and exploit low-rank characteristic of spatial dependencies.

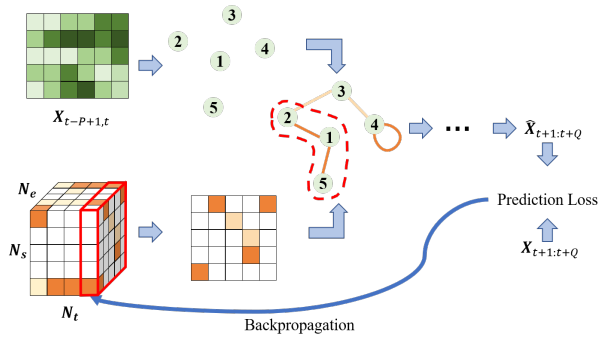


Figure 4: Dynamic graph convolution. The input is isolated data of multiple segments with no order. By fetching adjacency matrix with the time from dynamic graph tensor, the input can be organized according to the dynamic adjacency matrix and graph convolution operation is applied on it. According to the final prediction, adjacency tensor would be updated through backpropagation.

model latent and time varying spatial dependencies between nodes, we propose a dynamic version of graph convolution to conduct it on different graphs at different time.

Given input $\mathbf{X}_{t-P+1:t}$, $\mathbf{A}_{\phi(t)}$ reflects spatial relationship between nodes at time t. As shown in Figure 4, input hidden states of each node are isolated at first, and after fetching $\mathbf{A}_{\phi(t)}$ from adjacency tensor, the hidden states can be organized as graph signal according to their dynamic spatial relationships. Moreover, the method to aggregate information is inspired by DCRNN [19], which views the traffic flow as the diffusion procedure on graph. In math form, this operation can be realized by matrix multiplication of the adjacency matrix, hidden states of nodes and learnable parameters. At each step, the hidden states of focal node are updated by aggregating neighbors' hidden states through weighted links and the dynamic graph convolution can be defined as:

$$\mathbf{H}_l = \sum_{k=0}^{K} \left(\mathbf{A}_{\phi(t)}\right)^k \mathbf{H}_l^t \mathbf{W}^k, \tag{4}$$

where $\mathbf{H}_l^t$ represents output hidden states of temporal convolution layer in l-th block which serves as input of dynamic graph convolution in l-th block, $\mathbf{W}^k$ is parameters for depth k and K is max diffusion steps.

3.3 A General Framework for Auxiliary Feature

Auxiliary feature could not only affect primary feature at the same location, but also be helpful in predicting primary feature of highly related segments. For example, the increasing upstream flow will affect the downstream flow in the future and thus have an effect on the drop of the downstream traffic speed, which behaves as another diffusion pattern. So it is necessary to model spatial dependencies between primary nodes and auxiliary nodes. Here we propose a general framework powered by multi-faceted fusion module to integrate auxiliary feature with primary feature in graph perspective.

Meanwhile, we argue that there also exists spatio-temporal dependencies among nodes with auxiliary feature. In each layer, we assign a separate spatio-temporal block for each type of feature (primary feature and auxiliary feature). After the separate structure of l-th layer, we can get hidden states for primary feature $\mathbf{H}_l^p$ and hidden states for auxiliary feature $\mathbf{H}_l^a$. To model inter-dependencies between auxiliary feature and primary feature, we follow a "propagation and aggregation" paradigm and propose the multi-faceted fusion module:

$$\mathbf{H}_l^{p\prime} = \text{Aggregate}(\mathbf{H}_l^p, \text{Propagate}(\mathbf{H}_l^a)). \tag{5}$$

Specifically, instead of directly concatenating auxiliary hidden states with primary hidden states at same location, we generate a dynamic graph $\mathcal{G}^{ap}$ represented as $\mathbf{A}^{ap} \in \mathbb{R}^{N_t \times N_p \times N_a}$ to reflect spatial dependencies between them. The propagation module could propagate the most relative auxiliary hidden states to needed primary nodes and is calculated as:

$$\mathbf{H}_l^{ap} = \text{Propagate}(\mathbf{H}_l^a) = \mathbf{A}_{\phi(t)}^{ap} \mathbf{H}_l^a \mathbf{W}, \tag{6}$$

where $\mathbf{H}_l^{ap} \in \mathbb{R}^{N_p \times P \times d_p}$ represents effects of the auxiliary feature on primary nodes and d_p is the dimension of primary hidden states. To aggregate these unordered information (primary feature and effects of auxiliary feature), we choose SUM$(\cdot)$ as the aggregator function which is differentiable and maintains high representational capacity:

$$\mathbf{H}_l^{p\prime} = \text{Aggregate}(\mathbf{H}_l^p, \mathbf{H}_l^{ap}) = \mathbf{H}_l^p + \mathbf{H}_l^{ap}. \tag{7}$$

And the aggregator function could also be chosen as MEAN$(\cdot)$, MAX$(\cdot)$, or CONCAT$(\cdot)$ according to different tasks.

Our method passing auxiliary information in a graph perspective can not only consider spatial dependencies between primary nodes and auxiliary nodes, but also work in situation that auxiliary nodes are unaligned with primary nodes. Thanks to dynamic graph constructor, we can easily construct a dynamic graph for unaligned

N_a auxiliary nodes and N_p primary nodes which could support the procedure of message passing. This makes our method more generic since situation of unaligned data is extremely common in real world: if we want to predict the customer flow in a mall, considering traffic volume of road segments around the mall will be helpful but it cannot be achieved by traditional concatenation methods. And traditional concatenation methods could be viewed as a special case of this framework where the graph at each time is represented as an identity matrix.

Under situation where there exist multiple auxiliary features $\{\mathbf{H}_l^{a_1}, \mathbf{H}_l^{a_2}, \cdots, \mathbf{H}_l^{a_M}\}$ and numbers of auxiliary nodes are $N_{a_i}, i \in [1, M]$, Equation 5-7 in our framework could be extended to:

$$\mathbf{H}_l^{p\prime} = \mathbf{H}_l^p + \sum_{m=1}^{M} \mathbf{A}_{\phi(t)}^{ap_m} \mathbf{H}_l^{a_m} \mathbf{W}, \tag{8}$$

where $\mathbf{A}^{ap_m} \in \mathbb{R}^{N_t \times N_p \times N_{a_m}}$ is the generated dynamic graphs between m-th auxiliary feature and the primary feature.

3.4 Temporal Convolution Layer

Traffic speed of a segment is highly correlated with its historical status. This part processes data in time dimension to capture temporal dynamics of traffic. As shown in Figure 2, we adopt temporal convolution layer [23] with the consideration of training speed and simplicity. By stacking dilated convolution layers with increasing dilation factors, the receptive field of models grows exponentially. And compared to recurrent neural network, dilated convolution layers could be computed in parallel and thus lower time complexity a lot. Meanwhile, gating mechanism shows strengths in handling sequence data, so it's used in temporal convolution layer to improve model capacity. Specially, the temporal convolution layer is in the form:

$$\mathbf{H}_l^t = \tanh(\mathbf{W}_{f,l} \star \mathbf{F}_l) \odot \sigma(\mathbf{W}_{g,l} \star \mathbf{F}_l), \tag{9}$$

where $\odot$ denotes an element-wise multiplication operator, $\sigma(\cdot)$ is a sigmoid function, $\star$ is the dilated convolution operation and $\mathbf{W}_{(\cdot)}$ is learnable parameters of convolution filters.

3.5 Other Components

In the proposed model, $\mathbf{F}_l^{(\cdot)}$ is input of l-th block and output of $(l-1)$-th block. Before the first block, input data is transformed by a fully connected layer:

$$\mathbf{F}_1^p = \mathbf{W}_{in}^p \mathbf{X}^p + \mathbf{b}_{in}^p, \tag{10}$$

$$\mathbf{F}_1^a = \mathbf{W}_{in}^a \mathbf{X}^a + \mathbf{b}_{in}^a. \tag{11}$$

And residual links are added for each block:

$$\mathbf{F}_{l+1}^p = \mathbf{H}_l^{p\prime} + \mathbf{F}_l^p, \tag{12}$$

$$\mathbf{F}_{l+1}^a = \mathbf{H}_l^a + \mathbf{F}_l^a. \tag{13}$$

Moreover, skip connection is added after each temporal convolution layer of primary blocks to the output layer. Hidden states from layers of different depth are concatenated and passed into two fully-connected layers for the final prediction of primary traffic feature in the future Q steps:

$$\mathbf{H} = ||_{l=1}^{L} \text{reshape}(\mathbf{H}_l^{p,t}), \tag{14}$$

$$\hat{\mathbf{X}}_{t+1:t+Q}^p = \mathbf{W}_{fc2} \cdot \text{ReLU}(\mathbf{W}_{fc1} \cdot \text{ReLU}(\mathbf{H}) + \mathbf{b}_{fc1}) + \mathbf{b}_{fc2}, \tag{15}$$

where $||$ is the concatenation operation, reshape$(\cdot)$ is the function to reshape hidden states $\mathbf{H}_l^{p,t}$ for concatenation, $\mathbf{W}_{(\cdot)}$ and $\mathbf{b}_{(\cdot)}$ are learnable parameters. And MAE is used as the objective function to train the model in an end-to-end manner:

$$\mathcal{L} = \frac{1}{Q \times N_p} \sum_{i=t+1}^{t+Q} \sum_{j=1}^{N_p} |\mathbf{X}_{i,j}^p - \hat{\mathbf{X}}_{i,j}^p|. \tag{16}$$

4 EXPERIMENTS

In this section, we evaluate our proposed model by empirically examining on three real-world datasets. The following research questions (RQs) are used to guide our experiments:

- **RQ1.** How does the proposed model perform compared to existing traffic speed forecasting methods?
- **RQ2.** How does each component contribute to the performance of the proposed model?
- **RQ3.** How does the constructed dynamic graph help to improve the performance of the proposed model?
- **RQ4.** How does the proposed model perform with unaligned auxiliary features?

4.1 Datasets

Our experiments are conducted on three real-world datasets: PeMSD4, PeMSD8 and England respectively[1]. These datasets both contains traffic speed and traffic volume collected by sensors. The traffic speed is viewed as the primary feature to predict with traffic volume as auxiliary feature. Statistics of these datasets are shown in Table 2. And other details of the datasets are introduced below:

- **PeMSD4.** It is collected by Caltrans Performance Measurement System (PeMS)[2] and released in ASTGCN [14] consisting of average speed, traffic volume in San Francisco Bay Area. Time span is from January to February in 2018.
- **PeMSD8.** Similar as PeMSD4, it consists of average speed, traffic volume collected by PeMS in San Bernardino from July to August in 2016.
- **England.** This dataset is derived from Highways England Traffic Data from Opening up Government of UK[3] consisting of average speed, traffic volume around the country. In this paper, data from January to June in 2014 is selected.

4.2 Baselines

To validate performance of DMSTGCN, models including statistic methods, predefined graph based methods, attention based methods and adaptive graph based methods are selected as baselines to compare. The description of these baselines are as following:

- HA: Historical Average method. Here, considering the periodicity of traffic, we use the average value of traffic speed an the same time of day in training dataset as prediction.
- VAR [36]: A statistical model used to capture the relationship between multiple quantities.
- LR: A regression model which exploits linear correlation between input and output.

[1] Code and dataset are available at https://github.com/liangzhehan/DMSTGCN
[2] http://pems.dot.ca.gov/
[3] http://tris.highwaysengland.co.uk/detail/trafficflowdata

Table 1: Evaluations of DMSTGCN and baselines on three real-world datasets.

Dataset	Method	Horizon 1			Horizon 3			Horizon 6			Horizon 12		
		MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
PeMSD4	HA	3.5673	0.0803	6.7782	3.5679	0.0803	6.7787	3.5689	0.0804	6.7797	3.5707	0.0804	6.7815
	VAR	1.0376	0.0193	1.8578	1.6624	0.0327	3.0882	2.1250	0.0433	4.0156	2.5687	0.0536	4.8347
	LR	0.9933	0.0182	1.8537	1.5655	0.0309	3.1384	2.0758	0.0439	4.2918	2.7951	0.0627	5.5978
	XGBoost	0.9268	0.0169	1.7497	1.4471	0.0288	2.9732	1.9159	0.0413	4.0955	2.5327	0.0585	5.3239
	DCRNN	0.9267	0.0169	1.7216	1.4295	0.0281	2.8805	1.8531	0.0391	3.8731	2.3666	0.0527	4.8991
	ASTGCN	1.1132	0.0225	2.1089	1.5428	0.0323	3.1428	1.9374	0.0425	4.1062	2.4424	0.0552	5.1392
	GMAN	1.0739	0.0206	2.0416	1.3760	0.0279	2.9803	1.6213	0.0348	3.7906	**1.8650**	0.0416	4.4791
	GWNet	0.8922	0.0165	1.7050	1.3266	0.0266	2.8159	1.6456	0.0356	3.7640	1.9550	0.0446	4.5560
	MTGNN	0.9298	0.0173	1.7548	1.3485	0.0268	2.8517	1.6454	0.0348	3.7523	1.9306	0.0426	4.4972
	DMSTGCN	**0.8847**	**0.0161**	**1.6929**	**1.3002**	**0.0255**	**2.7769**	**1.5875**	**0.0333**	**3.6339**	*1.8787*	**0.0415**	**4.3814**
PeMSD8	HA	2.8130	0.0631	5.6782	2.8119	0.0631	5.6763	2.8099	0.0630	5.6730	2.8047	0.0627	5.6647
	VAR	0.9308	0.0172	1.7697	1.1327	0.0224	2.0712	1.7166	0.0356	3.3053	2.1661	0.0467	4.2500
	LR	0.7978	0.0144	1.5147	1.2711	0.0244	2.6335	1.6614	0.0335	3.5606	2.1681	0.0459	4.5794
	XGBoost	0.7622	0.0140	1.4641	1.1997	0.0241	2.5698	1.5604	0.0338	3.5030	2.0008	0.0458	4.4840
	DCRNN	0.7692	0.0141	1.4386	1.1957	0.0236	2.4679	1.5347	0.0325	3.3034	1.9051	0.0427	4.1451
	ASTGCN	1.0737	0.0234	2.3076	1.3794	0.0297	2.9934	1.6446	0.0358	3.6194	1.9945	0.0436	4.2880
	GMAN	0.9177	0.0179	1.8698	1.1374	0.0234	2.6752	1.3237	0.0292	3.3950	**1.5120**	**0.0355**	4.0524
	GWNet	0.7427	0.0139	1.4435	1.1184	0.0233	2.5533	1.3849	0.0319	3.5327	1.6043	0.0391	4.2424
	MTGNN	0.7749	0.0146	1.5000	1.1402	0.0234	2.5738	1.3977	0.0314	3.4913	1.6323	0.0394	4.2490
	DMSTGCN	**0.7222**	**0.0132**	**1.4217**	**1.0667**	**0.0209**	**2.4405**	**1.3082**	**0.0280**	**3.2861**	*1.5522*	*0.0358*	**4.0522**
England	HA	7.0474	0.0993	12.2131	7.0473	0.0993	12.2133	7.0441	0.0993	12.2106	7.0341	0.0992	12.2005
	VAR	2.7768	0.0342	5.3560	3.2135	0.0419	5.7254	4.1019	0.0563	7.5764	4.8372	0.0693	8.9487
	LR	2.3136	0.0279	4.4323	3.7732	0.0505	7.4331	5.2816	0.0737	9.7045	6.5664	0.0943	11.5991
	XGBoost	2.0770	0.0251	4.0843	3.1924	0.0440	6.7177	4.2988	0.0631	8.6673	5.5237	0.0829	10.4756
	DCRNN	2.0231	0.0244	3.9075	2.8363	0.0381	6.1062	3.4913	0.0487	7.4473	4.2813	0.0610	8.7217
	ASTGCN	2.3136	0.0289	4.3160	3.2141	0.0439	6.5404	3.9262	0.0556	7.9177	4.6010	0.0664	9.0793
	GMAN	2.1268	0.0267	4.4948	2.6117	0.0362	6.2874	2.9558	0.0431	7.3339	3.3066	0.0493	8.1404
	GWNet	1.9102	0.0234	3.9376	2.5216	**0.0348**	5.9610	2.9610	0.0436	7.1284	3.4461	0.0513	8.0687
	MTGNN	1.9411	0.0240	3.9590	2.5352	0.0354	5.9789	2.9216	0.0430	7.0889	3.3359	0.0507	7.9936
	DMSTGCN	**1.8945**	**0.0231**	**3.8973**	**2.5104**	*0.0350*	**5.9526**	**2.8899**	**0.0428**	**7.0407**	**3.2554**	**0.0493**	**7.8060**

Table 2: Statistics of datasets.

Datasets	# Samples	# Nodes	Sample Rate	Time Span
PeMSD4	16969	307	5 mins	2 months
PeMSD8	17833	170	5 mins	2 months
England	17353	314	15 mins	6 months

- XGBoost [5]: A method based on gradient boosting tree.
- DCRNN [19]: A spatio-temporal network integrating diffusion graph convolution and recurrent neural network.
- ASTGCN [14]: A model applying spatial and temporal attention mechanism before spatial and temporal convolutions. For fairness, we only use recent component of it.
- GMAN [35]: A graph multi attention model considering spatial and temporal correlation based on input features, spatial embedding and temporal embedding.
- Graph Wavenet [27]: A spatio-temporal network introducing a method to generate adaptive graph and integrating diffusion graph convolution with 1-D dilated convolution.
- MTGNN [26]: A spatio-temporal network using external features to generate unidirectional adaptive graph.

4.3 Setup of Experiments

In our model, the number of blocks in each part is set to 8, dilated ratio of each layer is $[1, 2, 1, 2, 1, 2, 1, 2]$ and the max depth of graph convolution layers is set to 2. Channel size of dilated convolution and graph convolution is 32 and hidden dimension in dynamic graph constructor is set to 16. Batch size is set to 64 and learning rate is set to 0.001. The main hypeparameters are tuned on validation set. The model is optimized by Adam optimizer and an early stop strategy is used with patience of 20. All datasets are split by ratio of 6:2:2 in chronological order. For PeMSD4 and PeMSD8, the missing values are filled by the linear interpolation. Predefined graph is initialized according distance between sensors. For the fairness of comparison experiments, except for HA and VAR, input features of each baseline include historical primary feature, historical auxiliary feature, time in a day and day in a week. For LR and XGBoost, we train a model for each prediction step. Every experiment is repeated 5 times and the average performance is reported. Experiments are executed on a machine with four TitanXp GPUs. Three metrics are adopted to evaluate performance of each model: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE) respectively.

4.4 Comparison with Existing Models (RQ1)

The results of the comparison with baselines are shown in Table 1. On all three datasets which are collected at multiple locations and with various sampling rates, our proposed model always achieves the start-of-the-art performance whether long-term or short-term, which demonstrates the effectiveness of our proposed

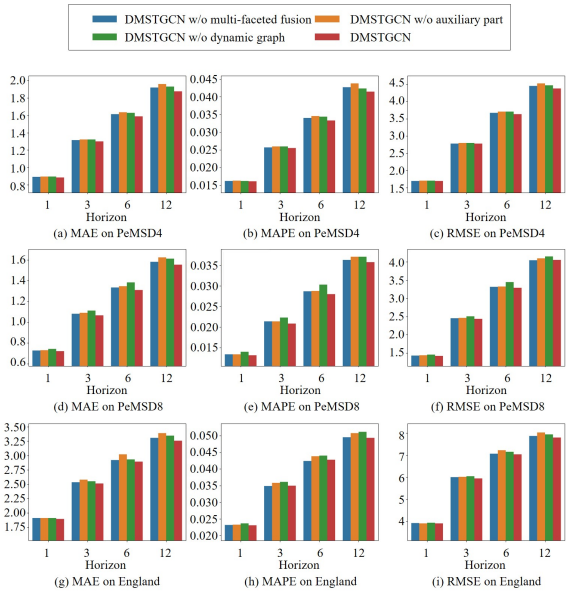


Figure 5: Ablation study.

model. Moreover, deep learning based models could achieve better performance than traditional statistic methods which demonstrates superior capacity of deep learning based models. Methods like DCRNN and ASTGCN highly rely on predefined graph which may not capture crucial dependencies between nodes leading to a worse performance. Although GMAN achieves competitive performance in Horizon 12, the results of short-term prediction has a distinct margin with our method. But compared with other work, GMAN has a clear improvement in Horizon 12, which may owe to the transform attention layer. Except for our model, Graph Wavenet and MTGNN perform well generally which might benefit from that they adopt adaptive graph to model relationships between nodes, indicating that adaptive graph based methods could effectively exploit valuable but latent spatial dependencies from historical traffic data. However, dynamic characteristic of traffic spatial dependencies and spatial dependencies between multi-faceted features are omitted in these work. The design of dynamic graph constructor and dynamic graph convolution help our model capture the subtler correlations between nodes. Moreover, as another way containing information of time in a day, our method has a better performance compared with taking it as input directly. And the interactions between auxiliary features and primary feature by multi-faceted fusion module also contribute a lot in this match. Taking these two factors into consideration make our model perform better than these methods consistently.

4.5 Evaluating Effectiveness of Key Designs in Proposed Model (RQ2)

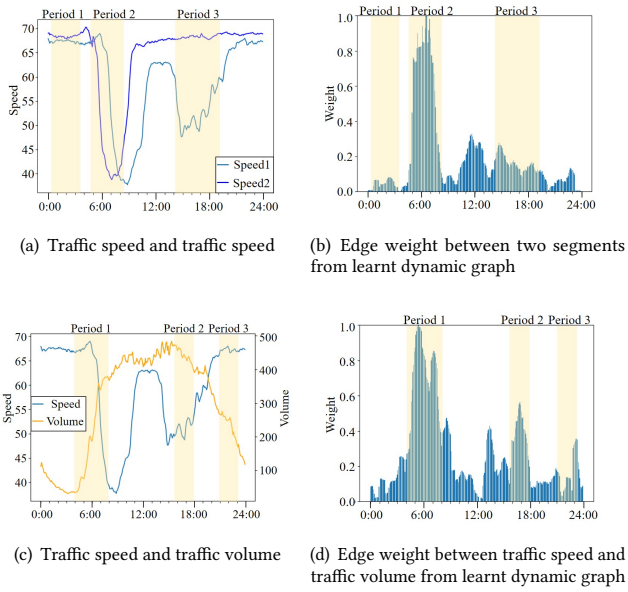
The two key designs of our proposed model are the dynamic graph constructor for time varying spatial dependencies of nodes and the multi-faceted fusion module to integrate auxiliary features with the primary feature in graph perspective. To validate effectiveness of our proposed components, we design three variants: DMSTGCN w/o multi-faceted fusion, DMSTGCN w/o auxiliary part and DMSTGCN w/o dynamic graph:

- DMSTGCN w/o multi-faceted fusion: In this variant, instead of using graph-based fusion module, we directly add auxiliary hidden states and primary hidden states together to demonstrate the effectiveness of our proposed module.
- DMSTGCN w/o auxiliary part: In this variant, we remove the auxiliary block wholly to demonstrate the importance of auxiliary information.
- DMSTGCN w/o dynamic graph: In this variant, we set the number of time slots as 1 to generate only one static adaptive graph to demonstrate effectiveness of dynamic graph constructor.

Results of ablation study are shown in Figure 5 in which it can be seen that key designs all contribute to improvement of proposed model. Compared to DMSTGCN w/o multi-faceted fusion, DMSTGCN achieves better performance indicating that the primary feature is affected by multiple neighbor auxiliary features and thus aggregating on dynamic graph could better support the prediction of the primary feature. Compared to DMSTGCN w/o auxiliary part, DMSTGCN w/o multi-faceted fusion performs better demonstrate importance of auxiliary feature. The outperformance of DMSTGCN over DMSTGCN w/o dynamic graph indicates that importance of modelling dynamic spatial dependencies between nodes.

4.6 Study of Dynamic Graph (RQ3)

Our learned dynamic graph could also be used in discovering dependencies between nodes which will benefit applications such as traffic control. Whether it could find reasonable traffic patterns is also a criteria to judge if the adaptive graph is well-trained. A case study in conducted on PeMSD4. For correlations between the same type of nodes, historical average traffic speed from two segments is illustrated in Figure 6(a) and edge weight of all time slots learnt by our model is illustrated in Figure 6(b) after smoothness and normalization. In period 1, the speed of two segments fluctuates randomly and edge weight of them is low in learnt dynamic graph. In period 2, the tendency of two segments is similar and one is ahead of another which could be helpful in prediction of latter one. In this period, the edge weight in learnt dynamic graph is high. And in period 3, the speed of one segment drops due to evening peak while the speed of another segment doesn't change a lot and the edge weight in learnt dynamic graph is low. For dependencies between auxiliary features and the primary feature, historical average traffic speed of one sensor and historical average traffic volume of one sensor are illustrated in Figure 6(c). In period 1 and period 2, traffic volume always changes ahead of traffic speed which shows high correlations between them. This property of traffic volume could be leveraged to predict upcoming changes on traffic speed which is hard to be exploited from historical traffic speed. When in period 3, traffic volume decreases rapidly, traffic speed returns to a high level and fluctuates randomly showing low correlations between them. Figure 6(d) shows edge weight in learnt dynamic graph, high weight in period 1, period 2 and low weight in period 3 verify our observations in real world. In a word, our proposed dynamic graph



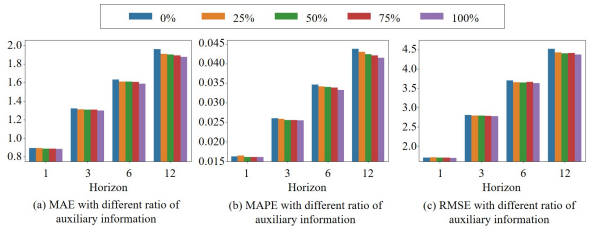
(a) Traffic speed and traffic speed

(b) Edge weight between two segments from learnt dynamic graph

(c) Traffic speed and traffic volume

(d) Edge weight between traffic speed and traffic volume from learnt dynamic graph

Figure 6: Illustration of learnt dynamic graph.



(a) MAE with different ratio of auxiliary information

(b) MAPE with different ratio of auxiliary information

(c) RMSE with different ratio of auxiliary information

Figure 7: The more auxiliary information is added, the higher performance our model could achieve.

constructor could effectively exploit dynamic spatial dependencies between segments and multi-faceted features.

4.7 Study of Utilizing Unaligned Auxiliary Information (RQ4)

Compared to directly concatenating auxiliary features and the primary feature of same segment, our method could integrate auxiliary features into the primary feature in case that they are unaligned. This property is extremely useful in real world such as predicting customer flow in a mall using traffic volume of road segments around it. To explore the capability to leverage unaligned auxiliary features, we design a set of experiments on PeMSD4 dataset by changing the ratio of auxiliary information. Specifically, the input contains primary features on all segments and auxiliary features on segments of difference ratio including 0%, 25%, 50%, 75% and 100%. As it is shown in Figure 7, the performance goes better as more auxiliary features are added. This indicates that our method works in the scenario that auxiliary information is unaligned with primary information and could always take advantage of relative auxiliary information to benefit prediction of the primary feature.

5 RELATED WORK

This section reviews the previous literature related to traffic speed forecasting and spatio-temporal graph neural network.

5.1 Traffic Speed Forecasting

Traffic forecasting has attracted much attention of enormous researchers in the past decades [21]. To exploit traffic patterns from data and provide prediction, simple statistic methods including ARIMA[3], VAR [36] for time series forecasting have been developed. Though they can leverage accumulated traffic data, they were built upon stationary assumptions. In recent years, researchers have been shifting to deep learning methods that are powerful to capture non-linear patterns. Thanks to gated mechanism, recurrent neural network, such as LSTM [15], could exploit patterns from sequential data and were leveraged in traffic forecasting [22]. However, spatial dependencies between segments were omitted which encourage researchers to propose methods taking spatial dependencies into consideration. Methods were proposed to split the city into grids and utilized the idea from computer vision to handle spatial dependencies in a local area [28–30, 33]. Though these methods took a big step to model spatial dependencies and temporal dependencies simultaneously, the spatial dependencies were in the Euclidean space while segments are naturally organized as a graph in real world. The inaccurate assumptions limited the performance and generality of them. More recently, several methods combining graph neural network and sequential learning network were proposed to handle spatial and temporal dependencies in traffic scenario and achieved satisfying performance [1, 14, 19, 26, 27, 32]. On the other hand, some work [8, 14, 20] realized the importance of auxiliary features, but they simply linked auxiliary features to the primary feature with the same location and time slot lacking generality and omitting spatial dependencies between multi-faceted features.

5.2 Spatio-temporal Graph Neural Networks

Due to advantage of both graph neural network and sequential learning methods, spatio-temporal graph neural network methods could handle the spatial dependencies in non-Euclidean space and temporal dependencies simultaneously. Along this line, it is a challenging problem to construct a suitable graph between segments. STGCN [32], DCRNN [19], ASTGCN [14] combined spatial and temporal network including Chebnet [17], GRU [7], diffusion convolution and 1-D convolution. GMAN [35] utilized node2vec [13], an embedding method for nodes in graph, and extended self-attention mechanism [25] to spatial dimension. CurbGAN [34] leveraged Pearson Correlation Coefficient to measure edge weights between nodes. Some works [4, 12] took a further step to consider multiple types of correlations between nodes include distance, Pearson Correlation Coefficient of historical data, functional similarity and interactions between each pair of nodes. These methods could handle correlations with arbitrary number of neighbors, but their graphs were designed, which is intuitive but incomplete. In recent years, some work to generate adaptive graph from data have been proposed to solve this problem. Graph Wavenet [27] added an adaptive term by generating one learnable correlation matrix to extend diffusion convolution from DCRNN for better capturing latent graph behind data. AGCRN [1] kept adaptive term only and still

achieved superior performance. MTGNN [26] took available external node features to generate one adaptive graph by graph learning layer. Though these methods could exploit correlations from data, they all focused on static graphs omitting dynamic characteristics of spatial dependencies between segments.

6 CONCLUSION

In this paper, we proposed a novel framework to solve the traffic speed forecasting problem with multi-faceted data. We generated an adaptive dynamic graph organized as a three-order tensor to model varying relationships between segments and extended existing graph convolution to a dynamic one. Moreover, benefiting from construction of dynamic graph, we can also model spatial relationships between auxiliary data and traffic speed. Our work could be widely applied in real world whether there is a predefined graph or not and whether the auxiliary information is aligned or unaligned. Experiments were conducted on three real-world datasets to demonstrate superiority of our proposed model. We also gave a direct viewing result of exploited dynamic graph and analysis for unaligned auxiliary data. The well-trained embeddings and learnt dynamic graph could also be potentially applied to other tasks.

ACKNOWLEDGMENTS

This work was partly supported by the National Natural Science Foundation of China (71901011, 51822802, 51778033, 51991395, U1811463) and the Science and Technology Major Project of Beijing (Z181100009018010).

REFERENCES

[1] LEI BAI, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. *Advances in Neural Information Processing Systems* 33 (2020).

[2] Robert Kerwin C Billones, Argel A Bandala, Edwin Sybingco, Laurence A Gan Lim, and Elmer P Dadios. 2016. Intelligent system architecture for a vision-based contactless apprehension of traffic violations. In *2016 IEEE Region 10 Conference (TENCON)*. IEEE, 1871–1874.

[3] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

[4] Di Chai, Leye Wang, and Qiang Yang. 2018. Bike flow prediction with multi-graph convolutional networks. In *Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems*. 397–400.

[5] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[6] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. 2019. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems* 21, 3 (2019), 1086–1095.

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[8] Shengdong Du, Tianrui Li, Yan Yang, and Shi-Jinn Horng. 2019. Deep air quality forecasting using hybrid deep learning framework. *IEEE Transactions on Knowledge and Data Engineering* (2019).

[9] Aleksandr Fedorov, Kseniia Nikolskaia, Sergey Ivanov, Vladimir Shepelev, and Alexey Minbaleev. 2019. Traffic flow estimation with data from a video surveillance camera. *Journal of Big Data* 6, 1 (2019), 1–15.

[10] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*. 1459–1468.

[11] Qiang Gao, Fan Zhou, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. 2019. Predicting human mobility via variational attention. In *The World Wide Web Conference*. 2750–2756.

[12] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 3656–3663.

[13] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[14] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 922–929.

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[16] Jemin Hwangbo, Inkyu Sa, Roland Siegwart, and Marco Hutter. 2017. Control of a quadrotor with reinforcement learning. *IEEE Robotics and Automation Letters* 2, 4 (2017), 2096–2103.

[17] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[18] William Koch, Renato Mancuso, Richard West, and Azer Bestavros. 2019. Reinforcement learning for UAV attitude control. *ACM Transactions on Cyber-Physical Systems* 3, 2 (2019), 1–21.

[19] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.

[20] Binbing Liao, Jingqing Zhang, Chao Wu, Douglas McIlwraith, Tong Chen, Shengwen Yang, Yike Guo, and Fei Wu. 2018. Deep sequence learning with auxiliary information for traffic prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 537–546.

[21] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. 2014. Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems* 16, 2 (2014), 865–873.

[22] Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies* 54 (2015), 187–197.

[23] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[24] Ledyard R Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31, 3 (1966), 279–311.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[26] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 753–763.

[27] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph WaveNet for deep spatial-temporal graph modeling. In *International Joint Conference on Artificial Intelligence 2019*. Association for the Advancement of Artificial Intelligence (AAAI), 1907–1913.

[28] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. 2019. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5668–5675.

[29] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[30] Junchen Ye, Leilei Sun, Bowen Du, Yanjie Fu, Xinran Tong, and Hui Xiong. 2019. Co-prediction of multiple transportation demands based on deep spatio-temporal neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 305–313.

[31] Junchen Ye, Leilei Sun, Bowen Du, Yanjie Fu, and Hui Xiong. 2020. Coupled Layer-wise Graph Convolution for Transportation Demand Prediction. *arXiv preprint arXiv:2012.08080* (2020).

[32] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3634–3640.

[33] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[34] Yingxue Zhang, Yanhua Li, Xun Zhou, Xiangnan Kong, and Jun Luo. 2020. Curb-GAN: Conditional Urban Traffic Estimation through Spatio-Temporal Generative Adversarial Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 842–852.

[35] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1234–1241.

[36] Eric Zivot and Jiahui Wang. 2006. Vector autoregressive models for multivariate time series. *Modeling Financial Time Series with S-Plus®* (2006), 385–429.