Modeling Spatial-Temporal Dynamics for Traffic Prediction

Huaxiu Yao*
Pennsylvania State University
huaxiuyao@ist.psu.edu

Xianfeng Tang*
Pennsylvania State University
xianfeng@ist.psu.edu

Hua Wei
Pennsylvania State University
hzw77@ist.psu.edu

Guanjie Zheng
Pennsylvania State University
gjz5038@ist.psu.edu

Yanwei Yu
Pennsylvania State University
yuy174@ist.psu.edu

Zhenhui Li
Pennsylvania State University
jessieli@ist.psu.edu

ABSTRACT

Spatial-temporal prediction has many applications such as climate forecasting and urban planning. In particular, traffic prediction has drawn increasing attention in data mining research field for the growing traffic related datasets and for its impacts in real-world applications. For example, an accurate taxi demand prediction can assist taxi companies to pre-allocate taxis to meet with commuting demands. The key challenge of traffic prediction lies in how to model the complex spatial and temporal dependencies. In this paper, we make two important observations which have not been considered by previous studies: (1) the spatial dependency between locations are dynamic; and (2) the temporal dependency follows strong periodicity but is not strictly periodic for its dynamic temporal shifting. Based on these two observations, we propose a novel Spatial-Temporal Dynamic Network (STDN) framework. In this framework, we propose a flow gating mechanism to learn the dynamic similarity between locations via traffic flow. A periodically shifted attention mechanism is designed to handle long-term periodic dependency and periodic temporal shifting. Furthermore, we extend our framework from region-based traffic prediction to traffic prediction for road intersections by using graph convolutional structure. We conduct extensive experiments on several large-scale real traffic datasets and demonstrate the effectiveness of our approach over state-of-the-art methods.

KEYWORDS

Spatial-Temporal Dynamics, Deep Learning, Traffic Prediction

ACM Reference Format:
Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2018. Modeling Spatial-Temporal Dynamics for Traffic Prediction. In *Proceedings of* . ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Spatial-temporal prediction is ubiquitous in real world, from climate forecasting, to next location prediction, to traffic prediction. In particular, traffic prediction has drawn increasing attention with the growing amount of traffic-related datasets such as taxi pick-ups and drop-offs, and traffic conditions obtained from surveillance cameras. In the meantime, an accurate traffic prediction model is essential to many real-world applications. For example, taxi demand prediction can help taxi companies pre-allocate taxis; traffic volume prediction can help transportation department better manage and control the traffic to ease traffic congestions.

In a typical traffic prediction setting, given historical traffic data (e.g., traffic volume of a region or a road intersection for each hour during the previous month), we aim to predict the traffic for the next time slot. A number of studies have investigated traffic prediction for decades. In time series community, autoregressive integrated moving average (ARIMA) and Kalman filtering have been widely applied to traffic prediction problems [16, 18, 21, 24]. While these methods study traffic time series for each individual location, recent studies further utilize spatial information (e.g., adding regularizations on model similarity for nearby locations) [9, 12, 37] and external context information (e.g., adding features of venue information, weather condition, and local events) [22, 27, 31]. However, these approaches are still based on traditional time series models or machine learning models and do not well capture the complex non-linear spatial-temporal dependency.

Recently, deep learning has shown its great learning and representation power in many challenging fields, such as computer vision [15], natural language processing [19], speech recognition [10], and recommendation [29]. The success has inspired several attempts to apply deep learning techniques on traffic prediction problem. For example, recent studies consider city-wide traffic as a heatmap image, where each pixel value represents the traffic in the corresponding region [20, 35, 36]. In these studies, convolutional neural network (CNN) is used to model the non-linear spatial dependency. To model non-linear temporal dependency, researchers propose to use recurrent neural network (RNN)-based framework [7, 34]. Yao et al. [32] further propose a method to jointly model both spatial and temporal dependencies by integrating CNN and long short-term memory (LSTM).

The existing methods using deep learning techniques for traffic prediction have two major limitations. First, the spatial dependency between locations relies only on the similarity of historical traffic [32, 35] and the model learns a static spatial dependency. However, the dependencies between locations could change over time. For example, in the morning, the dependency between a residential area and a business center is strong; whereas in late evening, the relation between a working place and a restaurant area might be

*equal contribution.

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

strong. Traffic flow data is a helpful information source to describe such dynamic relations.

Another limitation is that many existing studies ignore the shifting of long-term periodic dependency. Traffic data show a strong daily and weekly periodicity and the dependency on the previous periodic time frame can be useful for prediction. However, one challenge is that the traffic data is not strictly periodic. For example, the peak hours on weekdays are usually in the late afternoon, but could vary from 4:30pm to 6:00pm on different days. Though previous studies [35, 36] considers periodic traffic data, they fail to consider the sequential dependency and the temporal shifting in the periodicity.

To address the aforementioned challenges, we propose a novel deep learning architecture, spatial-temporal dynamic network (STDN) for traffic prediction. STDN is based on a spatial-temporal neural network, which handles spatial and temporal information via local CNN and LSTM, respectively. A flow-gated local CNN is proposed to handle spatial dependency by modeling the dynamic similarity among locations using traffic flow information. A periodically shifted attention mechanism is proposed to learn the long-term periodic dependency. The proposed mechanism captures both long-term periodic information and temporal shifting in traffic sequence via attention mechanism. Furthermore, our method uses LSTM to handle the sequential dependency in a hierarchical way. Finally, our approach can easily be extended to graph convolutional neural network and can predict traffic volumes at road intersections.

We evaluate the proposed method on large-scale real-world public datasets including taxi data of New York City (NYC), bike-sharing data of NYC, and road camera data of Jinan, China. The comprehensive comparisons with the state-of-the-art methods demonstrate the effectiveness of our proposed method.

To summarize, our contributions are as follows:

- We proposed a flow gating mechanism to explicitly model dynamic spatial similarity. The gate controls information propagation among nearby locations.
- We proposed a periodically shifted attention mechanism by taking long-term periodic information and temporal shifting simultaneously.
- We extended our model to road intersection traffic prediction, which uses graph convolutional structure to capture spatial dependency.
- We conducted experiments on several real-world traffic datasets. The results show that our model is consistently better than other state-of-the-art methods.

The rest of the paper is organized as follows. Related work is discussed in Section 2. We describe some notations and our problem in Section 3. The proposed method is presented in Section 4. We describe discuss the experimental results in detail in Section 5. Finally, we conclude the paper in Section 6.

2 RELATED WORK

Data-driven traffic prediction problems have received wide attention for decades. Essentially, the aim of traffic prediction is to predict a traffic-related value for a location at a timestamp based on historical data. In this section, we discuss the related work on traffic prediction problems.

In time series community, autoregressive integrated moving average (ARIMA), Kalman filtering and their variants have been widely used in traffic prediction problem [16, 18, 21, 24]. Recent studies further explore the utilities of external context data, such as venue types, weather conditions, and event information [22, 23, 31]. In addition, spatial information has also been explicitly modeled in recent studies. For example, Deng et al. [9] used matrix factorization on road networks to learn the latent space between road connected regions for predicting traffic volume. Several studies [12, 27, 37] were proposed to smooth the prediction differences for nearby locations and time intervals via several prior regularizations. *However, all of these methods fail to model the complex nonlinear relations of the space and time.*

Deep learning models provide a new promising way to capture non-linear spatiotemporal relations, which have achieved great success in the fields of computer vision [15, 30], natural language processing [19], speech recognition [10] and recommendation [29]. In traffic prediction, a series of studies have been proposed based on deep learning techniques. For example, Wei et al. [30] transferred spatial structure as vectors and used fully connected network to capture spatial information between neighbors for traffic demand prediction. Wang et al. [28] designed a neural network framework using context data from multiple sources and predict the gap between taxi supply and demand. *These methods used extensive features, but do not model the spatial and temporal interactions explicitly.*

A line of studies applied convolutional structure to capture spatial correlation. For example, Zhang et al. [36] and Zhang et al. [35] treated the whole city as image and constructed three same convolutional structures to capture trend, period and closeness information. Ma et al. [20] utilized CNN on the whole city for traffic speed prediction. Another line of studies used recurrent-neural-network-based model for modeling sequential dependency. Yu et al. [34] proposed to apply Long-short-term memory (LSTM) network and autoencoder to capture the sequential dependency for predicting the traffic under extreme conditions, particularly for peak-hour and post-accident scenarios. Cui et al. [7] proposed a deep stacked bidirectional and unidirectional LSTM network for traffic speed prediction. *However, while these studies explicitly model temporal sequential dependency or spatial dependency, none of them consider both aspects simultaneously.*

Recently, several studies use convolutional LSTM [25] to handle spatial and temporal dependency for taxi demand prediction [13, 38]. Yao et al. [32] further proposed a multi-view spatial-temporal network for demand prediction, which learns the spatial-temporal dependency simultaneously by integrating LSTM, local-CNN and semantic network embedding. Based on road network, Li et al. [17] extended convolutioanal GRU to graph convolutional GRU for traffic speed prediction. Yu et al. [33] proposed a gated graph convolutional network for efficiency. Cheng et al. [5] combine CNN, RNN and road structure to predict traffic congestion. *In these studies, the similarity between regions is based on static distance or road structure. They also overlook the long-term periodic influence and temporal shifting in time series prediction.*

In summary, our proposed model explicitly handle dynamic spatial similarity and temporal periodic similarity jointly via flow gating mechanism and periodically shifted attention mechanism, respectively.

3 PRELIMINARIES

In this section, we define some notations and the problem in this paper. We split the whole city to an $I \times J$ grid map which consists of I rows and J columns. We define the region set in the grid map as $L = \{l_1, l_2, ..., l_i, ..., l_{I\times J}\}$. Also, we split time as T non-overlapping time intervals and define the time interval set as $\mathcal{T} = \{ts_1, ts_2, ..., ts_t, ...ts_T\}$. For simplicity, we use t and i to represent time interval ts_t region l_i for the rest of the paper, respectively.

Additionally, we use (a, b) to define the spatiotemporal coordinate, where a is time and b is region. We denote the trip of an object as a start point $s = (a_s, b_s)$ and end point $e = (a_e, b_e)$. The trip set is defined as $\mathbb{M}$, which contains all trips' start-end (i.e., (s, e)) pairs.

Definition 1 (Start/End Traffic Volume) Given the trip set $\mathbb{M}$, the start traffic volume $y^i_{s,t}$ and end traffic volume $y^i_{e,t}$ for each region i at the time interval t are defined as:

$$y^i_{s,t} = |\{(s, e) \in \mathbb{M} : a_s \in ts_t \wedge b_s \in l_i\}|, \quad (1)$$

$$y^i_{e,t} = |\{(s, e) \in \mathbb{M} : a_e \in ts_t \wedge b_e \in l_i\}|. \quad (2)$$

Where $|\cdot|$ denotes the cardinality of the set. To some extent, the start and end traffic volume reflects the demand for the transportation and the popularity of region, respectively.

Definition 2 (Traffic Flow) The traffic flow is used to describe the dynamic interactions between regions. The flow $f^{i,j}_t$ from region i to j are defined as

$$f^{i,j}_t = |\{(s, e) \in \mathbb{M} : a_e \in ts_t \wedge b_s \in l_i \wedge b_e \in l_j\}|. \quad (3)$$

If i equals to j, it means the volume of self-loop flow. We consider the flow that happens at a certain time interval (i.e., $a_s \in ts_t$) and across time interval (i.e., $a_s < ts_t$).

Problem 1 (Traffic Volume Prediction) Given the data until time interval t, the traffic volume prediction problem aims to predict the start and end traffic volume at time interval $t + 1$. Similar to previous work [32], we also use context features, such as temporal statistical features (e.g., volume values of previous several time intervals) and spatial features (e.g., volume values of nearby regions). For each region i, the context features is defined as a vector $\mathbf{e}^i_t \in \mathbb{R}^r$. As we mentioned before, we predict start and end traffic volume simultaneously. Thus, the problem is formulated as

$$y^i_{s,t+1}, y^i_{e,t+1} = \mathcal{A}(\mathcal{Y}^{I\times J}_{s,t-h,...,t}, \mathcal{Y}^{I\times J}_{e,t-h,...,t}, \mathcal{E}_{t-h,...,t}). \quad (4)$$

Where $\mathcal{Y}^{I\times J}_{s,t-h,...,t}$ and $\mathcal{Y}^{I\times J}_{e,t-h,...,t}$ are historical start and end volume, respectively. $\mathcal{E}_{t-h,...,t}$ are context features for all locations. $\mathcal{A}(\cdot)$ is a prediction function.

4 METHODOLOGY

In this section, we describe the details for our proposed **S**patial-**T**emporal **D**ynamic **N**etwork (**STDN**). Figure 1 shows the architecture of our proposed method.

4.1 Local Spatial-Temporal Network

In order to capture spatial and temporal sequential dependency, combining local CNN and LSTM has shown the state-of-the-art performance in taxi demand prediction [32]. Here, we also use local CNN and LSTM to deal with spatial and short-term temporal dependency. In order to mutually reinforce the prediction of two types of traffic volumes (i.e., start and end volumes), we integrate and predict them together. This part of our proposed model is called **L**ocal **S**patial-**T**emporal **N**etwork (**LSTN**).

4.1.1 Local spatial dependency. Convolutional neural network (CNN) is used to capture the spatial interactions. Since treating the entire city as an image and simply apply CNN may not achieve the best performance. Including regions with weak correlations to predict a target region actually hurts the performance. Thus, we use the local CNN to model the spatial dependency.

For each time interval t, we treat the target region i and its surrounding neighbors as a $S \times S$ image with two channels $\mathbf{Y}^i_t \in \mathbb{R}^{S\times S\times 2}$. One channel contains start volume information, another one is end volume information. The target region is in the center of the image. The local CNN takes $\mathbf{Y}^i_t$ as input $\mathbf{Y}^{i,0}_t$, and the formulation of each convolutional layer k is:

$$\mathbf{Y}^{i,k}_t = \mathcal{F}(\mathbf{W}^k_{s,t} * \mathbf{Y}^{i,k-1}_t + \mathbf{b}^k_{s,t}), \quad (5)$$

where $\mathbf{W}^k_{s,t}$ and $\mathbf{b}^k_{s,t}$ are learned parameters. $\mathcal{F}(x) = \max(x, 0)$ is a ReLU activation function. After stacking K convolutional layers, we use a flatten layer to transform the output $\mathbf{Y}^{i,K}_t$ of convolutional layer. Then, a fully connected layer is used to reduce the dimension of spatial representations. Finally, we get the spatial representation of region i at time interval t as $\mathbf{s}^i_t$.

4.1.2 Short-term Temporal Dependency. We use Long Short-Term Memory (LSTM) network to capture the temporal sequential dependency, which is proposed to address the exploding and vanishing gradient issue of traditional RNN. In this paper, we use the original version of LSTM [11] and formulate it as:

$$\mathbf{h}^i_t = \text{LSTM}(\mathbf{h}^i_{t-1}, \mathbf{g}^i_t), \quad (6)$$

where h^i_t is the output representation of region i at time interval t. We concatenate the spatial representation $\mathbf{s}^i_t$ with context features $\mathbf{e}^i_t$ and use it as the input $\mathbf{g}^i_t$ of LSTM. We denote the output of the last time interval as $\mathbf{h}^i_{t,so}$, which is the representation of short-term information.

4.2 Spatial Dynamic Similarity: Flow Gating Mechanism

As we described before, local CNN is used to capture the spatial dependency. CNN handles the local structure similarity by local connection and weight sharing. In local CNN, the local spatial dependency relies on the similarity of historical traffic volume. However, the spatial dependency of volume is a static dependency, which can not fully reflect the relation between the target region and its neighbors. A more direct way to represent interactions between regions is traffic flow. If there are more flows existing between two regions, the relation between them is stronger (i.e., they are more similar). Traffic flow can be used to explicitly control the volume information propagation between regions. Therefore, we design a

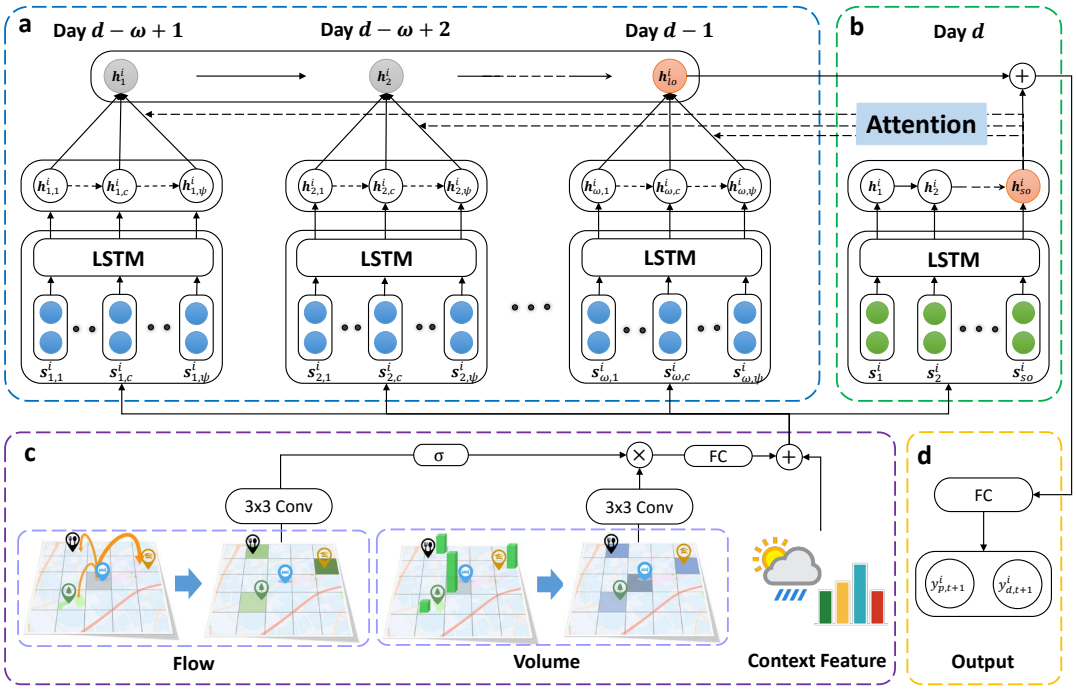


Figure 1: The architecture of STDN. (a) Periodically shifted attention mechanism captures the long-term periodic dependency and temporal shifting. For each day, we also use LSTM to capture the sequential information. (b) The short-term temporal dependency is captured by one LSTM. (c) The flow gating mechanism tracks the dynamic spatial similarity representation by controlling the spatial information propagation. (d) A unified multi-task prediction component predicts two types of traffic volumes simultaneously.

Flow Gating Mechanism (FGM), which explicitly capture dynamic spatial dependency in the hierarchy.

In Eq. (3), we define the flow from region i to region j as $f_t^{i,j}$. Similar to local CNN, we construct the local spatial flow image for each region i at time interval t to protect the spatial dependency of flow. The size of each flow image is also $S \times S$, and the target region is in the center. Additionally, the local flow image consists of inflow (i.e., $\{f_t^{1,i}, ..., f_t^{S\times S,i}\}$) and outflow (i.e., $\{f_t^{i,1}, ..., f_t^{i,S\times S}\}$) information. At each time interval t, we consider the flow happens from time $t-r+1$ to t. Thus, we get total $2r$ flow images and concatenate them as $\mathbf{F}_t^i \in \mathbb{R}^{S\times S\times 2r}$. Then, we use CNN to capture the spatial interaction of flow between regions, which takes $\mathbf{F}_t^i$ as input $\mathbf{F}_t^{i,0}$. For each layer k, the formulation is

$$\mathbf{F}_t^{i,k} = \mathcal{F}(\mathbf{W}_{f,t}^k * \mathbf{F}_t^{i,k-1} + \mathbf{b}_{f,t}^k), \tag{7}$$

where $\mathbf{W}_{f,t}^k$ and $\mathbf{b}_{f,t}^k$ are learned parameters.

At each layer, we use flow information to explicitly capture dynamic similarity between regions by constricting the spatial information via a flow gate. Specifically, the output of each layer is the spatial representation $\mathbf{G}_t^{i,k}$ modulated by the flow gate, which is formulated as:

$$\mathbf{G}_t^{i,k+1} = \mathbf{G}_t^{i,k} \otimes \sigma(\mathbf{F}_t^{i,k}), \tag{8}$$

where $\otimes$ is the element-wise product between tensors. Note that, $\mathbf{G}_t^{i,0} = \mathbf{Y}_t^{i,0}$ is the original volume image.

Then, we use a flatten layer to transfer the output $\mathbf{G}_t^{i,K}$ to a feature vector $\hat{\mathbf{G}}_t^{i,K}$ for region i and time interval t. At last, we use a fully connected layer to reduce the dimension and get the dynamic flow gated spatial representation $\hat{\mathbf{s}}_t^i$, which is formulated as

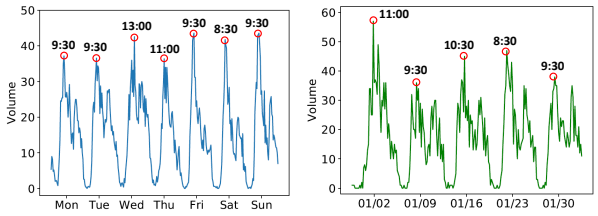
$$\hat{\mathbf{s}}_t^i = \mathcal{F}(\mathbf{W}_{r,t}\hat{\mathbf{G}}_t^i + \mathbf{b}_{r,t}), \tag{9}$$

where $\mathbf{W}_{r,t}$ and $\mathbf{b}_{r,t}$ are two learned parameter sets at time interval t. We replace the spatial representation $\mathbf{s}_t^i$ defined in Section 4.1.1 by $\hat{\mathbf{s}}_t^i$, and concatenate with context features $\mathbf{e}_t^i$ as the input $\mathbf{g}_t^i$ of LSTM.

4.3 Temporal Dynamic Periodic Similarity: Periodically Shifted Attention Mechanism

In local spatial-temporal network defined in Section 4.1, only previous several time intervals (usually several hours) are used for prediction. However, it overlooks the long-term dependency (e.g., periodicity), which is an important property of spatial-temporal prediction problem. In this section, we take long-term periodic information into consideration.

Simply using long-term information makes the length of LSTM very long and difficult to train. The gradient vanishing of too long sequence also weakens the periodic influence. Thus, only the relative time intervals of the predicted time (i.e., the same time of previous days) in previous days are considered.



(a) (b)

Figure 2: The temporal shifting of periodicity. (a) Temporal shifting between different days. (b) Temporal shifting between different weeks. Note that, each time in these figures represents a time interval (e.g., 9:30 means 9:00-9:30).

However, considering the same time of previous days raises another problem called temporal shifting of periodicity, i.e., traffic data is not strictly periodic. For example, the peak hours on weekdays are usually in the afternoon, but could vary from 4:30pm to 6:00pm. We show one example of temporal shifting between different days and weeks in Figure 2a and Figure 2b, respectively. These two time series are start volume near the Jacob K. Javits Convention Center, which are calculated from New York Taxi Trips [2]. Clearly, the traffic series are periodic but the peaks of traffic series (i.e., marked by the red circle) exist in different time intervals of the day. Furthermore, comparing these two figures, the periodicity is not strict daily or weekly. Thus, we design a **P**eriodically **S**hifted **A**ttention **M**echanism (**PSAM**) to tackle the problem. The details are described as follows.

As shown in Figure 1(a), we consider previous ω days for handling the periodic dependency. For each day, in order to tackle the temporal shifting problem, Ψ time intervals are selected as the predicted time is in these time intervals. Additionally, we use LSTM to protect the sequential information for each day, which is formulated as:

$$\mathbf{h}^i_{t,\omega_\psi} = \text{LSTM}(\mathbf{h}^i_{t,\omega_{\psi-1}}, \mathbf{g}^i_{t,\omega_\psi}). \tag{10}$$

We adopt an attention mechanism to capture the temporal shifting and get the weighted representation of each previous day. Formally, the representation of each previous days is a weighted sum of hidden representations and defined as:

$$\mathbf{h}^i_{t,\omega} = \sum_{\psi=1}^{\Psi} \alpha^i_{\omega_\psi} \mathbf{h}^i_{t,\omega_\psi}, \tag{11}$$

where $\alpha^i_{\omega_\psi}$ measures the importance of the time ω_ψ for predicted time at previous time j. The importance value $\alpha^i_{\omega_\psi}$ is derived by comparing the learned spatial-temporal representation from short-term memory (i.e., Eq. (6)) with previous hidden state $\mathbf{h}^i_{t,\omega_\psi}$. Formally, the weight $\alpha^i_{\omega_\psi}$ is defined as

$$\alpha^i_{\omega_\psi} = \frac{\exp(\text{score}(\mathbf{h}^i_{t,\omega_\psi}, \mathbf{h}^i_{t,so}))}{\sum_{\psi'=1}^{\Psi} \exp(\text{score}(\mathbf{h}^i_{t,\omega_{\psi'}}, \mathbf{h}^i_{t,so}))}. \tag{12}$$

In this work, similar to [19], the score function is regarded as content-based function, which is defined as

$$\text{score}(\mathbf{h}^i_{t,\omega_\psi}, \mathbf{h}^i_{t,so}) = \mathbf{v}^T \tanh(\mathbf{W_H}\mathbf{h}^i_{t,\omega_\psi} + \mathbf{W_X}\mathbf{h}^i_{t,so} + \mathbf{b_X}), \tag{13}$$

where $\mathbf{W_H}, \mathbf{W_X}, \mathbf{b_X}, \mathbf{v}$ are learned parameters, $\mathbf{v}^T$ denotes the transpose of $\mathbf{v}$. For each previous day ω, we get the periodic representation $\mathbf{h}^i_{t,\omega}$. Then, we use another LSTM to preserve the sequential information by using these periodic representations as input, i.e.,

$$\mathbf{h}^i_{\omega,lo} = \text{LSTM}(\mathbf{h}^i_{\omega-1,lo}, \mathbf{h}^i_{t,\omega}). \tag{14}$$

We regard the output of the last time interval as the representation of temporal dynamic similarity (i.e., long-term periodic information), and denote it as $\mathbf{h}^i_{t,lo}$.

4.4 Joint Training

We concatenate the short-term representation $\mathbf{h}^i_{t,so}$ and long-term representation $\mathbf{h}^i_{t,lo}$ as $\mathbf{h}^i_{t,r}$, i.e., $\mathbf{h}^i_{t,r} = \mathbf{h}^i_{t,so} \oplus \mathbf{h}^i_{t,lo}$. Then we feed $\mathbf{h}^i_{t,r}$ to a fully connected network and get the final prediction value of start and end traffic volume for each region i, which is denoted as $y^i_{s,t+1}$ and $y^i_{e,t+1}$, respectively. The final prediction function is defined as:

$$[y^i_{s,t+1}, y^i_{e,t+1}] = \tanh(\mathbf{W}_{fa}\mathbf{h}^i_{t,r} + \mathbf{b}_{fa}), \tag{15}$$

where $\mathbf{W}_{fa}$ and $\mathbf{b}_{fa}$ are learned parameters. The output of our model is (-1,1) since we normalize the value of pick-up and drop-off volume. We later denormalize the prediction to get the actual demand values.

4.5 Optimization

Since we predict start volume and end traffic volume simultaneously, the loss function is defined as:

$$\mathcal{L}(\theta) = \sum_{i=1}^{\xi} \lambda(y^i_{s,t+1} - \hat{y}^i_{s,t+1})^2 + (1-\lambda)(y^i_{e,t+1} - \hat{y}^i_{e,t+1})^2, \tag{16}$$

where θ are all learned parameters in STDN. λ is a parameter to balance the influence of start and end. In this study, λ is set as 0.5. STDN is optimized via Backpropagation Through Time (BPTT) and Adam [14]. We use Tensorflow [3] and Keras [6] to implement our proposed model.

4.6 Extension: Road Intersection Volume Prediction

The traffic volume of road intersection is another important type of traffic volume, which can be recorded by cameras. The traffic volume of adjacent intersections is highly correlated. Thus, road network structure captures the spatial dependency of traffic volume. In this section, we extend our framework to incorporate road network information.

We use graph convolution to handle the spatial information on the road network. The road network is defined as $\mathcal{G} = (V, E, W)$ with N nodes, the graph convolution is defined in the spectral domain. The Laplacian matrix of the graph $\mathcal{G}$ is defined as $L = D - W$ and the eigen decomposition is $L = U\Lambda U^T$, where D is degree matrix. Based on the fact that γth power of Laplacian is supported by exactly γ-hop neighbors [26], we calculate the γth power of Laplacian via Chebyshev polynomial expansion [8] for computational efficient as follows:

$$\mathbf{Y}^{i,k}_t = \sum_{\gamma=0}^{\Gamma-1} \theta_\gamma L^\gamma \mathbf{Y}^{i,k-1}_t \approx \sum_{\gamma=0}^{\Gamma-1} \tilde{\theta}_\gamma T_\gamma(\tilde{\Lambda})\mathbf{Y}^{i,k-1}_t, \tag{17}$$

where $\tilde{\theta}_\gamma \in \mathbb{R}^\Gamma$ is a vector of Chebyshev coefficients. $T_\gamma(\tilde{\Lambda}) \in \mathbb{R}^{S\times S}$ is the Chebyshev polynomial of order γ, which is evaluated as $\tilde{\Lambda} = \frac{2\Lambda}{\lambda_{\max}} - \mathbf{I}$. For more details about graph convolution, please refer to [8].

Furthermore, we also extend graph convolution to local graph convolution. For each intersection i, we choose the nearest S road intersections based on Euclidean distance, i.e., the number of nodes for each local graph is S. In order to extend our model, we replace the convolutional neural network as graph convolutional neural network. Both convolutional operators on flow image and volume image are replaced by graph convolutional operators. The experiment of this extension part is described in Section 5.8.

5 EXPERIMENT

In this section, we conduct experiments on two real datasets. We show a comprehensive quantitative evaluation by comparing with other methods and also show the effectiveness of flow gating mechanism and periodically shifted attention mechanism[1].

5.1 Datasets

We evaluate our proposed method on two large-scale public real-world datasets from New York City (NYC). Each dataset contains trip records, as detailed follows.

- **NYC-Taxi:** NYC-Taxi dataset contains 22, 349, 490 taxi trip records of NYC [2] in 2015, from 01/01/2015 to 03/01/2015. In the experiment, we use data from 01/01/2015 to 02/10/2015 (40 days) as training data, and the remained 20 days as testing data.
- **NYC-Bike:** The bike trajectories are collected from NYC Citi Bike system [1] in 2016, from 07/01/2016 to 08/29/2016. The dataset contains 2, 605, 648 trip records. The previous 40 days (i.e., from 07/01/2016 to 08/09/2016) are used as training data, and the rest 20 days as testing data.

5.2 Evaluation Metrics

In our experiment, we use Mean Average Percentage Error (MAPE) and Rooted Mean Square Error (RMSE) as the evaluation metrics, which are the same metrics used in literature [27, 32, 35]. These metric are defined as follows:

$$MAPE = \frac{1}{\xi}\sum_{i=1}^{\xi}\frac{|\hat{y}_{t+1}^i - y_{t+1}^i|}{y_{t+1}^i}, RMSE = \sqrt{\frac{1}{\xi}\sum_{i=1}^{\xi}(\hat{y}_{t+1}^i - y_{t+1}^i)^2}, \tag{18}$$

where $\hat{y}_{t+1}^i$ and y_{t+1}^i represent the prediction value and real value of region i for time interval $t+1$, and ξ is total number of samples.

5.3 Compared Algorithms

We compare our model with the following three categories of spatial-temporal prediction methods. Moreover, multiple types of traffic volume (e.g., start volume and end volume of bike trips) are predicted jointly in all baselines for fair comparison. For each method, we tune some key hyperparameters in our scenario. For the rest hyperparameters, we follow the setting in their original papers.

[1]Code and data: http://tangxianfeng.net

5.3.1 *Traditional time-series prediction methods.*

- **Historical average (HA):** Historical average predicts traffic for a given region basing on the average values of the previous relative time interval in the same region.
- **Autoregressive integrated moving average (ARIMA):** ARIMA considers moving average and autoregressive components.

5.3.2 *Regression-based methods.*

- **Ridge regression (Ridge):** We use Ridge (i.e., with ℓ_2-norm regularization) as the linear regression method.
- **LinUOTD** [27]: LinUOTD is a linear regression model with a spatial-temporal regularization.
- **XGBoost** [4]: XGBoost is a widely used boosting tree method. We set the number of trees is 500, the maximum depth of each tree is 4, the subsample rate is 0.6.

5.3.3 *Neural-network-based methods.*

- **MultiLayer Perceptron (MLP)**: We compare our method with a neural network with four fully connected layers. The hidden units of each layer is 128, 128, 64, 64.
- **ConvLSTM** [25]: ConvLSTM extends the fully connected LSTM to have convolutional structures in each transition.
- **DeepSD** [28]: DeepSD is an artificial neural network model to predict the gap between taxi demand and supply. It can be regarded as an extension model of MLP which considers the supply-demand patterns, weather and traffic information. Note that we do not have weather and traffic information, we remove these components for comparison.
- **ST-ResNet** [35]: ST-ResNet is a CNN-based deep learning framework for traffic prediction. The model uses CNN to capture trend, period, and closeness information. We set the length trend, period and closeness as 4, 4, 4, respectively;
- **DMVST-Net** [32]: DMVST-Net is a deep learning based model for taxi demand prediction. The method consists of three views: temporal view (use LSTM), spatial view (use local CNN), and semantic view (use semantic network embedding).

5.4 Settings

5.4.1 Preprocessing: We split the whole city as 10×20 regions. The size of each region is about $1km \times 1km$. The length of each time interval is set as 30 minutes. We use Min-Max normalization to convert traffic volume and flow to $[0, 1]$ scale. After prediction, we denormalize the prediction value and use it for evaluation. We use a sliding window on both training and testing data for sample generation. When testing our model, we filter the samples with volume values less than 10, which a common practice used in industry and academy [32]. Because in the real-world applications, cares with low traffic are of little interest.

5.4.2 Context features: We choose similar types of context features used in [27, 32]. These features include temporal features (e.g., volume values of previous several time intervals) and spatial features (e.g., volume values of nearby regions). Note that, we use the same features for all compared methods above.

5.4.3 Hyperparameters Setting: We select 80% of the training data to learn the models, and the remaining 20% for validation. For spatial information, we set all convolution kernel sizes to 3×3

Table 1: Comparison with Different Baselines: NYC-Taxi

Method	Start/Pick-up		End/Drop-off	
	RMSE	MAPE	RMSE	MAPE
Historical average	43.82	23.18%	33.83	21.14%
ARIMA	36.53	22.21%	27.25	20.91%
Ridge Regression	28.51	19.94%	24.38	20.07%
LinUOTD [27]	28.48	19.91%	24.39	20.03%
XGBoost [4]	26.07	19.35%	21.72	18.70%
MLP	26.67	18.43%	22.08	18.31%
ConvLSTM [25]	28.13	20.50%	23.67	20.70%
DeepSD [28]	26.35	18.12%	21.95	18.15%
ST-ResNet [35]	26.23	21.13%	21.63	21.09%
DMVST-Net [32]	25.71	17.36%	20.50	17.11%
STDN	**24.10**	**16.30%**	**19.05**	**16.25%**

Table 2: Comparison with Different Baselines: NYC-Bike

Method	Start		End	
	RMSE	MAPE	RMSE	MAPE
Historical average	12.49	27.82%	11.93	27.06%
ARIMA	11.53	26.35%	11.25	25.79%
Ridge Regression	10.92	25.29%	10.33	24.58%
LinUOTD [27]	11.04	25.22%	10.44	24.44%
XGBoost [4]	9.57	23.52%	8.94	22.54%
MLP	9.83	23.12%	9.12	22.40%
ConvLSTM [25]	10.40	25.10%	9.22	23.20%
DeepSD [28]	9.69	23.62%	9.08	22.36%
ST-ResNet [35]	9.80	25.06%	8.85	22.98%
DMVST-Net [32]	9.12	22.17%	8.49	21.55%
STDN	**8.85**	**21.84%**	**8.15**	**20.87%**

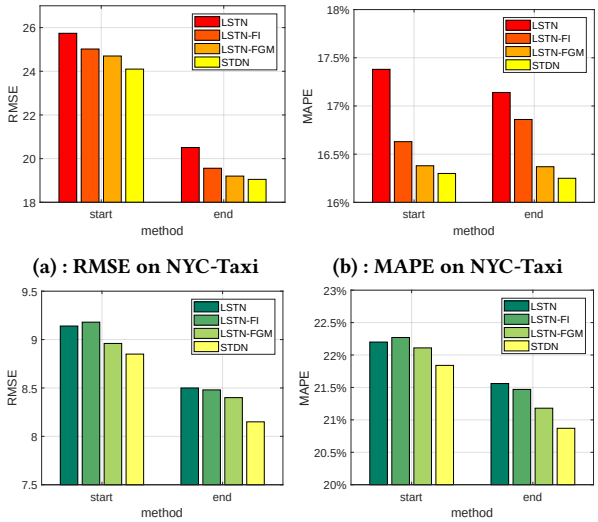
with 64 filters. The size of each neighborhood considered was set as 7×7 (i.e., S=7), which corresponds to about $7km \times 7km$ rectangles. We set $K = 3$ (number of layers), $r = 2$ (the time span of considered flow) and $d = 64$ (dimension of the output). For temporal information, we set $t = 7$ for short-term LSTM (i.e., previous 3.5 hours), $\omega = 3$ for long-term periodic information (i.e., previous 3 days), $\Psi = 3$ for periodically shifted attention mechanism (i.e., half an hour before and after of relative predicted time are considered). The dimension of hidden output of LSTM is 128. The batch size in our experiment is set to 64. Learning rate is set as 0.001. Both dropout and recurrent dropout rate in LSTM are set as 0.5. We also use early-stop in all the experiments.

5.5 Performance Comparison

Table 1 and 2 show the performance of our proposed method as compared to all other competing methods in NYC-Taxi and NYC-Bike datasets, respectively. We run each baseline 10 times and report the average results of each baseline. Our proposed STDN outperforms all competing baselines by achieving the lowest RMSE and MAPE on both datasets.

Specifically, the traditional time-series prediction methods (historical average and ARIMA) do not perform well, because they only rely on historical records of predicting value and overlook spatial and other context features. For regression-based methods (Ridge, LinUOTD, XGBoost), they further consider spatial correlations as features or regularizations. As a result, they achieve better performances compared with other conventional time-series approaches. However, they fail to capture the complex non-linear temporal dependencies and the dynamic spatial relationships. Therefore, our proposed method significantly outperforms those regression-based methods.

For neural-network-based methods, our proposed model achieves better performance than MLP and DeepSD. The potential reason is that MLP and DeepSD do not explicitly model spatial dependency and temporal sequential dependency. Also, our model outperforms ST-ResNet, because ST-ResNet uses CNN to capture spatial information, but overlooks the temporal sequential dependency. ConvLSTM extends fully connected LSTM by integrating convolutional operation to LSTM units for capturing both spatial and temporal information. DMVST-Net considers spatial-temporal information by local CNN and LSTM. However, these two models overlook the dynamic spatial similarity and periodic temporal shifting. The better performance of our proposed model demonstrates the effectiveness of flow gating mechanism and periodically shifted attention mechanism to capture the dynamic spatial-temporal similarity.



(a) : RMSE on NYC-Taxi **(b) : MAPE on NYC-Taxi**

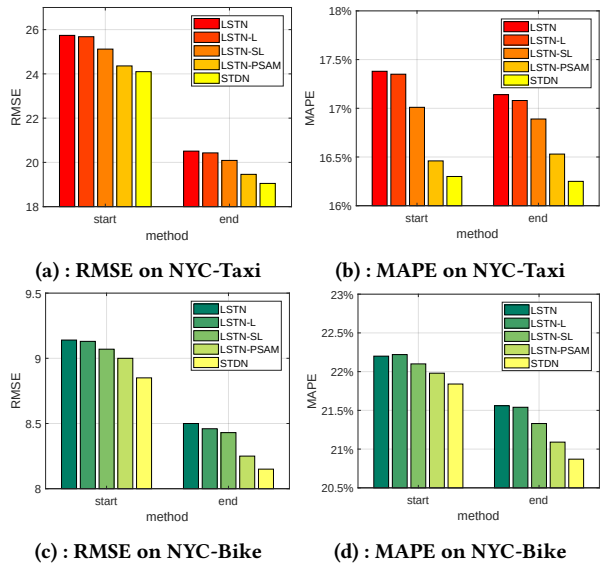
(c) : RMSE on NYC-Bike **(d) : MAPE on NYC-Bike**

Figure 3: Evaluation of flow gating mechanism (FGM) and its variants. LSTN: local CNN + LSTM; LSTN-FI: LSTN + Flow Feature; LSTN-FGM: LSTN + FGM; STDN: proposed model.

5.6 Effectiveness of Flow Gating Mechanism

In this section, we study the effectiveness of flow gating mechanism. We first list some variants of using traffic flow information as follows:

- **LSTN**: As described in Section 4.1.1, only short-term temporal dependency, and local spatial dependency are considered.
- **LSTN-FI**: LSTN-FI use traffic flow information as features. We simply concatenate flow information $F_t^{i,K}$ defined in Eq. (7) and



(a) : RMSE on NYC-Taxi (b) : MAPE on NYC-Taxi

(c) : RMSE on NYC-Bike (d) : MAPE on NYC-Bike

Figure 4: Evaluation of periodically shifted attention mechanism (PSAM) and its variants. LSTN: local CNN+LSTM; LSTN-L: one LSTN for long and short; LSTN-SL: short-term LSTN + long-term LSTN; LSTN-PSAM: LSTN + PSAM; STDN: proposed model.

spatial representation $\mathbf{Y}_t^{i,K}$. Then we feed it in to LSTM as spatial features instead of using a flow gating mechanism.

- **LSTN-FGM**: FGLSTN further utilize flow gating mechanism to represent the spatial dynamic similarity between local neighborhoods. The variant does not use periodically shifted attention mechanism.

The results of different variants in NYC-Taxi an NYC-Bike are shown in Figure 3a, 3b and Figure 3c, 3d, respectively. LSTN-FGM and STDN outperform LSTN, because LSTN overlooks the dynamic spatial similarity between regions (e.g., traffic flow). In order to model dynamic spatial similarity, a straightforward approach would be using local flow as another type of spatial representation, i.e., the variants LSTN-FI. However, compared to LSTN-FGM, which uses flow gating mechanism, LSTN-FI performs worse. One potential reason is that only using traffic flow as features can not incorporate the structure of spatial dynamic similarity. The results reveal the effectiveness of flow gating mechanism to explicitly capture the dynamic spatial similarity. Furthermore, the comparison to STDN demonstrates the importance of tackle temporal shifted periodic information.

5.7 Effectiveness of Periodically Shifted Attention Mechanism

The intuition of periodically shifted attention mechanism is the long-term periodic information and temporal shifting. In this section, we analyze the effectiveness of periodically shifted attention mechanism and several variants are listed as follows:

- **LSTN-L**: We extend LSTN by taking long-term sequential information into consideration. The long-term information (i.e., the information of relative predicted time in previous 3 days) are concatenated with short-term information (i.e., the information of previous 7 time intervals) and use one LSTM network as prediction component.
- **LSTN-SL**: LSTN-SL removes the periodically shifted attention mechanism in STDN. LSTN-SL consists of two LSTM network. One is used to capture short-term dependency, and another one uses relative time in previous 3 days information to capture long-term information. Note that we set $\Psi = 1$ (only relative predicted time in previous 3 days are considered) and LSTN-SL does not include flow gating mechanism.
- **LSTN-PSAM**: We add the periodically shifted attention mechanism attention on LSTN-SL. Compared to proposed STDN, this variant only removes the flow gating mechanism.

Table 3: Comparison with Different Baselines on Road Intersection Volume Prediction

Method	RMSE	MAPE
Historical average	80.33	39.53%
ARIMA	38.93	23.09%
Linear Regression	34.69	20.35%
MLP	31.55	19.09%
XGBoost [4]	31.72	19.97%
LinUOTD [27]	34.58	20.33%
DeepSD [28]	31.15	19.01%
DCRNN [17]	30.75	18.98%
STDN-Graph	**28.95**	**17.84%**

Figures 4a, 4b and Figures 4c, 4d show the comparison results in NYC-Taxi and NYC-Bike, respectively. We also show LSTN (described in Section 5.6) and STDN (our proposed model) for comparison. The results for LSTN and LSTN-L are similar. One potential reason is that when the long term information is concatenated before short term information in LSTM, only the short term information can be remembered. The other reason is the uneven time gap between long term information and short term information might be harmful for learning the periodic sequence. In one LSTM network, sequences with different sample rate may not achieve good performance. LSTN-SL further split the long-term and short-term information and use two LSTM networks to handle these dependencies. We can see that LSTN-SL performs better than LSTN-L, which shows the effectiveness of considering long-term and short-term information separately. Furthermore, the improvement from LSTN-PSAM to LSTN-SL shows the influence of temporal shifting. Using the proposed periodically shifted attention mechanism can capture the temporal shifting and improve the performance. Finally, the better performance of STDN than LSTN-PSAM further shows the effectiveness of flow gating mechanism.

5.8 Extension Experiments: Road Intersection Volume Prediction

As described in Section 4.6, our framework can be easily extended to traffic volume prediction on road intersection. In this section, we test the extended framework STDN-Graph.

We collect the road camera dataset from Jinan, China. Vehicles' trajectories are recorded when passing through the road intersections. The dataset contains records during 08/01/2017 to 08/31/2017 collected from more than 900 road intersections. The

dataset contains more than 9 billion vehicles' records in these intersections. We use previous 25 days as training and the rest 6 days as testing. The length of time interval is set as 10 minutes.

Since ConvLSTM, ST-ResNet and DMVST-Net are based on region volume prediction, we only compare with other baselines described in Section 5.3. In addition, we add one neural-network baseline DCRNN [17], which extends GRU with diffusion graph convolutional neural network.

Table 3 shows the results of our model on road intersection volume prediction. Similar to previous reasons mentioned in Section 5.5, our model still achieves better performance than traditional time-series prediction methods (HA, ARIMA), regression-based methods (Ridge, LinUOTD, XGBoost) and MLP. Furthermore, our proposed method outperforms DCRNN. One reason is that our model further consider dynamic spatial similarity and temporal periodic information.

6 CONCLUSION

In this paper, we propose a novel Spatial-Temporal Dynamic Network (STDN) for traffic prediction. Our approach tracks the dynamic spatial similarity between regions by flow gating mechanism and temporal periodic similarity by periodically shifted attention mechanism. The evaluation on two large-scale datasets show that our proposed model significantly outperforms the state-of-the-art methods. Furthermore, the experiments on road intersection volume prediction demonstrate our model can be easily extended to the road network and also achieves better performance.

REFERENCES

[1] 2017. NYC Bike:. https://www.citibikenyc.com/system-data. (2017).
[2] 2017. NYC Taxi:. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml. (2017).
[3] Martín Abadi et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). http://tensorflow.org/ Software available from tensorflow.org.
[4] 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*. ACM, 785–794.
[5] Xingyi Cheng, Ruiqing Zhang, Jie Zhou, and Wei Xu. 2017. DeepTransport: Learning Spatial-Temporal Dependency for Traffic Condition Forecasting. *arXiv preprint arXiv:1709.09585* (2017).
[6] François Chollet et al. 2015. Keras. https://github.com/fchollet/keras. (2015).
[7] Zhiyong Cui, Ruimin Ke, and Yinhai Wang. 2016. Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. In *ACM SIGKDD Workshop on Urban Computing*.
[8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Annual Conference on Neural Information Processing Systems*. 3844–3852.
[9] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, Linhong Zhu, Rose Yu, and Yan Liu. 2016. Latent space model for road networks to predict time-varying traffic. *KDD* (2016).
[10] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 6645–6649.
[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[12] Tsuyoshi Idé and Masashi Sugiyama. 2011. Trajectory regression on road networks. In *AAAI Conference on Artificial Intelligence*. AAAI Press, 203–208.
[13] Jintao Ke, Hongyu Zheng, Hai Yang, and Xiqun Michael Chen. 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies* 85 (2017), 591–608.
[14] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.
[16] Xiaolong Li, Gang Pan, Zhaohui Wu, Guande Qi, Shijian Li, Daqing Zhang, Wangsheng Zhang, and Zonghui Wang. 2012. Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science* 6, 1 (2012), 111–121.
[17] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *Sixth International Conference on Learning Representations*.
[18] Marco Lippi, Matteo Bertini, and Paolo Frasconi. 2013. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems* 14, 2 (2013), 871–882.
[19] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).
[20] Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang. 2017. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* 17, 4 (2017), 818.
[21] Luis Moreira-Matias, Joao Gama, Michel Ferreira, Joao Mendes-Moreira, and Luis Damas. 2013. Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems* 14, 3 (2013), 1393–1402.
[22] Bei Pan, Ugur Demiryurek, and Cyrus Shahabi. 2012. Utilizing real-world transportation data for accurate traffic prediction. In *ICDM*. IEEE, 595–604.
[23] Liefeng Rong, Hao Cheng, and Jie Wang. 2017. Taxi Call Prediction for Online Taxicab Platforms. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data*. Springer, 214–224.
[24] Shashank Shekhar and Billy Williams. 2008. Adaptive seasonal time series models for forecasting short-term traffic flow. *Transportation Research Record: Journal of the Transportation Research Board* 2024 (2008), 116–125.
[25] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. (2015), 802–810.
[26] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine* 30, 3 (2013), 83–98.
[27] Yongxin Tong, Yuqiang Chen, Zimu Zhou, Lei Chen, Jie Wang, Qiang Yang, and Jieping Ye. 2017. The Simpler The Better: A Unified Approach to Predicting Original Taxi Demands on Large-Scale Online Platforms. In *KDD*. ACM.
[28] Dong Wang, Wei Cao, Jian Li, and Jieping Ye. 2017. DeepSD: Supply-Demand Prediction for Online Car-Hailing Services Using Deep Neural Networks. In *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*. IEEE, 243–254.
[29] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.
[30] Hua Wei, Yuandong Wang, Tianyu Wo, Yaxiao Liu, and Jie Xu. 2016. ZEST: a Hybrid Model on Predicting Passenger Demand for Chauffeured Car Service. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2203–2208.
[31] Fei Wu, Hongjian Wang, and Zhenhui Li. 2016. Interpreting traffic dynamics using ubiquitous urban data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 69.
[32] 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. *AAAI Conference on Artificial Intelligence* (2018).
[33] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting. *arXiv preprint arXiv:1709.04875* (2017).
[34] Rose Yu, Yaguang Li, Ugur Demiryurek, Cyrus Shahabi, and Yan Liu. 2017. Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting. In *Proceedings of SIAM International Conference on Data Mining*.
[35] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. *AAAI* (2017).
[36] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. 2016. DNN-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 92.
[37] Jiangchuan Zheng and Lionel M Ni. 2013. Time-dependent trajectory regression on road networks via multi-task learning. In *AAAI Conference on Artificial Intelligence*. AAAI Press, 1048–1055.
[38] Xian Zhou, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. 2018. Predicting Multi-step Citywide Passenger Demands Using Attention-based Neural Networks. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 736–744.