Revisiting Spatial-Temporal Similarity: 
A Deep Learning Framework for Traffic Prediction

Huaxiu Yao,* Xianfeng Tang,* Hua Wei, Guanjie Zheng, Zhenhui Li
Pennsylvania State University
{huaxiuyao, xianfeng, hzw77, gjz5038, jessieli}@ist.psu.edu

Abstract
Traffic prediction has drawn increasing attention in AI research field due to the increasing availability of large-scale traffic data and its importance in the real world. For example, an accurate taxi demand prediction can assist taxi companies in pre-allocating taxis. The key challenge of traffic prediction lies in how to model the complex spatial dependencies and temporal dynamics. Although both factors have been considered in modeling, existing works make strong assumptions about spatial dependence and temporal dynamics, i.e., spatial dependence is stationary in time, and temporal dynamics is strictly periodical. However, in practice the spatial dependence could be dynamic (i.e., changing from time to time), and the temporal dynamics could have some perturbation from one period to another period. In this paper, we make two important observations: (1) the spatial dependencies between locations are dynamic; and (2) the temporal dependency follows daily and weekly pattern but it is not strictly periodic for its dynamic temporal shifting. To address these two issues, we propose a novel Spatial-Temporal Dynamic Network (STDN), in which a flow gating mechanism is introduced to learn the dynamic similarity between locations, and a periodically shifted attention mechanism is designed to handle long-term periodic temporal shifting. To the best of our knowledge, this is the first work that tackle both issues in a unified framework. Our experimental results on real-world traffic datasets verify the effectiveness of the proposed method.

Introduction
Traffic prediction - a spatial temporal prediction problem, has drawn increasing attention due to the growing traffic related datasets and for its impacts in real-world applications. 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 congestion.

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), one needs 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 (Li et al. 2012; Moreira-Matias et al. 2013; Shekhar and Williams 2008; Lippi, Bertini, and Frasconi 2015). While these earlier methods study traffic time series for each individual location, separately, recent studies started taking into account spatial information (e.g., adding regularizations on model similarity for nearby locations) (Deng et al. 2016; Idé and Sugiyama 2011; Zheng and Ni 2013) and external context information (e.g., adding features of venue information, weather condition, and local events) (Wu, Wang, and Li 2016; Pan, Demiryurek, and Shahabi 2012; Tong et al. 2017). 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 made achieved tremendous success in many challenging learning tasks (LeCun, Bengio, and Hinton 2015). The success has inspired several studies to apply deep learning techniques to traffic prediction problem. For example, several studies (Zhang, Zheng, and Qi 2017; Zhang et al. 2016; Ma et al. 2017) have modeled city-wide traffic as a heatmap image and use convolutional neural network (CNN) to model the non-linear spatial dependency. To model non-linear temporal dependency, researchers propose to use recurrent neural network (RNN)-based framework (Yu et al. 2017; Cui, Ke, and Wang 2016). Yao et al. further propose a method to jointly model both spatial and temporal dependencies by integrating CNN and long short-term memory (LSTM) (Yao et al. 2018).

Although both spatial dependency and temporal dynamics have been considered in deep learning for traffic prediction, the existing method have two major limitations. First, the spatial dependency between locations relies only on the similarity of historical traffic (Zhang, Zheng, and Qi 2017; Yao et al. 2018) 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 could be strong; whereas in late evening, the relation between these two places might be very weak. However, such dynamic dependencies have not been considered in previous
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 based on such periodicity can be useful for prediction. However, one challenge is that the traffic data are not strictly periodic. For example, the peak hours on weekdays usually happen in the late afternoon, but could vary from 4:30pm to 6:00pm on different days. Though previous studies (Zhang, Zheng, and Qi 2017; Zhang et al. 2016) consider periodicity, 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.

We evaluate the proposed method on large-scale real-world public datasets including taxi data of New York City (NYC) and bike-sharing data of NYC. The comprehensive comparisons with the state-of-the-art methods demonstrate the effectiveness of our proposed method. Our contributions are summarized below:

- We propose a flow gating mechanism to explicitly model dynamic spatial similarity. The gate controls information propagation among nearby locations.
- We propose a periodically shifted attention mechanism by taking long-term periodic information and temporal shifting simultaneously.
- We conduct experiments on several real-world traffic datasets. The results show that our model is consistently better than other state-of-the-art methods.

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 (Shekhar and Williams 2008; Li et al. 2012; Moreira-Matias et al. 2013; Lippi, Bertini, and Frasconi 2013). Recent studies further explore the utilities of external context data, such as venue types, weather conditions, and event information (Pan, Demiryurek, and Shahabi 2012; Wu, Wang, and Li 2016; Rong, Cheng, and Wang 2017). In addition, spatial information has also been explicitly modeled in recent studies (Deng et al. 2016; Tong et al. 2017; Idé and Sugiyama 2011; Zheng and Ni 2013). 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 computer vision and natural language processing (LeCun, Bengio, and Hinton 2015). In traffic prediction, a series of studies have been proposed based on deep learning techniques. The first line of studies stacked several fully connected layers to fuse context data from multiple sources for predicting traffic demand (Wei et al. 2016), taxi supply-demand gap (Wang et al. 2017). These methods used extensive features, but do not model the spatial and temporal interactions explicitly.

The second line of studies applied convolutional structure to capture spatial correlation for traffic flow prediction (Zhang, Zheng, and Qi 2017; Zhang et al. 2016). The third line of studies used recurrent-neural-network-based model for modeling sequential dependency (Yu et al. 2017; Cui, Ke, and Wang 2016). 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 (Shi et al. 2015) to handle spatial and temporal dependency for taxi demand prediction (Ke et al.; Zhou et al. 2018). Yao et al. 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 (Yao et al. 2018). Based on road network, several studies extended traditional CNN and RNN structure to graph-based CNN and RNN for traffic prediction, such as graph convolutional GRU (Li et al. 2018), graph attention (Zhang et al. 2018). 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.

Notations and Problem Formulation

We split the whole city to an $a \times b$ grid map with $n$ regions in total ($n = a \times b$), and use $\{1, 2, \ldots, n\}$ to denote them. We split the whole time period (e.g., one month) into $m$ equal-length continuous time intervals. The moving trip of any individual, which is an essential part of the entire city-wide traffic, always departs from a region, and arrives at the destination one after a while. We define the start/end traffic volume for a region as the number of trips departing/arriving from/in the region during a fixed time interval. Formally, $y_{i,t}^s$ and $y_{i,t}^e$ stand for the start/end traffic volume for region $i$ during the $t$-th time interval. Moreover, aggregation of individual trips formulates the traffic flow, which describes time-enhanced movements between certain pair of regions. Formally, the traffic flow starting from region $i$ in time interval
Problem (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 $t + 1$.

Spatial-Temporal Dynamic Network

In this section, we describe the details for our proposed Spatial-Temporal Dynamic Network (STDN). Figure 1 shows the architecture of our proposed method.

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 (Yao et al. 2018). 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 pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-

t volumes (i.e., start and end volumes), we integrate and pre-
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; FC means fully connected layers and Conv means several convolutional layers. (d) A unified multi-task prediction component predicts two types of traffic volumes simultaneously.

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., 9:30am means 9:00-9:30am).

for prediction. However, it overlooks the long-term dependency (e.g., periodicity), which is an important property of spatial-temporal prediction problem (Zonoozi et al. 2018; Feng et al. 2018). In this section, we take long-term periodic information into consideration.

Training LSTM to handle long-term information is a non-trivial task, since the increasing length enlarges the risk of gradient vanishing, thus significantly weakens the effects of periodicity. To address this issue, relative time intervals of the predicting target (e.g., same time of yesterday, and the day before yesterday) should be explicitly modeled. However, purely incorporating relative time intervals is insufficient ignores 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. Temporal shifting of periodic information is ubiquitous in traffic sequence because of accident or traffic congestion. An example of temporal shifting between different days and weeks is shown in Figure 2a and 2b, respectively. These two time series are start volume of the region containing Javits Center, calculated from New York Taxi Trips (nyc 2017b). Clearly, the traffic series are periodic but the peaks of those series (i.e., marked by the red circle) exist in different time of the day. Besides, comparing these two figures, the periodicity is not strict daily or weekly. Thus, we design a Periodically Shifted Attention Mechanism (PSAM) to tackle the limitations. The detailed approach is described as follows.

We focus on addressing the shifting in daily periodicity. As shown in Figure 1(a), relative time intervals from previous $P$ days are included for handling the periodic dependency. For each day, in order to tackle the temporal shifting problem, we further select $Q$ time intervals from each day in $Q$. For example, if the predicted time is 9:00-9:30pm, we select 1 hour before and after the predicted time (i.e., 8:00-10:30pm and $|Q| = 5$). These time intervals $q \in Q$ are used to tackle the potential temporal shifting. Additionally, we use LSTM to protect the sequential information for each day $p \in P$, which is formulated as:

$$h^{p,q}_{i,t} = \text{LSTM}([y^{p,q}_{i,t}; e^{p,q}_{i,t}; h^{p,q-1}_{i,t}]), \quad (5)$$

where $h^{p,q}_{i,t}$ is the representation of time $q$ in previous day $p$ for the predicted time $t$ in region $i$.

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 $H^{p}_{i,t}$ is a weighted sum of representations in each selected time...
interval \( q \), which is defined as:
\[
h_{i,t}^p = \sum_{q \in Q} \alpha_{i,t}^{p,q} h_{i,t}^{p,q},
\] (6)

where weight \( \alpha_{i,t}^{p,q} \) measures the importance of the time interval \( q \) in day \( p \) \( \in P \). The importance value \( \alpha_{i,t}^{p,q} \) is derived by comparing the learned spatial-temporal representation from short-term memory (i.e., Eq. (2)) with previous hidden state \( h_{i,t} \).

Formally, the weight \( \alpha_{i,t}^{p,q} \) is defined as:
\[
\alpha_{i,t}^{p,q} = \frac{\exp(\text{score}(h_{i,t}^{p,q}, h_{i,t}))}{\sum_{q' \in Q} \exp(\text{score}(h_{i,t}^{p,q'}, h_{i,t}))}.
\] (7)

In this work, similar to (Luong, Pham, and Manning 2015), the score function is regarded as content-based function defined as:
\[
\text{score}(h_{i,t}^{p,q}, h_{i,t}) = \mathbf{v}^T \tanh(\mathbf{W}_h h_{i,t}^{p,q} + \mathbf{W}_x h_{i,t} + \mathbf{b}_x),
\] (8)

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 \( p \), we get the periodic representation \( h_{i,t}^{p,q} \). Then, we use another LSTM to preserve the sequential information by using these periodic representations as input, i.e.,
\[
h_{i,t}^p = \text{LSTM}(h_{i,t}^{p,1}).
\] (9)

We regard the output of the last time interval \( h_{i,t}^p \) as the representation of temporal dynamic similarity (i.e., long-term periodic information).

**Joint Training**

We concatenate the short-term representation \( h_{i,t} \) and long-term representation \( h_{i,t}^p \) as \( \hat{h}_{i,t} \), which preserve both short-term and long-term dependencies for predicting region and time. Then we feed \( \hat{h}_{i,t} \) to a fully connected layer and get the final prediction value of start and end traffic volume for each region \( i \), which is denoted as \( \hat{y}_{s,i,t+1} \) and \( \hat{y}_{e,i,t+1} \), respectively.

The final prediction function is defined as:
\[
[y_{s,i,t+1}, y_{e,i,t+1}] = \tanh(\mathbf{W}_{fa} h_{i,t}^p + \mathbf{b}_{fa}),
\] (10)

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 start and end volume. We later denormalize the prediction to get the actual demand values.

In this work, we predict start volume and end traffic volume simultaneously, the loss function is defined as:
\[
\mathcal{L} = \sum_{i=1}^{n} \lambda(y_{s,i,t+1} - \hat{y}_{s,i,t+1})^2 + (1 - \lambda)(y_{e,i,t+1} - \hat{y}_{e,i,t+1})^2,
\] (11)

where \( \lambda \) is a parameter to balance the influence of start and end. The actual value of start and end volume in region \( i \) at time \( t+1 \) are denoted as \( \hat{y}_{s,i,t+1}, \hat{y}_{e,i,t+1} \).

**Experiment**

**Experiment Settings**

**Datasets** We evaluate our proposed method on two large-scale public real-world datasets from New York City (NYC).

- **NYC-Taxi**: NYC-Taxi dataset contains 22,349,490 taxi trip records of NYC (nyc 2017b) 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 (nyc 2017a) 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.

**Preprocessing** We split the whole city as \( 10 \times 20 \) 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 (Yao et al. 2018). Because in the real-world applications, cares with low traffic are of little interest. We select 80% of the training data to learn the models, and the remaining 20% for validation.

**Evaluation Metric & Baselines** In our experiment, two commonly metrics are used for evaluation: (1) Mean Average Percentage Error (MAPE) (2) Rooted Mean Square Error (RMSE). We compare STDN with widely used time series regression models, including (1) Historical average (HA) (2) Autoregressive integrated moving average (ARIMA); The following traditional regression methods are included: (3) Ridge Regression (Ridge); (4) Lasso Regression (Lasso); (5) XGBoost (Chen and Guestrin 2016). In addition, neural-network-based methods are also considered: (6) MultiLayer Perceptron (MLP); (7) Convolutional LSTM (ConvLSTM) (Shi et al. 2015); (8) DeepSD (Wang et al. 2017); (9) Deep Spatial-Temporal Residual Networks (ST-ResNet) (Zhang, Zheng, and Qi 2017); (10) Deep Multi-View Spatial-Temporal Network (DMVST-Net) (Yao et al. 2018).

**Hyperparameter Settings** We set the hyperparameters based on the performance on validation set. For spatial information, we set all convolution kernel sizes to \( 3 \times 3 \) with 64 filters. The size of each neighborhood considered was set as \( 7 \times 7 \). We set \( K = 3 \) (number of layers), \( l = 2 \) (the time span of considered flow). For temporal information, we set the length of short-term LSTM as \( 7 \) (i.e., previous 3.5 hours), \( |P| = 3 \) for long-term periodic information (i.e., previous 3 days), \( |Q| = 3 \) for periodically shifted attention mechanism (i.e., half an hour before and after of relative predicted time are considered), the dimension of hidden representation of LSTM is 128. STDN is optimized via Adam (Kingma and Ba 2014). 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. \( \lambda \) is set as 0.5 to balance start and end volume.
Results

Performance Comparison Table 1 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 mean and standard deviation of each baseline. Besides, we also conduct student t-test. Our proposed STDN significantly outperforms all competing baselines by achieving the lowest RMSE and MAPE on both datasets.

Specifically, the traditional time-series prediction methods (HA 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, STDN outperforms 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 spatiotemporal similarity.

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 3, 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 $F_{i,t}$ defined in Eq. (3) and spatial representation $Y_{i,t}$. 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, 3c and Figure 3b, 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.

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 $|Q| = 1$ (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 1: Comparison with Different Baselines

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Start</th>
<th></th>
<th>End</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RMSE</td>
<td>MAPE</td>
<td>RMSE</td>
<td>MAPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HA</td>
<td>43.82</td>
<td>23.18%</td>
<td>33.83</td>
<td>21.14%</td>
<td></td>
</tr>
<tr>
<td>ARIMA</td>
<td>36.53</td>
<td>22.21%</td>
<td>27.25</td>
<td>20.91%</td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>28.51</td>
<td>19.94%</td>
<td>24.38</td>
<td>20.07%</td>
<td></td>
</tr>
<tr>
<td>MLP</td>
<td>26.67±0.56</td>
<td>18.43±0.62%</td>
<td>22.08±0.50</td>
<td>18.31±0.83%</td>
<td></td>
</tr>
<tr>
<td>XGBoost</td>
<td>26.07</td>
<td>19.35%</td>
<td>21.72</td>
<td>18.70%</td>
<td></td>
</tr>
<tr>
<td>LinUOTD</td>
<td>28.48</td>
<td>19.91%</td>
<td>24.39</td>
<td>20.03%</td>
<td></td>
</tr>
<tr>
<td>ConvLSTM</td>
<td>28.13±0.25</td>
<td>20.50±0.10%</td>
<td>23.67±0.20</td>
<td>20.70±0.20%</td>
<td></td>
</tr>
<tr>
<td>DeepSD</td>
<td>26.35±0.53</td>
<td>18.12±0.38%</td>
<td>21.95±0.35</td>
<td>18.15±0.62%</td>
<td></td>
</tr>
<tr>
<td>ST-ResNet</td>
<td>26.23±0.33</td>
<td>21.13±0.63%</td>
<td>21.63±0.25</td>
<td>21.09±0.51%</td>
<td></td>
</tr>
<tr>
<td>DMVST-Net</td>
<td>25.74±0.26</td>
<td>17.38±0.46%</td>
<td>20.51±0.46</td>
<td>17.14±0.32%</td>
<td></td>
</tr>
<tr>
<td>STDN</td>
<td>24.10±0.25***</td>
<td>16.30±0.25%***</td>
<td>19.08±0.31***</td>
<td>16.25±0.26%***</td>
<td></td>
</tr>
</tbody>
</table>

NYC-Bike

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Start</th>
<th></th>
<th>End</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RMSE</td>
<td>MAPE</td>
<td>RMSE</td>
<td>MAPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HA</td>
<td>12.49</td>
<td>27.82%</td>
<td>11.93</td>
<td>27.06%</td>
<td></td>
</tr>
<tr>
<td>ARIMA</td>
<td>11.53</td>
<td>26.35%</td>
<td>11.25</td>
<td>25.79%</td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>10.92</td>
<td>25.29%</td>
<td>10.33</td>
<td>24.58%</td>
<td></td>
</tr>
<tr>
<td>MLP</td>
<td>9.83±0.19</td>
<td>23.12±0.47%</td>
<td>9.12±0.24</td>
<td>22.40±0.40%</td>
<td></td>
</tr>
<tr>
<td>XGBoost</td>
<td>9.57</td>
<td>23.52%</td>
<td>8.94</td>
<td>22.54%</td>
<td></td>
</tr>
<tr>
<td>LinUOTD</td>
<td>11.04</td>
<td>25.22%</td>
<td>10.44</td>
<td>24.44%</td>
<td></td>
</tr>
<tr>
<td>ConvLSTM</td>
<td>10.40±0.17</td>
<td>25.10±0.45%</td>
<td>9.22±0.19</td>
<td>23.20±0.47%</td>
<td></td>
</tr>
<tr>
<td>DeepSD</td>
<td>9.69</td>
<td>23.62%</td>
<td>9.08</td>
<td>22.36%</td>
<td></td>
</tr>
<tr>
<td>ST-ResNet</td>
<td>9.80±0.12</td>
<td>25.06±0.36%</td>
<td>8.85±0.13</td>
<td>22.98±0.53%</td>
<td></td>
</tr>
<tr>
<td>DMVST-Net</td>
<td>9.14±0.13</td>
<td>22.20±0.33%</td>
<td>8.50±0.19</td>
<td>21.56±0.49%</td>
<td></td>
</tr>
<tr>
<td>STDN</td>
<td>8.85±0.11***</td>
<td>21.84±0.36%**</td>
<td>8.15±0.15***</td>
<td>20.87±0.39%***</td>
<td></td>
</tr>
</tbody>
</table>

*** (**) means the result is significant according to Student’s T-test at level 0.01 (0.05) compared to DMVST-Net

Figure 4: Evaluation of periodically shifted attention mechanism (PSAM) and its variants.

(a) : RMSE on NYC-Taxi  (b) : RMSE on NYC-Bike
(c) : MAPE on NYC-Taxi  (d) : MAPE on NYC-Bike

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 proposed model outperforms the state-of-the-art methods. In the future, we plan to investigate the proposed model on other spatial-temporal prediction problems. In addition, we plan to explain the model (i.e., explain feature importance of traffic prediction), which 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.

Conclusion and Discussion

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 proposed model outperforms the state-of-the-art methods. In the future, we plan to investigate the proposed model on other spatial-temporal prediction problems. In addition, we plan to explain the model (i.e., explain feature importance of traffic prediction), which is...
important for policy makers. Data and code can be found in https://github.com/tangxianfeng/STDN.

Acknowledgments

The work was supported in part by NSF awards #1544455, #1652525, #1618448, and #1639150. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References


Tong, Y.; Chen, Y.; Zhou, Z.; Chen, L.; Wang, J.; Yang, Q.; and Ye, J. 2017. The simpler the better: A unified approach to predicting original taxi demands on large-scale online platforms. In KDD. ACM.


