Citywide Traffic Flow Prediction Based on Multiple Gated Spatio-temporal Convolutional Neural Networks

CEN CHEN, Hunan University and Infocomm for Research Institute
KENLI LI, Hunan University
SIN G. TEO, Infocomm for Research Institute
XIAOFENG ZOU, Hunan University
KEQIN LI, State University of New York and Hunan University
ZENG ZENG, Infocomm for Research Institute

Traffic flow prediction is crucial for public safety and traffic management, and remains a big challenge because of many complicated factors, e.g., multiple spatio-temporal dependencies, holidays, and weather. Some work leveraged 2D convolutional neural networks (CNNs) and long short-term memory networks (LSTMs) to explore spatial relations and temporal relations, respectively, which outperformed the classical approaches. However, it is hard for these work to model spatio-temporal relations jointly. To tackle this, some studies utilized LSTMs to connect high-level layers of CNNs, but left the spatio-temporal correlations not fully exploited in low-level layers. In this work, we propose novel spatio-temporal CNNs to extract spatio-temporal features simultaneously from low-level to high-level layers, and propose a novel gated scheme to control the spatio-temporal features that should be propagated through the hierarchy of layers. Based on these, we propose an end-to-end framework, multiple gated spatio-temporal CNNs (MGSTC), for citywide traffic flow prediction. MGSTC can explore multiple spatio-temporal dependencies through multiple gated spatio-temporal CNN branches, and combine the spatio-temporal features with external factors dynamically. Extensive experiments on two real traffic datasets demonstrates that MGSTC outperforms other state-of-the-art baselines.

CCS Concepts: • **Computing methodologies → Neural networks**;

Additional Key Words and Phrases: Traffic prediction, CNNs, traffic flow prediction, spatio-temporal analysis

The research was partially funded by the National Key R&D Program of China (Grant No.2018YFB1003401), the National Outstanding Youth Science Program of National Natural Science Foundation of China (Grant No.61625202), the International (Regional) Cooperation and Exchange Program of National Natural Science Foundation of China (Grant No. 61860206011), the National Natural Science Foundation of China (Grant No.61902120), and the Postdoctoral Science Foundation of China (Grant No. 2019M662768, 2019TQ0086).

Authors' addresses: C. Chen, College of Information Science and Engineering, Hunan University, Changsha, China, 410082, Infocomm for Research Institute, Singapore, Singapore; emails: chencen@hnu.edu.cn, chenc@i2r.a-star.edu.sg; K. Li (corresponding author), College of Information Science and Engineering, Hunan University, Changsha, China, 410082; email: lkl@hnu.edu.cn; S. G. Teo, Infocomm for Research Institute, Singapore, Singapore; email: teosg@i2r.a-star.edu.sg; X. Zou, College of Information Science and Engineering, Hunan University, Changsha, China, 410082; email: zouxiaofeng@hnu.edu.cn; K. Li, State University of New York, New York, College of Information Science and Engineering, Hunan University, Changsha, China, 410082; email: lik@newpaltz.edu; Z. Zeng (Corresponding author), Infocomm for Research Institute, Singapore, Singapore; email: zengz@i2r.a-star.edu.sg.

1556-4681/2020/05-ART42 $15.00
https://doi.org/10.1145/3385414

ACM Reference format:
Cen Chen, Kenli Li, Sin G. Teo, Xiaofeng Zou, Keqin Li, and Zeng Zeng. 2020. Citywide Traffic Flow Prediction Based on Multiple Gated Spatio-temporal Convolutional Neural Networks. *ACM Trans. Knowl. Discov. Data* 14, 4, Article 42 (May 2020), 23 pages.
https://doi.org/10.1145/3385414

1 INTRODUCTION

Traffic flow prediction is crucial for traffic management, environmental pollution, and public safety [23], and is a vital part in the domain of intelligent transportation system (ITS) [60]. As more traffic data of vehicles are collected from Global Positioning System (GPS) devices [44], traffic cameras [26], mobile devices [56], and traditional road sensors [32], the problem is being more complex and voluminous. Robust traffic prediction models are demanded to handle complexity of massive traffic data, consider spatio-temporal correlations of traffic information to predict traffic conditions for the near future.

The latest report from United Nations [39] states that, more than 55% of the world's population lives in city areas in 2017. As the urban population grows, this figure is expected to grow to 68% by 2050 [39]. Therefore, many cities, such as New York and Beijing, are facing many different pressures and challenges, among which traffic congestion is one of serious problems, resulting in low car speeds, long traveling, waiting time, and so on. Recently, many researchers attempt to leverage machine learning based methods to forecast traffic flows in cities [51, 56]. In these work, a city is first divided into many regions. The inflow and outflow of a region are the total traffic of crowds that have entered the region and the total traffic of crowds that have left the region, respectively. The future traffic flows of each region then are predicted based on the past traffic flows in the city.

Predicting the traffic flow of every region in a city is very challenging and affected by the following important factors.

Spatial dependencies. The inflow of one region, i.e., r_i, in a city would be affected by the nearby outflows as well as that of distant regions. The nearby regions are the neighbors that are either adjacent or near to r_i and distant regions otherwise. Similarly, the outflow of r_i can affect other regions in the city, and the inflow of r_i would affect its own outflow as well.

Multiple temporal dependencies. The inflow, outflow of r_i would be affected by the intervals of short, middle, and long term. For instance, the traffic congestion of r_i occurring at 6pm will affect the traffic condition of the same region at the following time, i.e., 7 pm. One of the rush hours of workdays, i.e., from Monday to Friday, is normally between 8 am and 9am in cities. It is easily observed that the rush hour patterns repeat among workdays.

External factors. The traffic flows of different regions in a city would be tremendously affected by external factors, such as traffic accidents, weather conditions, holidays, and other special events.

Many studies leveraged traditional machine learning methods for traffic prediction, e.g., k-nearest neighbours (KNN) [12, 48] is used to predict traffic speeds and volumes, and support vector machine (SVM) [22, 47] is used to predict traffic flow. However, all the existing machine learning methods cannot capture spatio-temporal features of traffic network and make traffic prediction on massive traffic data. In recent years, much deep learning-based research has been carried out in both academia and industry, with applications across many domains, e.g., computer vision, speech recognition, text understanding, and natural language processing. Due to its powerful feature learning capabilities, many researchers have successfully applied deep learning techniques to predict future traffic conditions using sequences of historical traffic conditions [2, 42, 54, 56].

Some researchers combine two-dimensional (2D) convolutional neural network (CNN) and long short-term memory network (LSTM) together to capture spatio-temporal features for traffic prediction [15, 52, 54, 56]. However, these methods only explore temporal features with spatial features in the high-level layers, while do not combine temporal features with spatial features in low-level layers together. Hence, the low-level spatio-temporal features cannot be fully explored in these methods.

To target on the limitations, in this work we propose a novel multiple gated spatio-temporal correlation based CNNs framework, termed as "MGSTC," that takes all the three categories of factors which are discussed above into consideration. To our best knowledge, the proposed MGSTC is the first time that pure CNN-based model is explored for solving traffic prediction problems that can capture spatio-temporal dependencies from low-level to high-level across the hierarchy of the whole stacks. An earlier and simpler version of this work was included in a conference proceeding [7]. The differences/advantages of this article compared with the conference version are summarized as follows. (i) We have introduced a novel gated mechanism for spatio-temporal features into the framework. (ii) We have introduced novel spatio-temporal convolutional blocks to extract the spatio-temporal features. Our proposed model based on spatio-temporal convolutional blocks outperform the model based on three-dimensional (3D) CNNs. Furthermore, it has less parameters, requires shorter training and testing time compared with original 3D CNN-based models.

It is worth noting that MGSTC is a general model for other traffic prediction problems, as long as the traffic data can be represented in the form of spatio-temporal data. Our contributions can be summarized as follows.

- We propose novel spatio-temporal correlation based CNNs to learn the spatio-temporal features simultaneously from from low-level to high-level layers for traffic flow prediction.
- Motivated by the gate mechanism utilized in LSTM, we also propose a novel spatio-temporal gated mechanism based on CNNs. This gated scheme allows the networks to control what spatio-temporal features should be propagated through the hierarchy of spatio-temporal CNN layers.
- We design a novel end-to-end framework, termed as MGSTC, based on multiple gated spatio-temporal CNNs for city-wide traffic flow prediction, considering multiple spatio-temporal dependencies and external factors. The MGSTC can combine the output features of the multiple gated spatio-temporal CNN branches, and assign weights to different branches dynamically.
- We evaluate our proposed models on NYC bike and Beijing taxi datasets. Experimental results have demonstrated the superiority of our proposed models over state-of-the-art baselines.

We organize the remaining of our work as follows. In the next section, the related work is discussed. Section 3 describes our problem definition, analyzes the limitations of 2D CNN-based approaches for traffic prediction and illustrates the importance of extracting spatio-temporal features simultaneously. Section 4 presents our proposed framework, MGSTC. Sections 5 shows the experimental results, while Section 6 concludes the work.

2 RELATED WORK

Due to the increasingly urgent traffic problems in many big cities, the citywide traffic flow prediction has been attracting a large amount of research attentions in recently years. The traffic prediction approaches can be mainly classified into the following two lines: traditional statistical methods and artificial neural networks (ANNs)-based methods.

Classical statistical methods usually construct different linear or non-linear models for time series prediction. KNN uses the periodicity of the traffic evolution for short-term traffic speed and volume forecasting [12, 48]. Some researchers also utilize tensor completion approaches for traffic data recovery [49]. SVM model [47] and the extensions [22] are selected due to the distinct advantage to handle larger size of data source than other similar methods [3]. Additionally, other classical time-series traffic predictive models, such as Bayesian networks [35], Markov chain [1], Kalman filter [30], Multiple Kernel Regression [34], Auto-regressive Integrated Moving Average[38, 40, 46], and Spatial Correlated Analysis[31], are introduced to solve traffic prediction problems. These methods can capture correlations of the time sequences of variables. However, spatial-temporal correlation based features cannot be extracted well in the work.

ANNs are widely applied across many areas, e.g., recommendation system [41], computer vision [28, 45, 55], and speech recognition [17], to mention a few. ANNs can capture the non-linear dynamics of spatio-temporal relations [6, 54]. Motivated by the latest significant progress on various tasks with ANNs-based methods, especially deep learning, many approaches based on deep learning have also been proposed to tackle the traffic forecasting problem [2, 8, 10, 15, 32, 54, 56, 57]. Some of the works, such as recurrent neural network (RNN) and the variants of RNN, such as gated recurrent unit (GRU), LSTM network, and the like, have shown promising performance in traffic prediction compared to the above-mentioned classical statistical methods [2, 54]. One of the main reasons is that RNN and the variants can effectively extract the characteristics from temporal dependencies [2, 54]. Recently, some researches extended RNN-based methods to support extracting spatio-temporal correlation features. For example, Jain et al. [24] proposed an approach for combining high-level spatio-temporal graphs and RNNs, which shows improvement over the state-of-the-art on modeling human motion and object interactions. Liang et al. [29] combined spatial and temporal attention mechanism into LSTM for geo-sensory time series forecasting problem, e.g., air quality and water quality prediction. Wang et al. [43] propose a spatio-temporal LSTM (ST-LSTM) unit that can extract and memorize spatial and temporal representations simultaneously and showed its advantages on three video prediction datasets. Zhang et al. [58] introduced an extremely efficient method to mimic multiple deep neural networks without increasing the network parameters by way of multitask learning. Ziat et al. [13] introduced a dynamical spatio-temporal model based on RNN for forecasting time series of spatial processes. Zhang et al. [59] proposed a simple yet effective strategy to perform ensemble learning by parameter sharing and shows its superiority with the application of video classification.

Moreover, some researches first modeled the traffic states into images and then utilized 2D CNNs for traffic prediction [32]. In [56], a residual neural network framework is employed to model the temporal information and collectively forecast the inflow and outflow of crowds of each region in a city. However, these approaches can only capture spatial or temporal dependencies of the traffic data independently. To contend with the limitation and better capture the spatial-temporal correlation, the combinations of RNNs and CNNs are straightforwardly considered [25, 50–52, 56, 57]. The pioneer work is [50], where they extended the fully connected long short-term memory (FC-LSTM) to have convolutional structures. Moreover, Yao et al. [52] apply local CNN, LSTM, and semantic network to capture the spatial, temporal, and correlations among similar regions, respectively. Du et al. [15] propose a hybrid multi-modal deep learning framework based on multiple CNN-GRU algorithms, which can effectively extract local spatial features and long dependency features together with spatio-temporal correlations from the multi-modal traffic data. However, these approaches consider one aspect at one time, either temporal or spatial dependency, while constructing the models. To address the limitations of the methods discussed above, we propose novel multiple gated spatio-temporal 3D CNN (MGSTC) for traffic flow forecasting problems in this work.

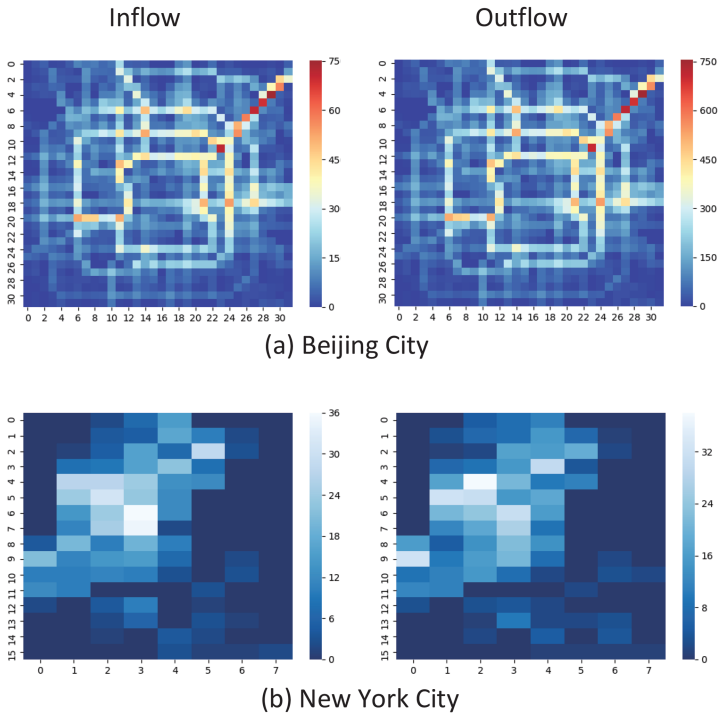

Fig. 1. Visualization of traffic flow maps in Beijing and New York City. A city is partitioned into a grid map. Two values in each grid denote the traffic inflow and outflow, respectively.

3 PROBLEM DEFINITION AND ANALYSIS

3.1 Citywide Traffic Flow Prediction Problem

The definition of citywide traffic flow prediction problem is briefly introduced in this subsection.

Definition 1 (Regions of a City). Following the same idea of the previous studies [14, 32, 51, 56], a city is partitioned into an $I \times J$ grid map where a grid denotes a region of a city. The regions of a city can be defined as pairs (i, j), where denotes that the region is in the i^{th} row and the j^{th} column of the grid map.

Definition 2 (Historical Traffic States). The entire time span T of historical traffic states can be split as non-overlapping time intervals $T = 1, 2, 3, \ldots, t-1$.

Definition 3 (Traffic Inflow and Outflow). Following the previous studies [51, 56], the inflow and outflow of a region are the total traffic of crowds that have entered the region and the total traffic of crowds that have left the region, respectively.

Problem 1 (Citywide Traffic Flow Prediction). The problem of citywide traffic flow prediction is to predict the inflow and outflow of each region of the city at the next time interval t, based on the historical citywide traffic flow data with time intervals $T = 1, 2, 3, \ldots, t-1$.

3.2 Importance of Extracting Spatio-temporal Features Simultaneously

In this section, we demonstrate that the spatial and temporal dependencies are required to be considered simultaneously. For example, as shown in Figure 2, the inflow of one region (i.e., r) of a city at time t is affected by the outflows of its nearby at time t, and also be affected by the outflows

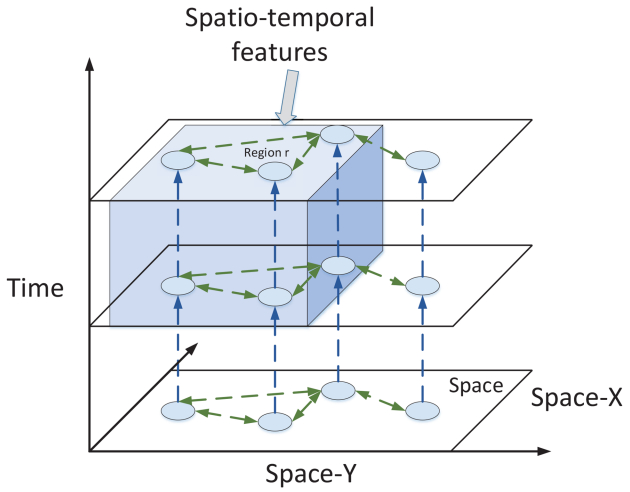

Fig. 2. Necessity of extracting spatio-temporal features simultaneously.

of its distant regions at the time before t. In the meanwhile, the traffic flow of the region r_i at the time $t-1$ will affect the region's traffic flow at the time following. We can observe that the traffic flow of one region is affected by the spatial and temporal factors simultaneously, which inspires us to design a appropriate model that can learn and extract the spatio-temporal features jointly.

3.3 Limitations of Approaches Based on 2D CNN for Traffic Prediction

As discussed above, in order to conduct accurate and robust traffic prediction, spatial and temporal dependencies are required to be considered jointly. However, in 2D CNN, only two dimensions of features could be captured by 2D convolutional operations. Convolutional layers are applied on 2D local neighborhood feature map to extract spatial features with a 2D convolutional kernel. 2D CNN performs well in learning from the features of images that only contains two dimensions (latitude and longitude) [27, 28]. However, fully learning from spatial and temporal features is vitally important to the traffic problem. It is hard for 2D CNN to learn the spatio-temporal features when an additional temporal dimension needs to be considered.

Many researchers have made efforts to learn the spatio-temporal features based on 2D CNN. Generally, these studies could be classified into the following four lines. (1) As shown in Figure 3(a), some flatten two dimensions of the space into a dimension, and then treat one dimension of a 2D image for space and another dimension for time. However, these methods cannot model the actual spatio-temporal dependencies well. That is because, the two-dimensions of spatial dependency are flattened into one dimension, thus losing actual spatial information. (2) The second one is to replace the channels of images with the slices of time and utilize 2D CNN to extract the spatio-temporal features. However, applying 2D convolution on a multiple channel image (multiple frames can be reconstructed into multiple channels) also leads to an image. Hence, temporal information is also lost after every convolutional layer. (3) The third one is to combine 2D CNN with RNN or the variants, e.g., LSTM [50, 52, 53] as illustrated in Figure 3(c). 2D CNN is utilized to capture the spatial dependencies and subsequently LSTMs are utilized to capture the temporal dependencies of the output. However, this type of approaches only builds temporal correlations on the high-level spatial features while leave the correlations in the low-level spatial features to not be fully exploited. (4) The fourth one is to extract spatial features by using 2D CNNs for a slice of 2D images along the time dimension, and then aggregate the outputs together by Tanh function [56]. Similar

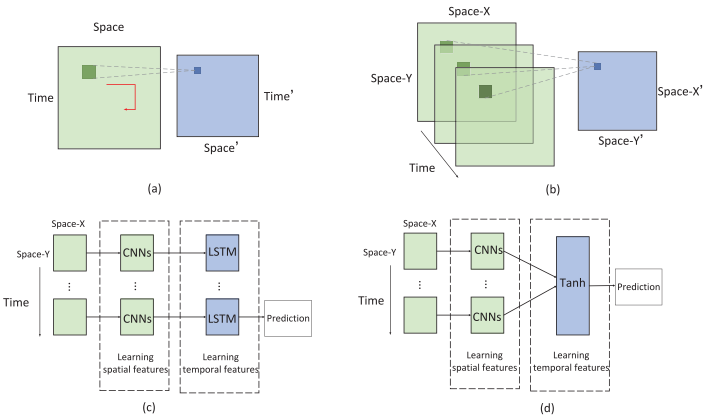

Fig. 3. 2D CNN-based approaches; (a) Flatten the two dimensions of the space into a dimension, and model spatio-temporal dependencies into a 2D image. Then utilize 2D CNN to extract the spatio-temporal features; (b) Model traffic states along the time dimension into channels of images to construct the temporal dependencies; (c) Utilize 2D CNN to capture the spatial dependencies and then feed the outputs to LSTM to capture the temporal dependencies; (d) Utilize 2D CNNs to extract the spatial features for a slice of 2D images along the time dimension, and then aggregate the outputs together by a Tanh function.

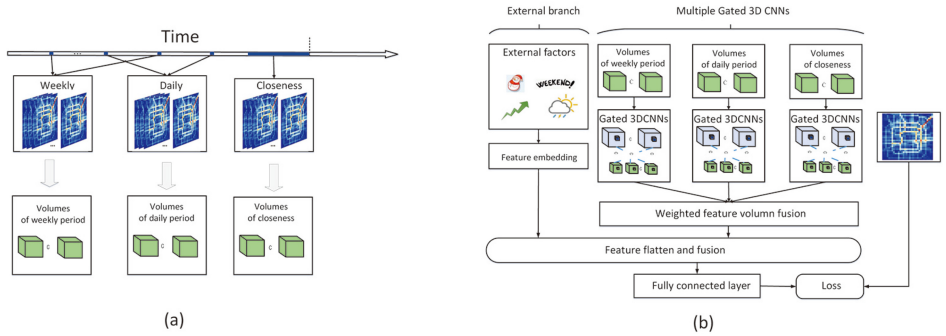

Fig. 4. (a) Modeling multiple spatio-temporal dependencies into multiple spatio-temporal 3D volumes; (b) The proposed architecture of MGSTC.

with the methods in item 3, the temporal dependencies among low-level spatial features cannot be explored.

4 PROPOSED MGSTC FRAMEWORK

4.1 Framework Overview

Utilizing MGSTC consists of three steps where each step is described in the following: (1) In the first step, the citywide traffic flow data is first converted into multiple 3D volumes with spatial and temporal information; (2) In the second step, a training dataset of 3D volumes is used to train a model using our proposed framework MGSTC; (3) In the third step, the trained model is utilized to predict traffic flow in citywide areas.

The process of modeling traffic flow data with 3D volumes is depicted in Figure 4(a) and the framework of MGSTC is depicted in Figure 4(b). There are two components in MGSTC, multiple gated 3D CNN branches and an external branch. A branch of our proposed gated 3D CNN stack

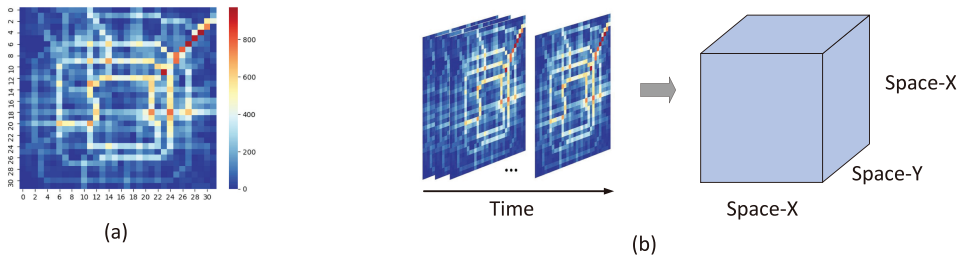

Fig. 5. Modeling spatio-temporal correlation dependency. (a) A multi-channel image that denotes the inflow/outflow of a city. (b) A multi-channel 3D volume that denotes a slice of multi-channel images along the time dimension.

can learn the spatial dependency and temporal dependency together. In this work, *closeness* dependency and *periodic*, e.g., *daily* and *weekly*, dependencies are considered. By utilizing multiple gated 3D CNN branches, MGSTC extracts multiple spatio-temporal correlation based features from traffic data. Then a weighted fusion method is utilized to fuse the extracted multiple spatio-temporal correlation features. For external factors, we extract some features manually from external datasets that contain holidays, weather conditions, and the like. Then, the external features are embedded by a two-layer fully-connected neural networks. Then, the extracted external features are fused together with the multiple spatio-temporal features. Lastly, a fully-connected neural network is applied to calculate the cross-entropy loss in traffic flow prediction.

We describe the way of data modeling, our proposed gated ST CNN for extracting saptio-temporal features, the way of data modeling, the architecture of MGSTC and the training process in the following.

4.2 Modeling Process

In this section, we present the way of modeling multiple spatio-temporal dependencies of city-wide traffic flow into multiple spatio-temporal correlation-based 3D volumes. In the following, we first illustrate the process of modeling citywide traffic flow with spatio-temporal correlations based on 3D data volumes. Lastly, we present the process of how to model multiple temporal dependencies.

4.2.1 Modeling Spatio-Temporal Correlations with 3D Data Volumes. In time interval t, the traffic situation of a city can be denoted by a tensor $X_t \in R^{i \times j \times k}$, where k is the number of traffic variables and the city is partitioned into a $i \times j$ grid map. As shown in Figure 5(a), the tensor can be considered as an image with k channels, i pixels height and j pixels width. This constructed multi-channel image can preserve the spatial dependency of citywide traffic states. Specifically, by setting $k = 2$ and filling the inflow/outflow values into these 2 channels, the 2 channel image can capture the spatial correlations of citywide traffic flow states. Given h time intervals, the traffic flow values of these time intervals can be denoted as a tensor $V \in R^{h \times i \times j \times 2}$. As shown in Figure 5(b), the generated tensor can be considered as a 3D data volume with the size of $h \times i \times j \times 2$, where h is the number of images. The reconstructed 3D volume can present the spatio-temporal information of citywide traffic flow situations.

4.2.2 Multiple Temporal Dependencies. Despite the future traffic states are affected by the historical traffic states in the recent time, they are also affected by periodic temporal dependencies. Figures 6(a) and 7(b) describe the inflow value at each time intervals in 7 days. From these figures,

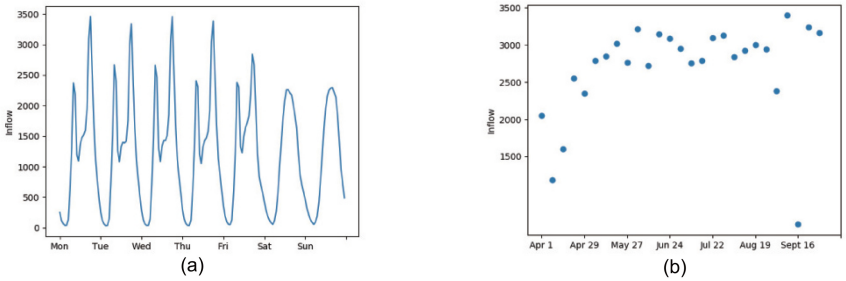

Fig. 6. Multiple temporal dependencies for NYCBike. (a) daily; (b) weekly.

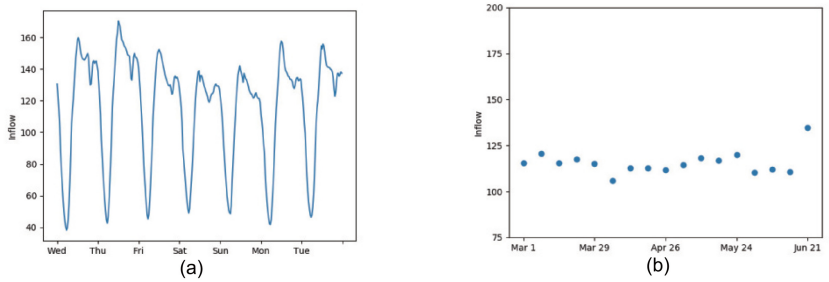

Fig. 7. Multiple temporal dependencies for BJTaxi. (a) daily; (b) weekly.

we can find that the traffic flow data show a certain repeatability pattern. Therefore, we can identify the daily periodic patterns clearly on the two datasets. Moreover, Figures 6(a) and 7(b) are plotted to present the weekly periodicity of the traffic data.

From the cases illustrated above, it is clear that daily periodicity and weekly periodicity have significant impacts on the traffic states, though the degrees of influences are not completely the same. Obviously, the traffic states in the recent time have significant impacts on the traffic states in the following time. Inspired by the observations, we can construct multiple 3D volumes separately for the multiple temporal properties, based on the images with spatial information. The steps of the modeling the multiple temporal properties are presented in Figure 4(a). For the *closeness* 3D volumes, a few 2-channel images of the recent time intervals are used to model the temporal *closeness* dependency. Let the traffic flow states of a recent fragment be $[X_{t-l_c}, X_{t-(l_c-1)}, \ldots, X_{t-1}]$. These frames can be modeled with a 3D volume, $V_c \in R^{l_c \times i \times j \times 2}$.

In the same way, we can reconstruct the *periods* volumes. We take the *daily* period as an example. Suppose that l_d is the time interval from the period, and d is the period span. The *daily* period of a dependent sequence is $[X_{t-l_d \times d}, X_{t-(l_d-1)\times d}, \ldots, X_{t-1}]$. This sequence can be reconstructed into a 3D volume $V_d \in R^{l_d \times i \times j \times 2}$. We only use *daily* and *weekly* periods in our implementation. Other kinds of *periods*, e.g., monthly and seasonal, can be supported in the same way.

4.3 Our Proposed Gated Spatio-Temporal CNN

4.3.1 Exploring Spatio-Temporal Features with 3D CNNs. Unlike 2D CNN, 3D CNN has the ability to capture the features with three dimensions by applying the 3D convolutional operations and other 3D activation functions. As shown in Figure 8(a), if the traffic data is represented into spatio-temporal correlation-based 3D volumes, 3D CNN can extract the spatio-temporal correlation features through 3D convolution operations [33, 37]. Due to the construction of 3D CNN, if we apply 3D convolutions with a 3D kernel, the 3D kernel can sweep over the entire 3D topology. Moreover,

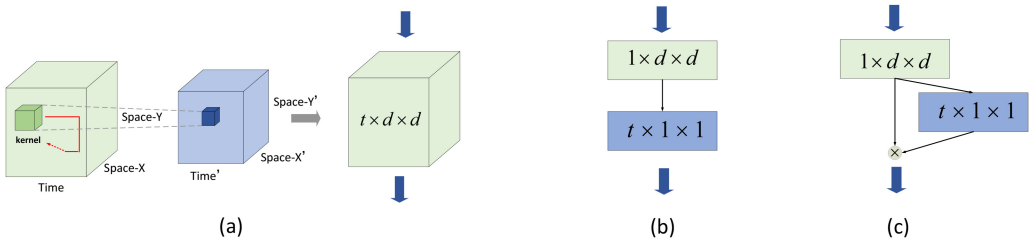

Fig. 8. Our proposed spatio-temporal convolutional blocks. (a) Extract spatio-temporal features by the original 3D CNN. (b) Our proposed spatio-temporal convolutional block without residual fusion. (c) Our proposed spatio-temporal convolutional block with residual fusion.

through adopting the same kernel sharing across space and time dimensions (highlighted in dark blue), the 3D CNN model can take full advantage of spatio-temporal dependencies and potential hidden correlations of the traffic flow data. We present the 3D convolutional operation as follows:

$$u_{ij}^{\beta}(x,y,z) = \sum_{m,n,l} V_i^{\beta-1}(x-m, y-n, z-l) W_{ij}^{\beta}(m,n,l), \tag{1}$$

where W_{ij}^{β} is the 3D kernel in the β^{th} layer convoluting over the 3D feature volume $V_i^{\beta-1}$, and $W_{ij}^{\beta}(m,n,l)$ is the element-wise weight in the 3D convolution kernel. Then, the formulation of 3D feature volume in β^{th} layer can be represented as follows:

$$V_j^{\beta} = f\left(\sum_i u_{ij}^{\beta}\right), \tag{2}$$

where f is the activation function.

4.3.2 Spatio-Temporal Convolutioal Blocks. As shown in Figure 8(a), a original 3D convolution is carried out using a filter $t \times d \times d$ where t denotes the temporal extent and d is the spatial width and height. Original 3D convolutions simultaneously model the spatial information like 2D filters and construct temporal connections across frames. To reduce the model size, we propose spatio-temporal convolutional blocks as shown in Figure 8(b). Suppose we have 3D convolutional filters with size of $t \times d \times d$, it can be naturally decoupled into $1 \times d \times d$ convolutional filters equivalent to 2D CNN on spatial domain, and $t \times 1 \times 1$ convolutional filters like 1D CNN equivalent to temporal domain in a cascaded manner. The 2D CNN for spatial domain and the 1D CNN for temporal domain can be grouped as a spatio-temporal convolutional block to extract the spatio-temporal features. These two kinds of filters can influence each other in the same path and only the temporal 1D filters are connected to the final output of the block directly.

To make the output of spatial and temporal convolutions influence the final output simultaneously, we adopt the residual fusion between the output of spatial convolution and the final output of the block. This mechanism is also motivated by the residual learning utilized in CNNs for image recognition [19, 20]. The spatio-temporal block with residual connections is shown in Figure 8(c).

4.3.3 Spatio-Temporal Gated Mechanism. RNN-based methods are becoming popular in sequential time series analysis. Gating mechanisms can control the path through which information flows in the network and have proven to be useful for RNN [21]. Some researchers have designed Gated Linear Units (GLUs) based on 2D CNNs for language processing and achieved better performance compared with RNN-based models (e.g., LSTM) [11, 16]. GLU allow the networks to focus on fewer elements if needed, which is similar with the gates utilized in LSTMs.

Inspired by [11, 16], we design spatio-temporal GLU based on 3D CNN or spatio-temporal CNN as shown in Figure 8(b). In the scheme, supposing the input is X, a simple gated mechanism based on 3D convolution or spatio-temporal CNN is illustrated as following:

$$\begin{aligned} \Theta([A, B]) &= A \otimes \sigma(B) \\ A &= ST(X) \\ B &= ST(X), \end{aligned} \tag{3}$$

where A and B are the inputs to the non-linearity, $\otimes$ is the point-wise multiplication, σ is the activation function, and ST denotes the 3D convolution or spatio-temporal convolution without activation function.

The gate $\sigma(B)$ controls whether the inputs A are relevant to the current spatio-temporal features or not. For the activation function $\otimes$, we utilize $tanh$ as introduced in [11].

4.3.4 Hierarchical Learning. By stacking and 3D convolution layers, spatio-temporal convolutional blocks and 3D pooling layers, the spatio-temporal features can be extracted hierarchically from low-level layers to high-level layers. In the pooling layers, the produced feature volumes can be sub-sampled with max-pooling operation, reducing variance and computational complexity, and extracting low-level features from cubic neighborhoods [33, 37]. In the hierarchical CNN layers, different spatio-temporal kernels are employed with the following non-linear activation functions.

4.4 Multiple Gated Spatio-temporal CNNs

After modeling, the 3D volumes can be fed into the proposed MGSTC as illustrated in Figure 4(b). Each branch takes one type of 3D volumes and targets on exploring this type of spatio-temporal dependency. For instance, the *closeness* branch takes the 3D volumes of *closeness* as inputs, and the *daily* branch takes 3D volumes of *daily* as inputs, respectively.

As discussed in Section 4.2, all the regions in the city involves multiple temporal dependencies, though the degrees of influence of different temporal dependencies may be different. To tackle this, we propose a novel parametric-tensor-based fusion method that combine features extracted by multiple branches and Equation (4) shows the fusion process:

$$V_{fusion} = W_c \otimes V_c + W_d \otimes V_d + W_w \otimes V_w, \tag{4}$$

where V_c, V_d, V_w are the spatio-temporal feature volume extracted by the branches of *closeness*, *daily*, and *weekly*, respectively; $\otimes$ is Hadamard product; W_c, W_d, W_w are the learnable parameters that can adjust the weights of the branches; and V_{fusion} is the volume of fused spatio-temporal features.

Then, V_{fusion} is flattened into a vector, termed as V_{mc}, the final feature extracted by multiple gated 3D CNNs branches.

4.5 External Branch

The external factors, eg., weather conditions, holidays, and special events, can affect the citywide traffic states significantly. For instance, in holidays like Chinese New Year and Christmas, the traffic flow in some regions are heavier compared to non-holidays, while in some regions, the traffic situations are opposite. Another example is that rainy weather can cause slow down the traffic speed due to slippery roads.

In this work, we mainly consider the weather conditions, holiday events, and other metadata, such as the day of week, workday, and weekend. Two fully-connected layers are conducted on E_t to embed the external factors into V_{ext}. We then concatenate the features V_{mc} which are extracted

by the multiple gated spatio-temporal CNNs branches with the external features V_{ext}. Finally, a fully-connected layer is followed, the output of which is the predicted traffic flows of the city.

4.6 Training Process of MGSTC

Algorithm 1 illustrates the training process of MGSTC. In time interval t, the 2-channel image which denotes the inflow/outflow values of the city is regarded as the ground-truth. The 3D volumes constructed by the historical traffic states are utilized as the inputs of MGSTC. We adopt *closeness* branch, *daily,* and *weekly* periods in our implementation. More periods can be considered in a similar way. All the trainable parameters in the proposed MGSTC are initialized randomly and then optimized by the back propagation. In this work, we adopts stochastic gradient descent (SGD) [4] to minimize the cross entropy loss function of MGSTC.

ALGORITHM 1: MGSTC Algorithm

Require: Historical observations: $X_{0,\ldots,n-1}$;
External factors: $E_{1,\ldots,n-1}$;
Lengths of *closeness*, *daily* and *weekly*: l_c, l_d, l_w;
Daily span: d, Weekly span: w.
Ensure: Learned MGSTC model.

$D \leftarrow \emptyset$;
//*Modeling traffic values into multiple temporal and spatial volumes*;
for all available time interval $t (1 \leq t \leq n-1)$ **do**
  $V_c \leftarrow [X_{t-l_c}, X_{t-(l_c-1)}, \ldots, X_{t-1}]$;
  $V_d \leftarrow [X_{t-l_d \times d}, X_{t-(l_d-1)\times d}, \ldots, X_{t-1}]$;
  $V_w \leftarrow [X_{t-l_w \times w}, X_{t-(l_w-1)\times w}, \ldots, X_{t-1}]$;
  put a training instance (V_c, V_d, V_w, E_t) into D;
end for
//*Training the model*;
Initialize all learnable parameters θ in MGSTC;
while stopping criteria is not met **do**
  Select a batch of instances D_b from D randomly;
  Feed each V_c, V_d, V_w, E_t of an instance in D_b into the corresponding branch respectively;
  Find θ by minimizing the objective with D_b;
end while
return

5 EXPERIMENTS

In this section, the performance of our proposed models is compared with other existing state-of-the-art methods.

5.1 Experiment Settings

We implement our methods on Tensorflow (1.2.1) and Keras (2.1.6) on 4 NVIDIA P100 GPUs. Two real-world traffic flow datasets, BeiJing and New York City datasets, are utilized in the experiments and illustrated in detail as follows and summarized in Table 1.

—**NYCBike:** This dataset is bike trajectory from NYC from 1st April to 30th September in 2014. In our experiment, the testing data are the last 10 days of the trajectory data, while others are treated as the training data. Moreover, holidays are provided and regarded as the external information.

Table 1. Statistics of the Datasets

Dataset	TaxiBJ	BikeNYC
Location	Beijing	New York
Data type	Taxi GPS	Bike rent
Sampling time-interval	30 minutes	1 hour
Gird map size	(32,32)	(16,8)
Available images	21,862	4,392
Train images	15,142	3,480
Test images	1,344	240

—**BJTaxi:** This is the taxi trajectory data collected from Beijing. In our experiments, the testing data contains traffic values of the last 4 weeks, while other traffic values are treated as the training data. Temperature, weather conditions and holidays are also provided in the dataset as external information.

5.2 Implementation Details

Data Preprocessing. For the NYCBike dataset, the entire city is split into a 8×16 grid map and the time span is set to 1 hour. For the BJTaxi dataset, the entire city is split into a 32×32 grid map and the time span is set to 30 minutes. Using Definition 3, we can get two types of traffic flows of NYCBike and BJTaxi. Day-of-week, weekend/weekday, weather conditions, holidays, and the like, are transformed into binary vectors and fed into the framework as external factors. Min–Max normalization is applied to convert original traffic values in $[0, 1]$ scale, while the prediction values are de-normalized for evaluations. Similarly, we also apply Min–Max normalization to the existing methods before compared them with our proposed MGSTC. The method of transformation we used is similar to the method proposed in [56].

Parameters. The number of time intervals of *closeness*, *daily,* and *weekly* on NYCBike dataset are set to 4, 4, and 4, respectively. The number of time intervals of *closeness*, *daily,* and *weekly* on BJTaxi dataset are set to 6, 4, and 4, respectively, as the the time segment in BJTaxi is set to half an hour. For all models, the learning rate is set to 0.0002 and the Adam optimizer is applied to optimize the model. We train proposed models for a maximum of 200 epochs (training iterations) and adopt an early stop strategy, i.e., we stop training if the validation loss does not decrease for 15 consecutive epochs. Additionally, we apply *relu* as the activation function for all layers. Batch normalization is used and the batch size is set to 64 in the experiment.

While designing the structure of the neural networks, we need to consider the following two important factors: (i) hyper parameters of convolutional layer and pooling layer (e.g., convolutional filter size and polling size); and (ii) depth of the networks. Owing to the size of our generated 3D volumes of NYCBike dataset is small (i.e., $4 \times 8 \times 16$ in all the branches), only two 3D convolutional layers are applied in all the branches for NYCBike dataset. The kernel sizes in all the branches are set to $(2, 2, 3)$, where the first 2 denotes the temporal kernel size and $(2, 3)$ denotes the spatial kernel size. The number of 3D convolutional filters of the first layer is 32, and that of the second layer is 64. Then a spatial-temporal gate with the kernel size $(2, 2, 3)$ is applied to control the flow of spatio-temporal features. A pooling layer with the size of $(1, 2, 2)$ is followed, and an extra dropout layer is set to 0.25 to avoid the over-fitting issue.

For the BJTaxi dataset, the grid map is 32×32 which are larger than that of the NYCBike dataset. We apply three 3D convolutional layers in all the multiple branches. The parameters of MST3D and MGST3D are described as follows. In the *closeness* branch, the kernel size is set to $(2, 2, 3)$ for these

three layers. In the *daily* and *weekly* branches, the kernel size of the first layer is $(2, 2, 3)$, while the other two layers are set to $(1, 2, 3)$. The above settings can align the output sizes of the *closeness*, *daily*, and *weekly* branches. The number of 3D convolutional filters of three 3D convolutional layers are set to 32, 64, and 64, respectively. Then a spatial-temporal gate is also followed with these three layers to control the flow of spatio-temporal features with the kernel $(2, 2, 3)$. The parameters of the max pooling layer and the dropout layer are the same as those set for the NYCTaxi dataset.

The parameters of MSTC and MGSTC are described as follows. For fair comparison, we replace the 3D CNNs in MST3D and MGST3D with our proposed spatio-temporal CNN blocks with the same filters. For example, for the 3D convolutional operation with kernel $(2, 2, 3)$, we decouple the operation into the 2D CNN on spatial domain with kernel $(1, 2, 3)$ and 1D CNN on temporal domain with the kernel $(2, 1, 1)$. The residual fusion mechanism is adopted in the spatio-temporal CNN blocks.

Metrics for Evaluation. Mean Average Percentage Error (MAPE) and Rooted Mean Square Error (RMSE) are utilized as the evaluation metrics [14, 51, 52, 56]. Samples with values less than 10 are ignored to calculate the MAPE result, which is a common practice used in traffic prediction [51, 52]. The two evaluation metrics are defined in Equations (5) and (6).

$$MAPE = \frac{1}{N}\sum_{\alpha=1}^{N}\frac{|\hat{y_t} - y_t|}{y_t}, \tag{5}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{\alpha=1}^{N}(\hat{y_t} - y_t)}, \tag{6}$$

where y_t and $\hat{y_t}$ denote the real value and the prediction value on time interval t, respectively, and N is the number of all samples.

5.3 Methods Under Comparisons

The compared methods are listed as follows.

Classical Time-Series Prediction Approaches.

- **HA:** Historical average (HA) predicts the traffic values based on the average values of the previous time intervals.
- **ARIMA:** Auto-regressive integrated moving average (ARIMA) is a widely used approach for time series analysis [5].

Classical Statistical Prediction Approaches.

- **LinUOTD [36]:** A linear regression method with a spatio-temporal regularization.
- **XGBoost [9]:** A well-known boosting tree method.

Deep Learning Methods.

- **Multilayer Perceptron (MLP):** We compare our proposed MGSTC with a neural network [18] which contains four fully connected layers.
- **ConvLSTM [50]:** ConvLSTM adds convolutional layers to LSTM.
- **ST-ResNet [57]:** ST-ResNet models citywide traffic flow at different times into 2D images. 2D CNNs are utilized to extract spatial features for closeness, period and trend information. Then a Tanh function is utilized to aggregate the relevant features together. We set the length trend, period and closeness as 4, 4, and 4, respectively.

Table 2. Baseline Comparisons on NYCBike and BJTaxi

Method	NYCBike		BJTaxi	
	RMSE	MAPE	RMSE	MAPE
HA	14.35	39.22%	57.69	38.34%
ARIMA	10.07	29.23%	22.78	22.13%
LinUOTD	9.76	28.12%	21.23	20.22%
XGBoost	6.93	23.12%	17.84	17.62%
MLP	7.68	24.93%	18.25	17.83%
ConvLSTM	7.98	25.62%	19.54	18.63%
ST-ResNet	6.33	21.81%	16.89	15.48%
STDN	6.20	21.57%	16.65	15.27%
MST3D	5.81	20.68%	15.99	14.78%
MSTC	5.79	20.67%	15.82	14.67%
MGST3D	5.76	20.55%	15.86	14.65%
MGSTC	**5.75**	**20.55%**	**15.81**	**14.63%**

—**STDN [51]:** STDN combines local 2D CNNs, LSTM and attention mechanism together for traffic flow prediction. The network of STDN is an extension of [52]. Therefore, we just utilizes STDN as a comparison. We modify some codes of STDN to make STDN predict the inflow/outflow of the city maintaining the network structure of STDN.

The variants of our proposed MGSTC used in this experiment are listed as follows. These variants all contains *closeness*, *daily*, *weekly*, and *external* branches

—**MST3D:** Our proposed framework which utilize 3D CNN and does not adopt the spatio-temporal gated mechanism.
—**MSTC:** Our proposed framework which utilizes our proposed spatio-temporal CNN blocks with residual fusion and does not adopt the spatio-temporal gated mechanism.
—**MGST3D:** Our proposed framework which utilize 3D CNN and adopts the spatio-temporal gated mechanism.
—**MGSTC:** Our proposed framework which utilizes our proposed spatio-temporal CNN blocks and adopts the spatio-temporal gated mechanism.

5.4 Overview of Performance Evaluations

Table 2 presents the RMSE and MAPE results of our proposed models as compared to existing methods for the inflows and outflows together on the NYCBike and BJTaxi datasets. Tables 3 and 4 present the detailed results of inflows and outflows separatedly. We run all the methods 10 times and record the average results of each method.

We can observe that even the MST3D which does not adopt the spatio-temporal gated mechanism outperforms other compared baselines regarding to RMSE and MAPE on both datasets, i.e., the RMSE and the MAPE of our MST3D in the NYCBike dataset are 5.81 and 20.68%, respectively, the RMSE and the MAPE of our MST3D in the BJTaxi dataset are 15.98 and 14.78%, respectively. By adopting the spatio-temporal gated mechanism, the RMSE and MAPE are further decreased, i.e., the RMSE and the MAPE of our MGST3D in the NYCBike dataset are 5.76 and 20.55%, respectively, the RMSE and the MAPE of our MGST3D in the BJTaxi dataset are 15.88 and 14.65%, respectively. From Tables 3 and 4, we can observe that MST3D can outperform other compared baselines for the predictions of both inflow and outflow. The results demonstrate the effectiveness of our schemes of

Table 3. Inflow and Outflow Results on NYCBike

Methods	Inflow		Outflow	
	RMSE	MAPE	RMSE	MAPE
HA	14.02	38.75%	14.91	39.68%
ARIMA	9.97	28.97%	10.49	29.49%
LinUOTD	9.56	27.59%	10.11	28.74%
XGBoost	6.89	22.93%	7.06	23.43%
MLP	7.25	24.63%	8.07	25.29%
ConvLSTM	7.74	25.57%	8.32	25.72%
ST-ResNet	6.08	21.23%	6.63	22.17%
STDN	5.98	21.01%	6.51	21.96%
MST3D	5.66	20.21%	5.96	21.14%
MSTC	5.61	20.19%	5.96	21.13%
MGST3D	5.59	**20.13%**	5.92	20.99%
MGSTC	**5.57**	20.14%	**5.91**	**20.98%**

Table 4. Inflow and Outflow Results on BJTaxi

Methods	Inflow		Outflow	
	RMSE	MAPE	RMSE	MAPE
HA	57.57	37.76%	57.89	39.68%
ARIMA	22.58	22.12%	22.96	22.19%
LinUOTD	21.19	20.02%	21.44	20.33%
XGBoost	17.61	17.42%	18.23	17.69%
MLP	18.23	17.54%	18.30	18.21%
ConvLSTM	19.29	18.55%	19.98	18.72%
ST-ResNet	16.74	15.01%	17.01	15.78%
STDN	16.43	15.12%	16.78	15.44%
MST3D	15.98	14.71%	16.11	14.85%
MSTC	15.77	14.60%	15.98	14.84%
MGST3D	15.82	14.61%	15.97	14.76%
MGSTC	**15.79**	**14.60%**	**15.96**	**14.75%**

adopting multiple gated 3D CNNs to capture the spatio-temporal features and extracting external features. Note that, MGST3D outperforms MST3D.

Furthermore, we also can observe that our proposed MSTC and MGSTC outperform other compared baselines regarding to RMSE and MAPE on both datasets, which demonstrates the effectiveness of utilizing our proposed spatio-temporal convolutional blocks to extract the spatio-temporal features for traffic flow datasets. MSTC and MGSTC also achieve better results than MGST3D an MST3D, respectively. Similar to MST3D and MGST3D, MGST3D performs better than MST3D, which demonstrates the effectiveness of adopting the gated spatio-temporal mechanism.

In contrast, the traditional time-series prediction methods (i.e., ARIMA and HA) cannot achieve good results as they only utilize historical values to predict the future values and do not explore spatio-temporal dependencies and other external factors. The regression-based methods can achieve better performance than other traditional time-series prediction approaches as they explore spatial correlations. However, they still failed to capture the spatial dependency and the

Table 5. Effects of the Residual Fusion

Method	NYCBike		BJTaxi	
	RMSE	MAPE	RMSE	MAPE
MST3D	5.81	20.68%	15.99	14.78%
MSTC-NRes	5.81	20.72%	15.91	14.75%
MSTC	5.79	20.67%	15.82	14.67%

complex non-linear temporal dependency. Our proposed models significantly outperforms the above methods.

Our proposed models also outperforms MLP and ST-ResNet. One of the possible reasons is that MLP cannot explicitly model spatial and temporal dependencies. Moreover, ST-ResNet only uses 2D CNNs to learn low-level spatial features and then take advantages of Tanh function to learn the temporal dependency on the extracted spatial features, which leaves the low-level spatio-temporal dependencies not fully exploited and also ignores the temporal sequential dependency. MGSTC also outperforms ConvLSTM and STDN which combine 2D CNNs and LSTM together for traffic prediction. Similarly with ST-ResNet, ConvLSTM and STDN also overlooks the temporal correlations with low-level spatial features.

5.5 Effects of the Residual Fusion

To make the output of spatial and temporal convolutions influence the final output simultaneously, we adopt the residual connections between the output of spatial convolution and the final output of the spatio-temporal convolutional block. In this section, we evaluate the effectiveness of residual fusion of the outputs of spatial and temporal convolutions. The experimental results are presented in Table 5. MSTC-NRes means our proposed MSTC model without residual fusion. We can observe that MST3D and MSTC-NRes obtain similar results. After adopting the residual fusion mechanism, the performance on both datasets is improved.

5.6 Effects of Multiple Branches and External Factors

To evaluate the effects of adopting multiple gated CNN branches and the effectiveness of considering external factors, we apply different branches to examine the perform of the following variants.

- **MGS3D-C:** This variant only utilizes *closeness* branch on original 3D CNNs.
- **MGS3D-CD:** This variant utilizes both *closeness* and *daily* branches on original 3D CNNs.
- **MGS3D-CDW:** This variant utilizes *closeness*, *daily,* and *weekly* branches on original 3D CNNs.
- **MGSTC-C:** This variant only utilizes *closeness* branch on spatio-temporal convolutional blocks with residual fusion.
- **MGSTC-CD:** This variant utilizes both *closeness* and *daily* branches on spatio-temporal convolutional blocks with residual fusion.
- **MGSTC-CDW:** This variant utilizes *closeness*, *daily,* and *weekly* branches on spatio-temporal convolutional blocks with residual fusion.

Figure9 shows the results of MGST3D, MGSTC, and its variants We can clearly observe that MGSTC-C which only uses the *closeness* branch outperforms the other baselines. It demonstrates the advantages of adopting 3D CNNs and spatio-temporal convolutional blocks to learn spatio-temporal features in traffic prediction. The RMSE and MAPE are further decreased by adding the

Table 6. Time Complexity of Different Approaches

Methods	NYCBike		BJTaxi	
	Training time (s)	Testing time (s)	Training time (s)	Testing time (s)
ST-ResNet	150	0.75	4,994	1.92
STDN	18,980	89.8	379,600	207.4
MST3D	126	0.23	5,902	2.74
MSTC	134	0.16	3,610	1.22
MGST3D	279	0.32	6,749	3.94
MGSTC	169	0.19	3,840	1.45

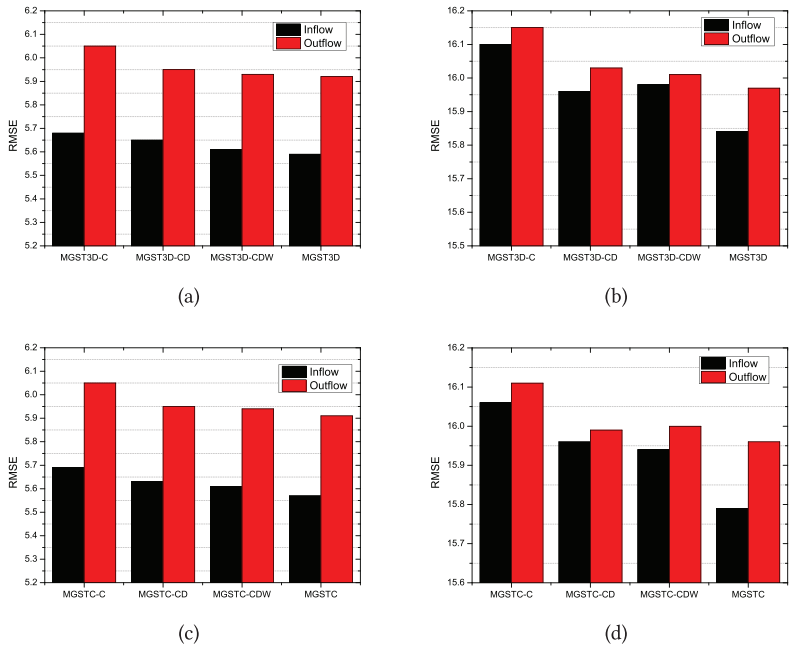

Fig. 9. Results with MGST3D and MGSTC on RMSE. (a) MGST3D on NYCBike. (b) MGST3D on BJTaxi. (c) MGSTC on NYCBike. (d) MGSTC on BJTaxi.

daily and *weekly* branches. It proves that exploring periodic dependencies can improve the prediction performance. An interesting observation is that, for BJTaxi dataset, the RMSE of MGSTC-CD is a bit bigger than that of MGSTC-C. It illustrates that adding *daily* branch has a little effect on BJTaxi dataset. The performance is further improved by adding the *external* branch. It demonstrates that external factors can significantly affect the performance of our method. Lastly, we can see that the method that combines the *closeness*, *daily*, *weekly*, and the *external* branches together can achieve the lowest RMSE.

5.7 Time Complexity of Different Approaches

In this section, we study the time complexity of our proposed MGST3D and MGSTC compared with other baselines. Table 6 shows the running time of MGST3D, MGSTC, and other baselines in the training and prediction procedures. For simplicity, only one P100 GPU is utilized for all

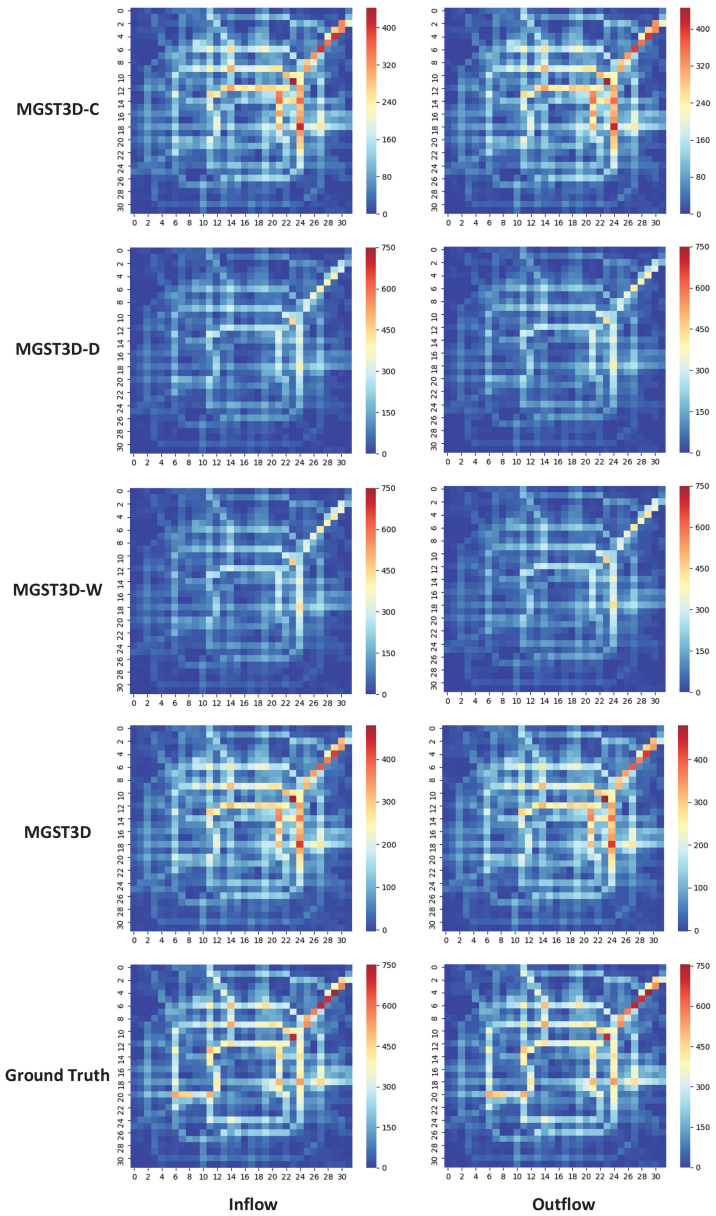

Fig. 10. Visual comparison of predicted results of different variants on TaxiBJ dataset. The top four rows are the predicted results of MGST3D-C, MGST3D-D, MGST3D-W, and MGST3D. The last row is the ground truth map. The left column is inflow maps and the right column is outflow map.

the approaches. We only compare our MGSTC with the existing methods which have achieved relatively good results. We can observe that the running time of STDN are the longest in both training and testing procedures. That is because, STDN utilizes local CNNs to only predict the center of the value, and it uses a sliding window across the whole city to train and then predict every region. For example, it needs to repeats 8×16 times to predict the traffic values of the whole city for NYCTaxi dataset.

For NYCTaxi dataset, the MST3D performs slightly better than ST-ResNet. One of the main reasons is that MST3D uses two 3D CNN layers, while ST-ResNet only uses two convolutional layers and 12 residual units. For BJTaxi dataset, the testing time of MST3D that uses three 3D CNN layers is slightly longer than that of the 2D ST-ResNet. Obviously, the 3D CNN layers are more complex than the 2D CNN layers in neural networks. However, the training time of the MST3D is similar to that of the ST-ResNet. It demonstrates the speed of convergence of our proposed MST3D is faster than that of ST-ResNet. After adopting the spatio-temporal gated mechanism, the training time and testing time is increased.

We also can observe that the running time and testing time of MGSTC are shorter than that of MGST3D. It shows that replacing original 3D CNNs with our proposed spatio-temporal convolutional blocks can achieve better results with shorter running time and testing time.

5.8 Analysis of Spatio-temporal Convolutional Blocks and Original 3D CNNs

We first discuss the parameters of spatio-temporal convolutional blocks compared with 3D CNN layers. Given a 3D CNN layer with 3D convolutional filter which has the size of $t \times d \times d$, the parameter of this 3D convolutional layer is $M \times t \times d \times d$, where M is the number of the channels. After replacing this 3D convolutional layer with a spatio-temporal block, the parameters become $M \times 1 \times d \times d + M \times t \times 1 \times 1$. We can find that the parameters of a spatio-temporal block is much less than that of a 3D CNN layer.

From Tables 2, 3, and 4, we can observe that MSTC and MGSTC outperforms MST3D and MGST3D, respectively, which shows that adopting our proposed spatio-temporal convolutional blocks can achieve better results with less parameters than adopting original 3D CNNs. Table 6 also shows that replacing original 3D CNNs with our proposed spatio-temporal convolutional blocks can achieve better results with shorter running time and testing time.

6 CONCLUSIONS

Traffic flow prediction is very important and challenging because it involves many complicated factors, such as spatio-temporal correlation-based dependencies, multiple temporal patterns, and external influences. We proposed spatio-temporal convolutional blocks to extract the spatio-temporal features jointly across the whole neural network stack. Moreover, motivated by the gated mechanism utilized in RNN, we also introduce a spatio-temporal CNN-based gated mechanism to control the flow of spatio-temporal features. In the end, a framework, termed as MGSTC, that consists of multiple gated spatio-temporal convolutional branches and an external branch is designed for citywide traffic flow prediction. We utilize multiple gated patio-temporal convolutional branches to capture the spatial and multiple temporal dependencies together. A novel weighted fusion method is proposed in MGSTC to combine the features extracted by multiple gated patio-temporal convolutional branches branches. MGSTC can predict the traffic inflows and outflows simultaneously. We will further investigate the learned spatio-temporal features for better interpretability and explore graphical information of road networks in our future work.

REFERENCES

[1] Afshin Abadi, Tooraj Rajabioun, and Petros A. Ioannou. 2015. Traffic flow prediction for road transportation networks with limited traffic data. *IEEE Transactions on Intelligent Transportation Systems* 16, 2 (2015), 653–662.

[2] Florent Altché and Arnaud De La Fortelle. 2017. An LSTM network for highway trajectory prediction. In *Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems*. 353–359.

[3] Muhammad Tayyab Asif, Justin Dauwels, Chong Yang Goh, Ali Oran, Esmail Fathi, Muye Xu, Menoth Mohan Dhanya, Nikola Mitrovic, and Patrick Jaillet. 2014. Spatiotemporal patterns in large-scale traffic speed prediction. *IEEE Transactions on Intelligent Transportation Systems* 15, 2 (2014), 794–804.

[4] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the 19th International Conference on Computational Statistics. Paris France, August 22–27, 2010 Keynote, Invited and Contributed Papers*. Springer Science & Business Media, 177.

[5] George E. P. Box and David A. Pierce. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association* 65, 332 (1970), 1509–1526.

[6] Cen Chen, Kenli Li, Aijia Ouyang, Zhuo Tang, and Keqin Li. 2017. Gpu-accelerated parallel hierarchical extreme learning machine on flink for big data. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47, 10 (2017), 2740–2753.

[7] Cen Chen, Kenli Li, Sin G. Teo, Guizi Chen, Xiaofeng Zou, Xulei Yang, Ramaseshan C. Vijay, Jiashi Feng, and Zeng Zeng. 2018. Exploiting spatio-temporal correlations with multiple 3D convolutional neural networks for citywide vehicle flow prediction. In *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM'18)*. IEEE, 893–898.

[8] Cen Chen, Kenli Li, Sin G. Teo, Xiaofeng Zou, Kang Wang, Jie Wang, and Zeng Zeng. 2019. Gated residual recurrent graph neural networks for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 485–492.

[9] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794.

[10] Weihong Chen, Jiyao An, Renfa Li, Li Fu, Guoqi Xie, Md Zakirul Alam Bhuiyan, and Keqin Li. 2018. A novel fuzzy deep-learning approach to traffic flow prediction with uncertain spatial–temporal data features. *Future Generation Computer Systems* 89 (2018), 78–88.

[11] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the International Conference on Machine Learning*. 933–941.

[12] Gary A. Davis and Nancy L. Nihan. 1991. Nonparametric regression and short-term freeway traffic forecasting. *Journal of Transportation Engineering* 117, 2 (1991), 178–188.

[13] Edouard Delasalles, Ali Ziat, Ludovic Denoyer, and Patrick Gallinari. 2019. Spatio-temporal neural networks for space-time data modeling and relation discovery. *Knowledge and Information Systems* 61, 3 (2019), 1–27.

[14] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, and Linhong Zhu. 2017. Situation aware multi-task learning for traffic prediction. In *Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM'17)*. 81–90.

[15] Shengdong Du, Tianrui Li, Xun Gong, Zeng Yu, and Shi-Jinn Horng. 2018. A hybrid method for traffic flow forecasting using multimodal deep learning. In *arXiv preprint arXiv:1803.02099*.

[16] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the International Conference on Machine Learning*. 1243–1252.

[17] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 6645–6649.

[18] Simon Haykin. 1994. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. In *Proceedings of the European Conference on Computer Vision*. Springer, 630–645.

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

[22] Wei-Chiang Hong. 2011. Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm. *Neurocomputing* 74, 12–13 (2011), 2096–2107.

[23] Mohammad Reza Jabbarpour, Houman Zarrabi, Rashid Hafeez Khokhar, Shahaboddin Shamshirband, and Kim-Kwang Raymond Choo. 2018. Applications of computational intelligence in vehicle traffic congestion problem: A survey. *Soft Computing* 22, 7 (2018), 2299–2320.

[24] Ashesh Jain, Amir R. Zamir, Silvio Savarese, and Ashutosh Saxena. 2016. Structural-RNN: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5308–5317.

[25] 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.

[26] Seon Ho Kim, Junyuan Shi, Abdullah Alfarrarjeh, Daru Xu, Yuwei Tan, and Cyrus Shahabi. 2013. Real-time traffic video analysis using intel viewmont coprocessor. In *Proceedings of the International Workshop on Databases in Networked Information Systems*. 150–160.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Conference and Workshop on Neural Information Processing Systems*. 1097–1105.

[28] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[29] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. 2018. GeoMAN: Multi-level attention networks for Geo-sensory time series prediction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3428–3434.

[30] 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.

[31] Xutong Liu, Feng Chen, Yen-Cheng Lu, and Chang-Tien Lu. 2017. Spatial prediction for multivariate non-gaussian data. *ACM Transactions on Knowledge Discovery from Data* 11, 3 (2017), 36.

[32] 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* 14, 4 (2017), 818.

[33] Zhaofan Qiu, Ting Yao, and Tao Mei. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *Proceedings of the 2017 IEEE International Conference on Computer Vision*. 5534–5542.

[34] Doyen Sahoo, Steven C.H. Hoi, and Bin Li. 2019. Large scale online multiple kernel regression with application to time-series prediction. *ACM Transactions on Knowledge Discovery from Data* 13, 1 (2019), 9.

[35] Shiliang Sun, Changshui Zhang, and Guoqiang Yu. 2006. A Bayesian network approach to traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems* 7, 1 (2006), 124–132.

[36] Yongxin Tong, Yuqiang Chen, Zimu Zhou, Lei Chen, Jie Wang, Qiang Yang, Jieping Ye, and Weifeng Lv. 2017. The simpler the better: A unified approach to predicting original taxi demands based on large-scale online platforms. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1653–1662.

[37] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV'15)*. 4489–4497.

[38] Quang Thanh Tran, Zhihua Ma, Hengchao Li, Li Hao, and Quang Khai Trinh. 2015. A multiplicative seasonal ARIMA/GARCH model in EVN traffic prediction. *International Journal of Communications, Network and System Sciences* 8, 4 (2015), 43–49.

[39] DESA Un. 2018. World urbanization prospects: The 2017 revision. In *United Nations Department of Economics and Social Affairs, Population Division: New York, NY.*

[40] Mascha Van Der Voort, Mark Dougherty, and Susan Watson. 1996. Combining Kohonen maps with ARIMA time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies* 4, 5 (1996), 307–318.

[41] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*. 1235–1244.

[42] Kang Wang, Kenli Li, Liqian Zhou, Yikun Hu, Zhongyao Cheng, Jing Liu, and Cen Chen. 2019. Multiple convolutional neural networks for multivariate time series prediction. *Neurocomputing* 360 (2019), 107–119.

[43] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and S. Yu Philip. 2017. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Proceedings of the Advances in Neural Information Processing Systems*. 879–888.

[44] Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'14)*. 25–34.

[45] 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 Conference on Information and Knowledge Management (CIKM'16)*. 2203–2208.

[46] Billy M. Williams and Lester A. Hoel. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of Transportation Engineering* 129, 6 (2003), 664–672.

[47] Chun-Hsin Wu, Jan-Ming Ho, and Der-Tsai Lee. 2004. Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems* 5, 4 (2004), 276–281.

[48] Dawen Xia, Binfeng Wang, Huaqing Li, Yantao Li, and Zili Zhang. 2016. A distributed spatial–temporal weighted model on MapReduce for short-term traffic flow forecasting. *Neurocomputing* 179, C (2016), 246–263.

[49] Kun Xie, Can Peng, Xin Wang, Gaogang Xie, Jigang Wen, Jiannong Cao, Dafang Zhang, and Zheng Qin. 2018. Accurate recovery of Internet traffic data under variable rate measurements. *IEEE/ACM Transactions on Networking* 26, 3 (2018), 1137–1150.

[50] S. H. I. Xingjian, 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. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15)*. 802–810.

[51] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2018. Modeling spatial-temporal dynamics for traffic prediction. In *arXiv preprint arXiv:1803.01254*.

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

[53] Haiyang Yu, Zhihai Wu, Shuqin Wang, Yunpeng Wang, and Xiaolei Ma. 2017. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors* 27, 17 (2017), 1501.

[54] Rose Yu, Yaguang Li, Cyrus Shahabi, Ugur Demiryurek, and Yan Liu. 2017. Deep learning: A generic approach for extreme condition traffic forecasting. In *Proceedings of the 2017 SIAM International Conference on Data Mining*.

[55] Zeng Zeng, Xulei Yang, Yu Qiyun, Yao Meng, and Zhang Le. 2019. SeSe-Net: Self-supervised deep learning for segmentation. *Pattern Recognition Letters* 128 (8 2019), 23–29. DOI: https://doi.org/10.1016/j.patrec.2019.08.002

[56] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 1655–1661.

[57] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, Xiuwen Yi, and Tianrui Li. 2018. Predicting citywide crowd flows using deep spatio-temporal residual networks artificial intelligence. 259 (2018), 147–166.

[58] Le Zhang, Zenglin Shi, Ming-Ming Cheng, Yun Liu, Jia-Wang Bian, Joey Tianyi Zhou, Guoyan Zheng, and Zeng Zeng. 2019. Nonlinear regression via deep negative correlation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. DOI: https://doi.org/10.1109/TPAMI.2019.2943860

[59] Le Zhang, Zenglin Shi, Joey Tianyi Zhou, Ming-Ming Cheng, Yun Liu, Jia-Wang Bian, Zeng Zeng, and Chunhua Shen. 2020. Ordered or orderless: A revisit for video based person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. DOI: https://doi.org/10.1109/TPAMI.2020.2976969

[60] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology* 5, 3 (2014), 38.

Received July 2019; revised January 2020; accepted February 2020