DeepSTD: Mining Spatio-Temporal Disturbances of Multiple Context Factors for Citywide Traffic Flow Prediction

Chuanpan Zheng, Xiaoliang Fan, *Senior Member, IEEE*, Chenglu Wen, *Senior Member, IEEE*, Longbiao Chen, Cheng Wang, *Senior Member, IEEE*, and Jonathan Li, *Senior Member, IEEE*

***Abstract*— Deep learning techniques have been widely applied to traffic flow prediction, considering underlying routine patterns, and multiple context factors (e.g., time and weather). However, the complex spatio-temporal dependencies between inherent traffic patterns and multiple disturbances have not been fully addressed. In this paper, we propose a two-phase end-to-end deep learning framework, namely DeepSTD to uncover the spatio-temporal disturbances (STD) to predict the citywide traffic flow. In the *STD Modeling* phase, we propose an STD modeling method to model both the different regional disturbances caused by various region functions and the spatio-temporal propagating effects. In the *Prediction* phase, we eliminate the STD from the historical traffic flow to enhance the leaning of inherent traffic patterns and combine the STD at the prediction time interval to consider the future disturbances. The experimental results on two real-world datasets demonstrate that DeepSTD outperforms the state-of-the-art methods.**

***Index Terms*— Traffic flow prediction, spatio-temporal disturbances, deep learning, intelligent transportation systems.**

I. Introduction

ACCURATE citywide traffic flow prediction is of great importance to the development of modern intelligent transportation systems (ITS) [1]. It has the potential to alleviate traffic congestion, reduce fuel consumption, and thus enhance the overall efficiency of transportation networks [2].

Traffic flow prediction heavily depends on historical traffic data collected from various sensor sources, such as vehicle license plate recognition (VLPR) devices [3], ridesharing platform [4], loop detectors [5], taxi GPS systems [6], and call detail records [7]. Tremendous studies have been conducted on traffic flow prediction in past decades, such as autoregressive integrated moving average (ARIMA) [8], support vector regression (SVR) [9], artificial neural networks (ANN) [10].

Indeed, traffic tends to exhibit inherent patterns [11], [12] and thus shows high potential predictability [13], [14]. Recently, deep neural networks show more superior performance than shallow models in traffic flow prediction [15]–[17], due to the strong capability of capturing spatial or temporal dependencies. Meanwhile, traffic could be affected by multiple factors such as time [18] (e.g., travel peaks during holidays), weather [19] (e.g., a sudden rainstorm in evening peak), and social events [20] (e.g., a pop concert to be held in downtown). We coin the impact of these factors on the traffic flow as disturbance. Some researchers have considered multiple disturbances, separately or collectively, in the prediction models [21]–[24].

However, we believe that the major limitations of state-of-the-art deep learning methods are twofold.

First, the impact of multiple disturbances (e.g., time, weather) in the spatio-temporal view has not been fully addressed. On the one hand, the disturbances vary from region to region due to different region functions. For example, the traffic flow in tourist areas will significantly decrease during rough weather, whereas office areas will be less affected. One the other hand, it is probable that the disturbances could propagate in both spatial [20], [21] (i.e., extend from one region to its neighboring regions), and temporal [18] (i.e., spread over time spans) dimensions.

Second, many works usually directly utilize the historical traffic data to train deep neural networks, which might inevitably introduce noises into the model and thus degrade the prediction accuracy. Indeed, the historical traffic data contain not only regular traffic patterns [11], but also multiple disturbances. Thus, it is necessary to minimize the impact of multiple disturbances from historical traffic data before feeding into prediction models.

In this paper, we propose ***DeepSTD***, a two-phase framework to uncover the spatio-temporal disturbances (STD) to predict the citywide traffic flow. Specifically, in the *STD Modeling* phase, we first partition the city into regions and calculate the inherent influence of each point-of-interest (POI) category in every region as Inherent Influence Factor (IIF), which implies the potential region functions. Second, we model the impact of multiple factors on each POI category as Disturbance

Manuscript received February 18, 2018; revised July 30, 2018, January 25, 2019 and June 8, 2019; accepted July 25, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61872306, Grant 61771413, and Grant 61802325. The Associate Editor for this paper was F.-Y. Wang. *(Corresponding authors: Xiaoliang Fan; Jonathan Li.)*

C. Zheng, X. Fan, C. Wen, L. Chen, and C. Wang are with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Digital Fujian Institute of Urban Traffic Big Data Research, Xiamen University, Xiamen 361005, China (e-mail: zhengchuanpan@stu.xmu.edu.cn; fanxiaoliang@xmu.edu.cn; clwen@xmu.edu.cn; longbiaochen@xmu.edu.cn; cwang@xmu.edu.cn).

J. Li is with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Digital Fujian Institute of Urban Traffic Big Data Research, Xiamen University, Xiamen 361005, China, and also with the Department of Geography and Environmental Management, Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@xmu.edu.cn; junli@uwaterloo.ca).

Digital Object Identifier 10.1109/TITS.2019.2932785

Influence Factor (DIF). Third, we fuse IIF and DIF as STD via 3D convolutional neural networks (3D CNN) to model the spatio-temporal propagating effects. In the *Prediction* phase, we first eliminate the STD from the historical traffic flow. Then, the traffic flow without STD is fed into a residual neural network (ResNet) to learn the inherent traffic patterns. Finally, the output of ResNet is combined with the STD of the prediction time interval to generate the final prediction. We evaluate the performance of DeepSTD on two real-world datasets from two major cities of China. Specifically, the Xiamen dataset contains 3.15 billion VLPR records, 83,961 POIs, and the corresponding time and weather data from August 1st, 2015 to August 31st, 2016. The Chengdu dataset includes 1.10 billion GPS records provided by Didi Chuxing,[1] 69,049 POIs, and the corresponding time and weather data in November 2016. The experimental results demonstrate that DeepSTD outperforms state-of-the-art methods.

The contributions of this study are summarized as follows.

- To effectively model the multiple disturbances on the citywide traffic flow in the spatio-temporal view, we design a STD modeling method. On one hand, to model the different regional disturbances caused by various region functions, we calculate the inherent influence factor (IIF) of each POI category in every region, and model the disturbance influence factor (DIF) of multiple factors on every POI category in each time interval. On the other hand, to capture the spatio-temporal propagating effects of STD, we apply a 3D CNN based method to fuse IIF and DIF as STD.
- To further incorporate the STD into the citywide traffic flow prediction, we design a combinational mechanism. We first eliminate the STD from the historical traffic flow to represent the inherent traffic patterns. Then, we feed the traffic flow without STD into a ResNet. Finally, to further consider the future disturbances, the STD of the prediction time interval is combined with the output of ResNet as the final prediction. We conduct extensive experiments to evaluate the effectiveness of both the STD modeling method and the combinational mechanism, respectively.

The rest of this paper is organized as follows. Section II reviews the studies on traffic flow prediction. Section III presents some preliminaries of this study. Section IV details the method of DeepSTD. Section V describes the experimental setup. Section VI compares the predictive performance between DeepSTD and other methods on two real-world datasets. Finally, Section VII concludes this paper.

II. Related Works

Traffic flow prediction has gained increasing attention with the rapid development and widely deployment of intelligent transportation systems in past decades [25], [26]. Existing approaches can be generally divided into three categories: parametric, non-parametric and hybrid approaches [27].

Parametric approaches are also known as model-driven methods, where the model structure is predetermined based on certain theoretical assumptions [15], including historical average [28], autoregressive integrated moving average (ARIMA) [8], seasonal ARIMA [29], etc. [30], [31].

Non-parametric methods, such as support vector regression (SVR) [9], [32], artificial neural network (ANN) [10], [22] are designed to capture the non-linear variations in traffic data. Recently, deep neural networks has been successfully applied in various domains, e.g., computer vision, speech recognition, and natural language processing [33]–[36]. This success inspired several attempts to utilize deep learning techniques on traffic prediction problems. Huang et al. [37] introduced deep learning approaches into transportation with deep belief networks. Lv *et al.* [25] designed a stacked autoencoder (SAE) model to learn latent traffic features for traffic flow prediction. Ma *et al.* [15] developed long short-term memory (LSTM) networks to capture the long-term temporal dependency for traffic speed prediction. In general, non-parametric techniques provide more accurate prediction results than parametric approaches due to the strong generalization and learning ability [38].

To obtain adaptive models, some works explored hybrid methods. Tan *et al.* [39] applied moving average, exponential smoothing, and ARIMA, to forecast three relevant time series (i.e., weekly similarity, daily similarity, and hourly time series), and then adopted ANN to aggregate the three forecasting values as the final prediction. Zheng *et al.* [40] proposed a neural network model that combines Bayesian and ANN to predict short-term freeway traffic flows. Results show that hybrid models could outperform singular predictors [11].

Recently, many studies focus on the prediction of large-scale transportation networks. Ma *et al.* [16] applied convolutional neural networks (CNNs) to extract the spatio-temporal features from traffic data to predict large-scale, network-wide traffic speed. Zhang *et al.* [23] proposed ST-ResNet, which employed residual neural networks [41] to model the spatio-temporal dependencies for citywide crowd flows prediction. These methods capture the underlying spatio-temporal traffic patterns, but may introduce noises because they directly feed the historical traffic flow into deep neural networks.

Furthermore, transportation systems can be heavily affected by multiple factors, e.g., time, weather, and social events. The impact of weather on the traffic flow was studied in [19]. Researchers in [20] adopted real-time social media data to extract event information to forecast the subway passenger flow under event occurrences. A study presented in [22] demonstrated that the time context such as time-of-day and day-of-week were very effective for predicting traffic flows. However, these methods fail to capture the different regional disturbances caused by various region functions and neglect the spatio-temporal propagating effects.

In summary, many works have been developed for traffic flow prediction. However, the complex dependencies between inherent traffic patterns and multiple disturbances in the spatio-temporal view have not been fully addressed. In this paper, we propose an end-to-end deep-learning-based approach to effectively uncover the spatio-temporal disturbances (STD) and integrate the STD with the inherent traffic patterns in a unified framework to predict the citywide traffic flow.

[1] http://www.didichuxing.com

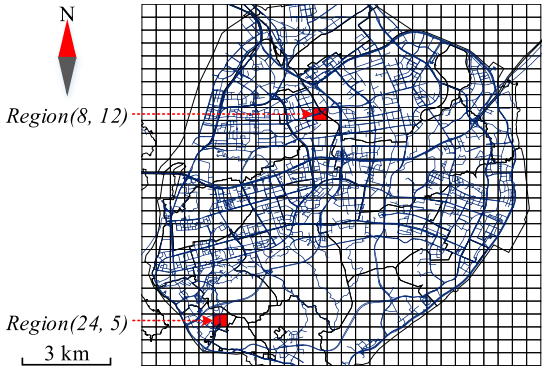


Fig. 1. Grid-based region partition and two representative regions in Xiamen.

III. Preliminaries

Definition 1 (Region): A city is uniformly partitioned into $I \times J$ grids, where a grid denotes a region. As shown in Fig. 1, $region(i, j)$ represents the grid lies in i^{th} row and j^{th} column in the city, where $i = 0, 1, ..., I-1$, $j = 0, 1, ..., J-1$.

Based on Definition 1, the traffic flow in $region(i, j)$ at time interval n can be calculated as

$$X_n(i, j, 0) = |\{v | v \in region(i, j) \wedge \tau \in n\}|, \quad (1)$$

where $|\bullet|$ denotes the cardinality of a set, v represents the recorded vehicle and τ is the recorded time. Thus, the citywide traffic flow at time interval n can be represented as $X_n \in \mathbb{R}^{I \times J \times 1}$.

According to Definition 1, the number of k^{th} POI category located in $region(i, j)$ could be counted, represented as $POI(i, j, k)$, where $k = 0, 1, ..., K-1$, and K is the number of POI categories. Thus, The number of all POI categories in all $I \times J$ regions is represented as $POI \in \mathbb{R}^{I \times J \times K}$.

In this paper, we mainly consider time and weather as context factors. At time interval n, we represent the context feature as $\mathcal{C}_n \in \mathbb{R}^{D_C}$, where D_C is the dimension of the context feature.

Problem 1 (Citywide Traffic Flow Prediction): At time interval t, given the citywide traffic flow of historical N time intervals $X_t^{history} = (X_{t-N+1}, ..., X_t) \in \mathbb{R}^{I \times J \times N}$, the context feature of both historical N time intervals and the next time interval $\mathcal{C} = (\mathcal{C}_{t-N+1}, ..., \mathcal{C}_t, \mathcal{C}_{t+1}) \in \mathbb{R}^{(N+1) \times D_C}$, and the POI number $POI \in \mathbb{R}^{I \times J \times K}$, we aim to predict the citywide traffic flow of the next time interval X_{t+1}.

IV. Our Approach

As shown in Fig. 2, we propose a two-phase framework, named DeepSTD to predict the citywide traffic flow. In the *STD Modeling* phase, we first calculate the inherent influence of each POI category in every region as Inherent Influence Factor (IIF). Then, we model the disturbance influence of multiple context factors on each POI category as Disturbance Influence Factor (DIF). Next, we combine IIF with DIF as STD via 3D convolutional neural networks (3D CNN) [42] to model the spatio-temporal propagating effects of the disturbance. In the *Prediction* phase, we first eliminate the impact of STD from the historical traffic flow. Then, we employ a residual neural network (ResNet) [41] to learn the underlying traffic patterns.

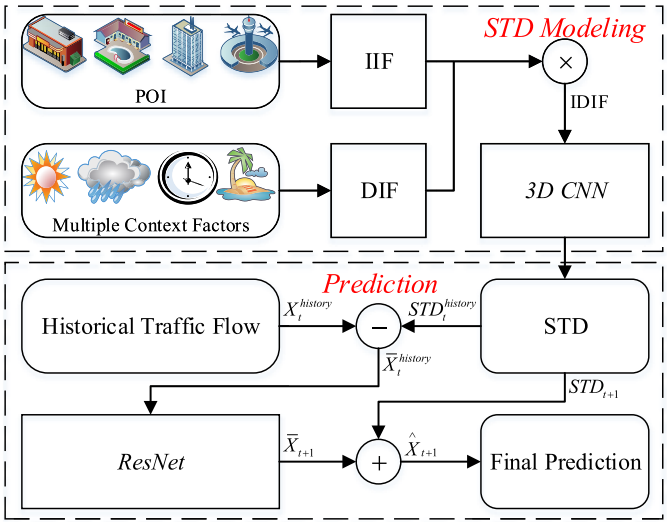
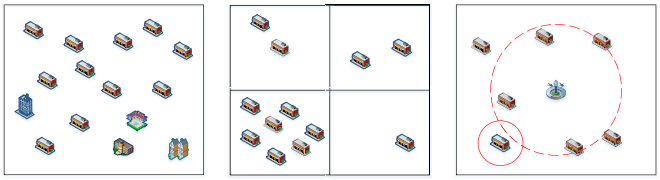


Fig. 2. The DeepSTD framework.



(a) Frequency (b) Density (c) IBD

Fig. 3. Inherent Influence Factor. (a) Frequency. (b) Density. (c) Imbalance degree (IBD), the size of red circles denotes the influence scope.

Finally, the output of ResNet is combined with the STD of the prediction time interval to generate the final prediction.

A. STD Modeling

We propose a three-step method to model the spatio-temporal disturbances (STD): Inherent Influence Factor, Disturbance Influence Factor, and Fusion.

1) Inherent Influence Factor (IIF): A POI is a specific point location of a certain function and the POIs distribution could reflect the region functions [43]. We calculate the inherent influence of each POI category in every region from the following three aspects.

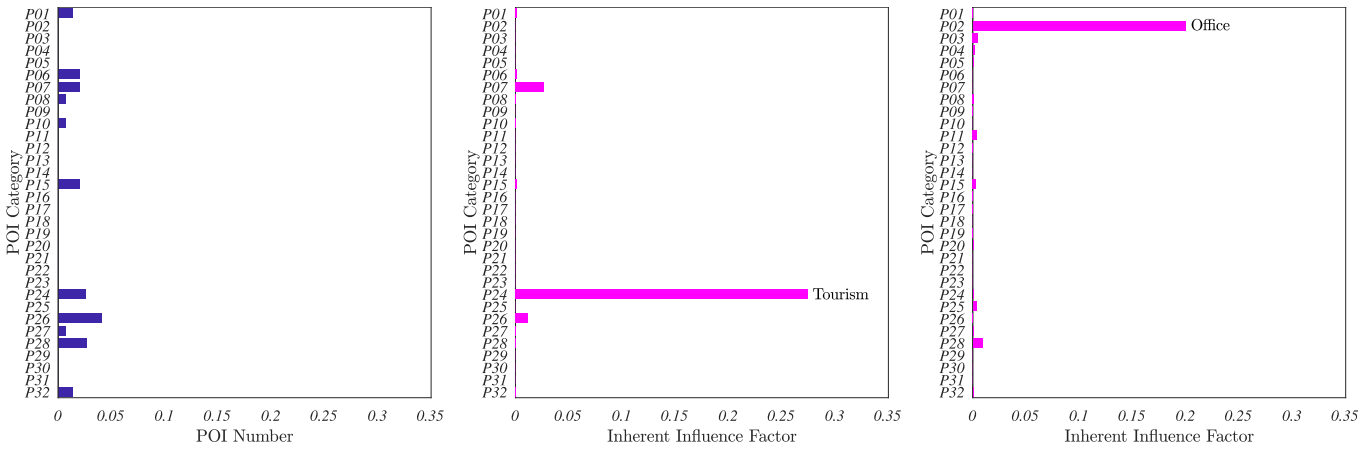
a) The *frequency* of the POI category in the located region. For example, as shown in Fig. 3(a), shops may be more influential than other POI categories due to the high frequency. Formally, the frequency of k^{th} POI category in $region(i, j)$ is defined as

$$Fre(i, j, k) = \frac{POI(i, j, k)}{\sum_k POI(i, j, k)}, \quad (2)$$

where $POI(i, j, k)$ is the number of k^{th} POI category located in $region(i, j)$.

b) The *density* of the POI category of the region in the city. For example, as shown in Fig. 3(b), a region contains more shops may be more influential than those contain fewer shops. Formally, the density of k^{th} POI category in $region(i, j)$ is defined as

$$Den(i, j, k) = \frac{POI(i, j, k)}{\sum_i \sum_j POI(i, j, k)}. \quad (3)$$



(a) POI number in $region(24, 5)$ (b) IIF value in $region(24, 5)$ (c) IIF value in $region(8, 12)$

Fig. 4. The number, IIF value of each POI category in $region(24, 5)$ and $region(8, 12)$. The list of POI categories and the two regions are presented in Table II and Fig. 1, respectively.

c) The *imbalance degree (IBD)* of the POI category distributed in the city. The imbalance distribution of the POI category could strongly influence human mobility over a large scale of the transportation network because it offers the unique service (e.g., airport, railway station) that people can only find in few regions [44]. For example, as shown in Fig. 3(c), the airport would be more influential than shops as the airport could attract visitors from all over the city.

We employ the Shannon entropy to measure the imbalance degree. Shannon entropy is a measurement of uncertainty or randomness of a single random variable [45], in which higher values indicate more uniform distribution. Formally, the Shannon entropy of k^{th} POI category is defined as

$$S(k) = -\sum_i \sum_j \left(Den(i, j, k) \times \log Den(i, j, k)\right), \quad (4)$$

where $Den(i, j, k)$ is defined in Eq. 3. When the POI category distributes uniformly in all regions, the Shannon entropy reaches the maximum:

$$S_{max} = \log(I \times J). \quad (5)$$

Thus, we define the IBD of k^{th} POI category as

$$IBD(k) = 1 - \frac{S(k)}{S_{max}}. \quad (6)$$

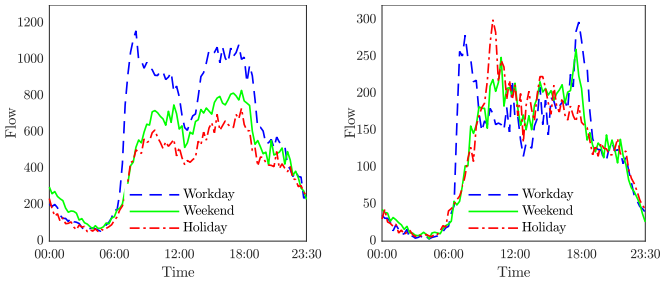
The $IBD(k)$ denotes the imbalance degree of k^{th} POI category distributed in the city, in which higher value means the POI category is more imbalance and more influential.

Finally, the IIF of k^{th} POI category of $region(i, j)$ is defined as

$$IIF(i, j, k) = Fre(i, j, k) \times Den(i, j, k) \times IBD(k). \quad (7)$$

Therefore, the IIF of all POI categories in all regions is represented as $IIF \in \mathbb{R}^{I \times J \times K}$, where K is the number of POI categories.

The IIF implies the potential region functions in every region. For example, Fig. 4(a) and 4(b) present the number, IIF value (normalized into $[0, 1]$, respectively) of each POI category in $region(24, 5)$ of Xiamen. $Region(24, 5)$ is a well-known tourist area, including the famous tourist spot, i.e., *South Putuo Temple*. We observe that the proposed IIF value effectively uncovers the region function, as shown in Fig. 4(b).

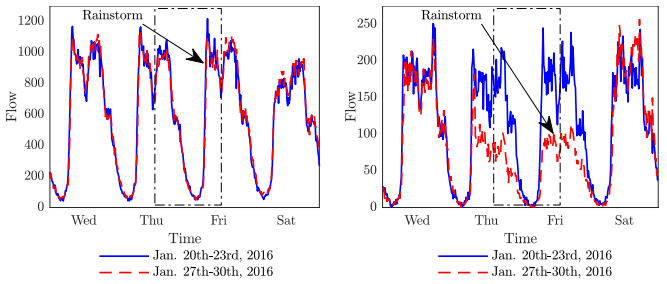


(a) $region(8, 12)$ (b) $region(24, 5)$

Fig. 5. The impact of time context on $region(8, 12)$ and $region(24, 5)$. The blue line is October 12nd, 2015 (workday), the green line is October 11st, 2015 (weekend), and the red line is October 5th, 2015 (holiday).

2) Disturbance Influence Factor (DIF): The disturbances of multiple factors on different regions differ greatly due to different region functions. Fig. 5(a) and 5(b) depict the different influence patterns of time context on $region(8, 12)$ and $region(24, 5)$. As shown in Fig. 4(c), $region(8, 12)$ tends to be an office area, and the traffic flow on weekend and holiday is less than that on workday. Whereas the phenomenon is different in $region(24, 5)$ (a tourist area, see Fig. 4(b)). Similarly, as shown in Fig. 6(a) and (b), the impact of the rainstorm is much greater on $region(24, 5)$ than that on $region(8, 12)$.

To capture the different influence patterns of multiple factors on different regions, we design a simple yet effective component to model the disturbance influence on each POI category. Specifically, we stack two fully-connected layers upon the context feature $\mathcal{C} \in \mathbb{R}^{(N+1) \times D_{\mathcal{C}}}$, where the first layer is viewed as an embedding layer followed by an activation (i.e., *relu* [46]), and the second layer is used to extract the disturbance influence of multiple factors on each POI category with K neurons (K is the number of POI categories). Consequently, we obtain an output represented as $DIF \in \mathbb{R}^{(N+1) \times K}$, indicating the disturbance influence of multiple context factors



(a) $region(8, 12)$ (b) $region(24, 5)$

Fig. 6. The impact of weather context on $region(8, 12)$ and $region(24, 5)$. The blue line is January 20th-23rd, 2016 and the red line is January 27th-30th, 2016.

on K POI categories in $N + 1$ time intervals (i.e., historical N time intervals and the prediction time interval $t + 1$).

3) Fusion: We propose a two-step fusion method to fuse IIF and DIF. In the first step, we combine IIF with DIF as

$$IDIF(n, i, j, k) = IIF(i, j, k) \times DIF(n, k). \tag{8}$$

$IDIF(n, i, j, k)$ indicates the disturbance of multiple context factors on k^{th} POI category in $region(i, j)$ at time interval n. The IDIF can be denoted as $IDIF \in \mathbb{R}^{(N+1)\times I\times J\times K}$.

In the second step, we consider the spatio-temporal propagating effects of the disturbances. On the one hand, the disturbances could propagate along the spatial dimension. The disturbances in one region could lead to a chain of reactions that influence large-scale transportation networks [20], [21]. On the other hand, the disturbances would propagate along the temporal dimension. As shown in Fig. 6(b), we observe that the impact of the rainstorm is not only limited in the rainstorm periods but also before and after the rainstorm. Based on the observations, we employ a 3D CNN upon IDIF to model the spatio-temporal propagating effects. 3D CNN could extract features from both spatial and temporal dimensions by adopting 3D convolutional operations [42], which has been widely applied in various domains [47]–[49].

We adopt six layers of 3D CNN to model the spatio-temporal propagating effects. Specifically, we use $3 \times 3 \times 3$ time-space filters for all convolutional layers. We use 64 filters in first five layers and use one filter in the output layer. To keep size constant, we do not apply pooling layers and utilize zero-padding in all three dimensions in all convolutional operations. Consequently, we obtain an output $STD \in \mathbb{R}^{(N+1)\times I\times J\times 1}$, indicating the spatio-temporal disturbances in all $I \times J$ regions of $(N + 1)$ time intervals (historical N time intervals and the prediction time interval $t + 1$). Then, we slice the spatio-temporal disturbances of historical N time intervals from STD and reshape it to $STD_t^{history} \in \mathbb{R}^{I\times J\times N}$. Likewise, we could obtain the STD of time interval $t + 1$ as $STD_{t+1} \in \mathbb{R}^{I\times J\times 1}$.

B. Prediction

As the traffic flow contains both regular traffic patterns and multiple disturbances, it is essential to avoid the effect of multiple disturbances when modeling the traffic patterns. To address this issue, in the *Prediction* phase, we propose a three-step approach to incorporate STD with the historical traffic flow to uncover the complex dependencies between inherent traffic patterns and multiple disturbances for predicting the citywide traffic flow. In the first step, we eliminate the STD from the historical traffic flow, as

$$\bar{X}_t^{history} = X_t^{history} - STD_t^{history}, \tag{9}$$

where $\bar{X}_t^{history} \in \mathbb{R}^{I\times J\times N}$ denotes the historical traffic flow without disturbances. It would be easier for neural networks to learn the inherent traffic patterns from $\bar{X}_t^{history}$ rather than directly feed the raw traffic flow $X_t^{history}$ into prediction models.

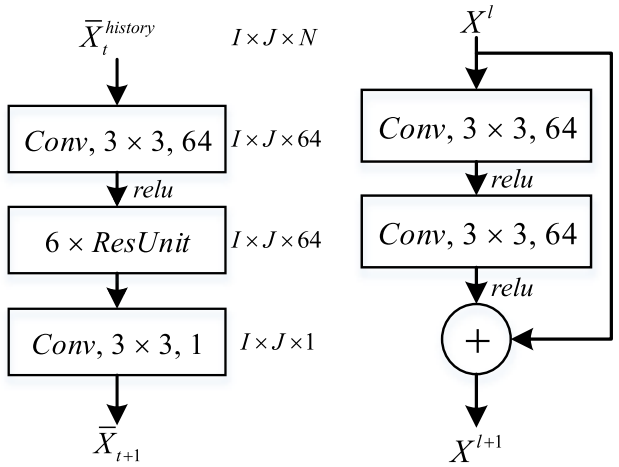
In the second step, we feed $\bar{X}_t^{history}$ into a prediction model to capture the underlying traffic patterns for prediction, as

$$\bar{X}_{t+1} = f(\bar{X}_t^{history}), \tag{10}$$

where f denotes a prediction algorithm. Herein, we employ the residual neural network (ResNet) [41] to construct the prediction model due to its superior learning ability. ResNet allows CNNs to have very deep structures by adopting residual units (ResUnits) [50], as

$$X^{l+1} = X^l + \mathbb{F}(X^l), \tag{11}$$

where X^l and X^{l+1} are the input and output of the l^{th} ResUnit, respectively, and $\mathbb{F}(\bullet)$ is the residual function. As shown in Fig. 7, we utilize 14 layers including 6 ResUnits, an input convolution layer, and an output convolution layer to construct the prediction model. In each ResUnit, we stack two convolution layers using 64 filters of 3×3 with zero-padding. We utilize 64 filters in the input layer and one filter in the output layer, respectively. Consequently, we obtain an output as $\bar{X}_{t+1} \in \mathbb{R}^{I\times J\times 1}$, which represents the predicted traffic flow without disturbances at time interval $t + 1$.



(a) ResNet (b) ResUnit

Fig. 7. The structure of the prediction model.

In the third step, we combine the output of ResNet with the STD of time interval $t+1$ to consider the future disturbances, as

$$\hat{X}_{t+1} = \bar{X}_{t+1} + STD_{t+1}, \tag{12}$$

where $\hat{X}_{t+1} \in \mathbb{R}^{I\times J\times 1}$ denotes the final prediction, containing both regular traffic patterns and multiple disturbances at time interval $t + 1$.

TABLE I
SUMMARY OF DATASETS

Data Type	Item	Xiamen	Chengdu
Traffic	Type	VLPR records	GPS records
	Number	3.15 billion	1.10 billion
POI	Categories	32	32
	Number	83,961	69,049
Time	time-of-day, day-of-week, and day type		
Weather	temperature, wind speed, visibility, rainfall, and weather condition		
	Time period	08/01/2015-08/31/2016	11/01/2016-11/30/2016

TABLE II
POI CATEGORY TAXONOMY

ID	Category	ID	Category
P01	Residence	P17	Supermarket
P02	Office	P18	Furniture building materials market
P03	Hospital	P19	Couture
P04	Clinic	P20	Electronic products store
P05	Pharmacy	P21	Fish-bird-flower market
P06	Governmental authority and public organization	P22	Bank
P07	University and scientific research institution	P23	Financial insurance institution
P08	Science and education	P24	Tourism
P09	School	P25	Food
P10	Training institution	P26	Tea bar and café
P11	Beauty salon	P27	Fast food restaurant
P12	Logistic and express	P28	Chinese restaurant
P13	Entertainment	P29	Foreign restaurant
P14	Intermediary	P30	Station/airport/dock
P15	Convenience store	P31	Bus stop
P16	Farmers market	P32	Carpark

C. Parameters Learning

As shown in Fig. 2, DeepSTD is composed of two phases (i.e., *STD Modeling* and *Prediction*). The framework can be trained end-to-end by minimizing mean squared error (MSE) between the predicted values and the ground truths:

$$\theta = \arg\min_{\theta}(X_{t+1} - \hat{X}_{t+1})^2, \tag{13}$$

where θ represents all learnable parameters in DeepSTD, which can be learned through Adam [51] optimizer via backpropagation.

V. EXPERIMENTS

A. Experimental Setup

1) Data Description: We evaluate the proposed method in two cities (i.e., Xiamen and Chengdu, China) using four types of data, i.e., traffic, POI, time and weather data. The summary of two datasets is presented in Table I.

The traffic data in Xiamen is the vehicle license plate recognition (VLPR) records, which are collected from numerous traffic cameras deployed around the city. The VLPR data has been used in various transportation researches, such as queue length estimation [52], carpooling discovery [53], and unlicensed vehicles detection [54]. We collect 3.15 billion records generated by 262 VLPR devices from August 1st, 2015 to August 31st, 2016. Each record includes the device ID (associated with location information, i.e., longitude and latitude), vehicle license plate number, recorded time, etc.

The traffic data in Chengdu is the GPS records provided by Didi Chuxing, which is one of the largest online car-hailing platforms in the world [55]. This data has been applied in many transportation researches such as ride-splitting analysis [55], passenger demand forecasting [56], and taxi demand prediction [57]. We collect 1.10 billion GPS records between November 1st and 30th, 2016. Each record includes the vehicle ID, recorded time, location information (i.e., longitude and latitude), etc.

The POI data covers 83,961 and 69,049 POIs of 32 categories (listed in Table II) in Xiamen and Chengdu, respectively. Each POI is associated with the geographical information (e.g., longitude and latitude), corresponding category, etc.

The time data indicates that each time interval is associated with the corresponding time-of-day, day-of-week, and day type (workday/weekend/holiday).

The weather data contains the numerical variables (i.e., temperature, wind speed, visibility, rainfall) and weather conditions (e.g., sunny, cloudy, rainy).

2) Data Processing: We provide three standard grid sizes (i.e., $250m \times 250m$, $500m \times 500m$, and $1000 \times 1000m$) and three standard time intervals (i.e., 5-minute, 15-minute, and 30-minute) to process the datasets. In the evaluation, we first present the experimental results using $500m \times 500m$ grid size and 15-minute time interval in Section VI-A and Section VI-B. We will discuss the impacts of grid size and time interval in Section VI-C. When adopting $500m \times 500m$ grid size, the investigated areas of Xiamen and Chengdu are divided into 28×28 and 16×16 regions, respectively, based on Definition 1.

We use the traffic data to calculate the citywide traffic flow, as introduced in Section III. The POI data is firstly used to calculate the number of each POI category in every region, and then be used to calculate the IIF, as detailed in Section IV-A. The time and weather data are considered as context data, in which the time and weather condition are encoded into binary vectors using one-hot coding, and the numerical weather variables are normalized into $[0, 1]$ using min-max normalization. Then the time and weather representations are concatenated as the context feature. The traffic flow and the IIF value are also normalized, respectively. In the evaluation, we re-scale the predicted traffic flows back to normal values.

In Xiamen dataset, the first 9 months are used for training and the rest 4 months are used for testing. In Chengdu dataset, the first 23 days are used for training and the rest 7 days are used for testing. In both datasets, we select 20% of the training data as validation sets for early-stopping to avoid over-fitting.

3) Hyperparameters: We use the traffic flow of historical 4 time intervals ($N = 4$) to predict the citywide traffic flow

of the next time interval. In constructing the neural network, we use 6 layers of 3D CNN in the *STD Modeling* phase, 6 ResUnits in the *Prediction* phase, and 64 filters in all hidden convolutional layers. We will discuss the impact of these hyperparameters in Section VI-A. In addition, We also adopt the dropout technique [58] with 20% dropout to prevent overfitting. During the training phase, the batch size is set as 32.

B. Evaluation Metrics

We evaluate the performance of DeepSTD using the following metrics:

$$RMSE = \sqrt{\frac{1}{z}\sum_i (X_i - \hat{X}_i)^2}, \tag{14}$$

$$MAE = \frac{1}{z}\sum_i |X_i - \hat{X}_i|, \tag{15}$$

$$MAPE = \frac{1}{z}\sum_i \frac{|X_i - \hat{X}_i|}{X_i}, \tag{16}$$

where X_i and $\hat{X}_i$ are the available ground truths and the corresponding predicted values, respectively; z is the number of all available ground truths.

C. Computing Environment

We conducted experiments mainly on a server with 14 CPU cores (Intel Xeon CPU E5-2683 v3 @ 2.00 GHz) and two GPUs (GeForce GTX TITAN X). We used python 3.5.2 with scikit-learn [59], tensorflow [60], and keras [61] on Ubuntu 16.04.3 LTS to build models.

VI. Results and Analysis

A. Experimental Results on Xiamen Dataset

Design of experiments. The experiments on Xiamen dataset consist of the following three parts.

- *Comparison with baseline methods.* We first compare the overall predictive performance between DeepSTD and baseline methods; second, we evaluate the performance on different days, i.e., workdays, weekends, and holidays; third, we investigate the prediction accuracy under unconventional disturbances, e.g., holidays and rough weather.
- *Comparison with variants of DeepSTD.* We first verify the effectiveness of the STD modeling method; second, we study the effect of the elimination and combination approaches; third, we evaluate the effect of multiple context factors.
- *Impact of hyperparameters.* We further study the impact of four hyperparameters in DeepSTD, i.e., number of historical time intervals, number of 3D CNN layers, number of residual units, and number of filters.

1) Comparison With Baseline Methods:

a) Overall predictive performance comparison: We compare the overall predictive performance between DeepSTD and 11 baseline methods. As shown in Table III, DeepSTD achieves the lowest RMSE (49.31), MAE (27.32) and MAPE (10.29%) among all the methods, which is 5.46% (RMSE), 6.25% (MAE), and 7.38% (MAPE) relative improvement over the best performance among baseline methods. More specifically, ARIMA [8], SVR [9], ANN [10], SAE [25], RNN [62], LSTM [63], and GRU [64] perform poorly, as they purely consider temporal dependencies. CNN [16], ConvLSTM [65], ResNet [41], and ST-ResNet [23] further capture spatial dependencies and thus achieve better performance. However, these models neglect the disturbances in the historical traffic flow. Take ST-ResNet (generally performs the best among baseline methods) for example, although it employs multiple context factors, it directly learns traffic and context features from the raw traffic and context data, and then combines them for prediction. In contrast, our DeepSTD effectively models the STD, eliminates the STD from the historical traffic flow to better learn inherent traffic patterns, and then combines the STD at the prediction time interval to consider future disturbances. As a result, our DeepSTD significantly outperforms ST-ResNet.

TABLE III
Overall Predictive Performance Comparison on Xiamen Dataset

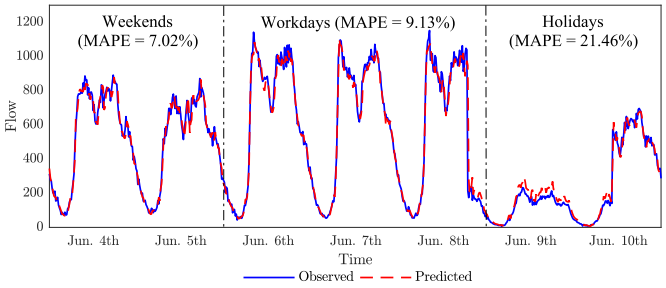
Method	RMSE	MAE	MAPE (%)
ARIMA	65.47	36.34	12.69
SVR	60.84	34.60	13.35
ANN	59.10	33.42	12.43
SAE	58.97	33.47	12.37
RNN	59.17	33.46	12.35
LSTM	57.65	32.64	12.15
GRU	57.05	32.40	11.98
CNN	54.11	30.58	11.67
ConvLSTM	55.80	31.70	12.13
ResNet	52.76	29.66	11.63
ST-ResNet	52.16	29.14	11.11
DeepSTD	**49.31**	**27.32**	**10.29**

TABLE IV
Performance on Workdays/Weekends/Holidays/All Days on Xiamen Dataset

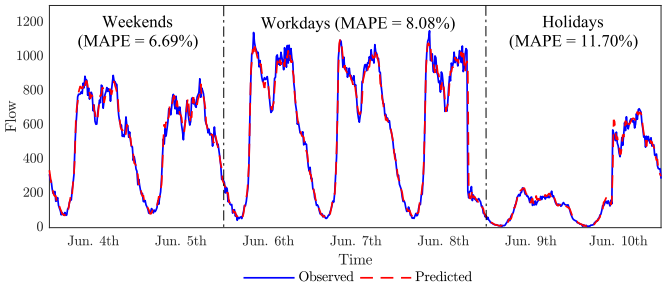
Method	RMSE			
	Workdays	Weekends	Holidays	All days
CNN	57.78	44.17	45.02	54.11
ConvLSTM	58.88	47.96	46.06	55.80
ResNet	55.90	44.78	41.85	52.76
ST-ResNet	55.46	43.46	42.85	52.16
DeepSTD	**52.28**	**41.59**	**40.20**	**49.31**

b) Performance on workdays/weekends/holidays/all days: To evaluate the performance on different days, we present the RMSE of DeepSTD and four competing baseline methods (CNN, ConvLSTM, ResNet, and ST-ResNet) in Table IV. Due to the space limitation, we only report the RMSE. We conclude the same conclusions in terms of MAE and MAPE. We exclude the results of other seven baseline methods because of their poor performances. The results show that DeepSTD outperforms other methods consistently on different days, i.e., workdays, weekends, holidays, and all days, demonstrating the robustness of our method.

c) Prediction accuracy under unconventional disturbances : To investigate the prediction accuracy under unconventional disturbances, we show the prediction results

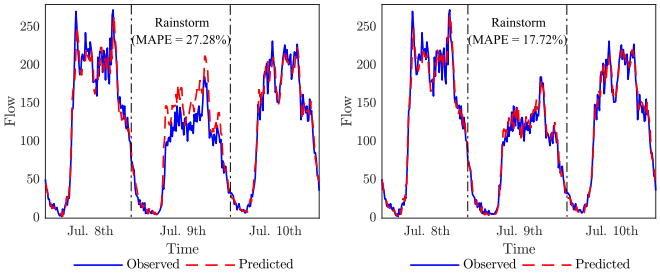


(a) The prediction result of ST-ResNet (MAPE = 12.05%)



(b) The prediction result of DeepSTD (MAPE = 8.72%)

Fig. 8. The prediction results of ST-ResNet and DeepSTD in *region*(8, 12) from June 4th to 10th, 2016.



(a) The prediction result of ST-ResNet (b) The prediction result of DeepSTD

Fig. 9. The prediction results of ST-ResNet and DeepSTD in *region*(24, 5) during the rainstorm (July 9th, 2016).

of DeepSTD and ST-ResNet (generally performs the best among baseline methods) during holidays and rainstorm in Fig. 8 and 9, respectively. As shown in Fig. 8, DeepSTD captures the fluctuation of the sharp decrease of holidays more accurately that ST-ResNet. Likewise, as shown in Fig. 9, it is easy to observe that DeepSTD is more capable of fast responding to the dynamic disturbances of the rainstorm. We believe the poor performance of ST-ResNet comes from the over-reliance on the historical regular patterns. In contrast, our proposed DeepSTD effectively models the spatio-temporal dependencies between inherent traffic patterns and multiple disturbances for prediction, and thus shows high accuracy under unconventional disturbances.

2) Comparison With Variants of DeepSTD:

a) Effect of STD modeling: In the STD modeling, we propose IIF and DIF to model the different regional disturbances caused by various region functions, and 3D CNN based fusion method to capture the spatio-temporal propagating effects. To evaluate the effectiveness of these approaches, we conduct experiments of using partial components. As show in Table V,

TABLE V
EFFECT OF STD MODELING

Method	RMSE	MAE	MAPE (%)
IIF	52.64	29.21	10.93
DIF	52.37	28.82	10.46
IIF + DIF (without 3D CNN)	51.78	28.58	10.81
POI + DIF (with 3D CNN)	50.49	27.63	10.63
IIF + DIF (with 3D CNN)	**49.31**	**27.32**	**10.29**

TABLE VI
EFFECT OF ELIMINATION AND COMBINATION

Method	RMSE	MAE	MAPE (%)
NoElimination + NoCombination	52.76	29.66	11.63
Elimination + NoCombination	50.16	27.62	11.01
NoElimination + Combination	52.19	28.56	10.63
Elimination + Combination	**49.31**	**27.32**	**10.29**

we observe that: (1) "IIF + DIF (without 3D CNN)" outperforms "IIF" and "DIF", showing that both IIF and DIF are essential for modeling the STD; (2) "IIF + DIF (with 3D CNN)" performs better than "IIF + DIF (without 3D CNN)", indicating that the propagating effect is important that should not be neglected; (3) in addition, "IIF + DIF (with 3D CNN)" performs better than "POI + DIF (with 3D CNN)", illustrating that it is beneficial to calculate the influence of POIs as IIF rather than directly use the number of POIs.

b) Effect of elimination and combination: In the *Prediction* phase, we eliminate STD from the historical traffic flow and combine the STD at the prediction time interval. In order to verify the benefits of these approaches, we conduct experiments of using or without using *Elimination* or *Combination* steps. As presented in Table VI, "NoElimination + NoCombination" means we do not consider STD and directly feed the historical traffic flow into the ResNet to generate the prediction, and it performs the worst. "Elimination + NoCombination" and "NoElimination + Combination" indicate that we adopt either *Elimination* or *Combination* approach, and both of them perform better than "NoElimination + NoCombination". This demonstrates that it is beneficial to eliminate the STD in historical time intervals and to combine the STD at the prediction time interval. Consequently, "Elimination + Combination" that utilizes both *Elimination* and *Combination* approaches performs the best.

c) Effect of multiple context factors: To evaluate the effect of multiple context factors, we show the performance of various variants that employ different factors in Fig. 10. It can be observed that the variants with combinational factors clearly perform better than those with only one factor, demonstrating the necessity to include various factors. In particular, "T + WN + WC" performs the best, indicating that the integration of multiple context factors is effective for modeling the STD in traffic flow prediction.

3) Impact of Hyperparameters: We further analyze the impact of four hyperparameters in DeepSTD, including the number of historical time intervals, 3D CNN layers, residual units, and filters. As introduced in Section V-A, these four hyparameters are set as 4, 6, 6, 64, respectively. In the

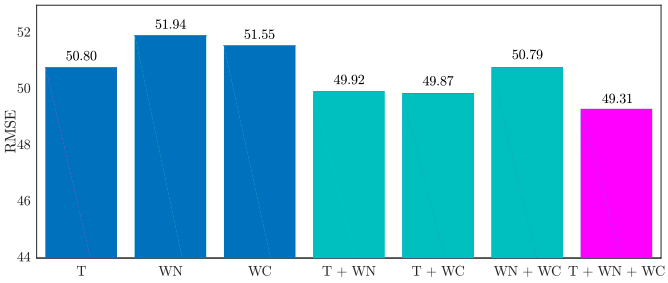


Fig. 10. Effect of multiple context factors. T, WN, and WC mean we adopt only one factor, i.e., time, numerical weather variables, and weather condition, respectively; T + WN, T + WC, and WN + WC indicate we use two factors; and T + WN + WC means we adopt three factors.

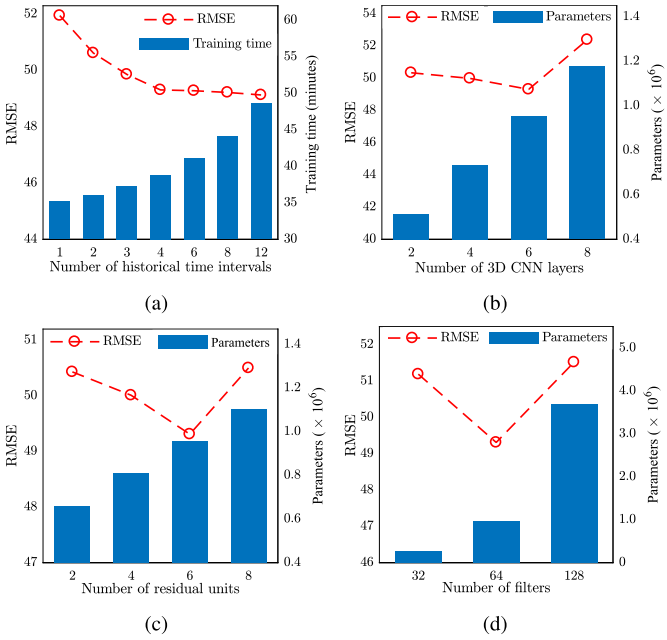


Fig. 11. The impact of four hyperparameters in DeepSTD. (a) RMSE with respect to the number of historical time intervals. (b) RMSE with respect to the number of 3D CNN layers. (c) RMSE with respect to the number of residual units. (d) RMSE with respect to the number of filters.

following discussion, we change one hyperparameter while keeping other three hyperparameters unchanged.

Fig. 11(a) presents the impact of the number of historical time intervals. In general, the RMSE gradually decreases while the training time rises with the increase of the number of historical time intervals. These results are intuitive, the more temporal information is considered, the better predictive performance can be achieved, whereas longer training time is needed.

Fig. 11(b), 11(c) and 11(d) show that the RMSE first drops and then rises with the increase of the number of 3D CNN layers, residual units, or filers. We observe that when we adopt six 3D convolutional layers, six residual units, and 64 filters, our method achieves the best performance. In general, when these hyperparameters further increase, the performance degrades. One potential reason is that when the depth or complexity of the network increases, the training becomes more difficult as more parameters need to be learned.

TABLE VII
PREDICTIVE PERFORMANCE COMPARISON ON DIFFERENT DAYS ON CHENGDU DATASET

Method	RMSE		
	Workdays	Weekends	All days
ARIMA	16.59	15.97	16.41
SVR	14.67	14.45	14.62
ANN	15.09	14.75	14.99
SAE	15.12	14.75	15.01
RNN	12.95	13.62	13.15
LSTM	11.04	11.81	11.26
GRU	11.40	11.90	11.55
CNN	11.81	11.99	11.86
ConvLSTM	12.25	12.23	12.25
ResNet	11.91	11.80	11.88
ST-ResNet	11.03	12.41	11.44
DeepSTD	**10.57**	**10.95**	**10.68**

B. Experimental Results on Chengdu Dataset

Predictive performance comparison on different days: The baseline methods on Chengdu dataset are same as that on Xiamen dataset. Due to the space limitation, we only report RMSE of different methods in Table VII. We conclude the same conclusions in terms of MAE and MAPE. The results show that DeepSTD consistently outperforms all baseline methods on workdays, weekends, and all days (no holiday during the time period).

More specifically, as shown in Table VII, we observe that ST-ResNet achieves on par or even poorer performance in comparison with other competitive baseline methods (e.g., LSTM) on Chengdu dataset. While, on Xiamen dataset, as shown in Table III, ST-ResNet performs much better than other baseline methods. It is noticed that Chengdu dataset (1 month) is smaller than Xiamen dataset (13 months). This indicates that ST-ResNet could perform well on large datasets but may has little advantage on small datasets. However, DeepSTD outperforms all baseline methods on both small and large datasets, demonstrating its robustness to different data sizes.

C. Scalability Analysis

We further study the scalability of DeepSTD in terms of grid size and time interval. Tables III and VII show the results on two datasets using $500m \times 500m$ grid size and 15-minute time interval. In the following discussion, we keep one setting unchanged while changing the other setting.

1) Predictive Performance Comparison Under Different Grid Sizes: In this paper, we partition a city into grids, where each grid denotes a region (refer to Definition 1), and we aim to predict the traffic flow in every region. Tables III and VII show the experimental results with $500m \times 500m$ grid size. To investigate the scalability of DeepSTD in terms of the grid setting, we further conduct experiments using $250m \times 250m$ and $1000m \times 1000m$ grid sizes on both datasets. Table VIII shows the RMSE of DeepSTD and four competing baseline methods (i.e., CNN, ConvLSTM, ResNet, and ST-ResNet) with different grid sizes. We observe that the RMSE grows with the increase of the grid size. This is reasonable, as larger

TABLE VIII
RMSE UNDER DIFFERENT GRID SIZES

Method	Xiamen			Chengdu		
	250m × 250m	**500m × 500m**	**1000m × 1000m**	**250m × 250m**	**500m × 500m**	**1000m × 1000m**
CNN	45.29	54.11	76.58	8.92	11.86	17.65
ConvLSTM	45.65	55.80	79.84	9.12	12.25	18.65
ResNet	43.43	52.76	74.11	8.77	11.88	17.69
ST-ResNet	42.56	52.16	71.63	8.22	11.44	17.52
DeepSTD	**40.18**	**49.31**	**67.98**	**7.89**	**10.68**	**15.81**

TABLE IX
RMSE UNDER DIFFERENT TIME INTERVALS

Method	Xiamen			Chengdu		
	5-minute	**15-minute**	**30-minute**	**5-minute**	**15-minute**	**30-minute**
CNN	20.58	54.11	128.58	6.30	11.86	18.39
ConvLSTM	21.87	55.80	133.53	6.33	12.25	19.13
ResNet	20.51	52.76	119.85	6.31	11.88	17.99
ST-ResNet	20.41	52.16	117.62	6.30	11.44	18.12
DeepSTD	**20.15**	**49.31**	**110.88**	**6.07**	**10.68**	**16.30**

TABLE X
PREDICTIVE PERFORMANCE AND TRAINING TIME

Method	Xiamen		Chengdu	
	RMSE	**Time (min)**	**RMSE**	**Time (min)**
CNN	54.11	**11.4**	11.86	**1.7**
ConvLSTM	55.80	63.5	12.25	9.4
ResNet	52.76	15.5	11.88	1.9
ST-ResNet	52.16	24.9	11.44	1.8
DeepSTD	**49.31**	38.8	**10.68**	3.1

region size usually generates more traffic flow, and leads to larger RMSE. However, DeepSTD consistently performs the best regardless of all grid sizes, demonstrating the scalability of our method in terms of the grid size.

2) Predictive Performance Comparison Under Different Time Intervals: In general, the time interval for short-term traffic flow prediction varies from 5 to 30 minutes [12], [37]. We report the experimental results with 15-minute time interval in Tables III and VII. To investigate the scalability of DeepSTD in terms of the time interval, we further conduct experiments using 5-minute and 30-minute time intervals on both datasets. The experimental results are shown in Table IX. It is reasonable that the RMSE increases when the time interval is longer, as longer time interval usually generates more traffic flow, and leads to larger RMSE. However, DeepSTD consistently performs the best regardless of all time intervals, demonstrating the scalability of our method in terms of the time interval.

D. Runtime Analysis

Table X presents the RMSE and the training time of five methods (i.e., CNN, ConvLSTM, ResNet, ST-ResNet, and DeepSTD) using a TITAN X GPU on both datasets (i.e., Xiamen and Chengdu). We notice that DeepSTD significantly outperforms other four methods whereas takes more time for training than CNN, ResNet and ST-ResNet. However, the training time (e.g., 38.8 minutes on Xiamen dataset) is acceptable as the training procedure is offline. Generally, in real applications, considerable amounts of offline resources for the training process are available [19] . In addition, after the model is trained and when it comes to the real-time prediction, DeepSTD can predict the citywide traffic flow in seconds.

VII. CONCLUSION

In this paper, we propose DeepSTD to model the spatio-temporal dependencies between inherent traffic patterns and multiple disturbances for citywide traffic flow prediction. DeepSTD contains two phases: *STD Modeling* and *Prediction*. In the *STD Modeling* phase, we propose a novel method to model the spatio-temporal disturbances (STD) by integrating different regional disturbances caused by various region functions with the spatio-temporal propagating effects. In the *Prediction* phase, we eliminate STD from the historical traffic flow to prevent multiple disturbances for capturing the inherent traffic patterns, and combine future STD to consider the disturbances at the prediction time interval. Experimental results on two real-world datasets demonstrate that DeepSTD outperforms state-of-the-art methods under various settings, e.g., prediction on different days, prediction under different grid settings and time intervals.

Emergencies could significantly disturb traffic flows in city-level. In the future, we will investigate the prediction of traffic flow during significant and rare events, such as severe traffic accidents in peak hours, temporary traffic control in a festival.

REFERENCES

[1] A. Ding, X. Zhao, and L. Jiao, "Traffic flow time series prediction based on statistics learning theory," in *Proc. IEEE 5th Int. Conf. Intell. Transp. Syst.*, Sep. 2002, pp. 727–730.

[2] B. Ran, P. J. Jin, D. Boyce, T. Z. Qiu, and Y. Cheng, "Perspectives on future transportation research: Impact of intelligent transportation system technologies on next-generation transportation modeling," *J. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 226–242, Aug. 2013.

[3] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, "Automatic license plate recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 1, pp. 42–53, Mar. 2004.

[4] L. Zhang *et al.*, "A taxi order dispatch model based on combinatorial optimization," in *Proc. 23rd SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 2151–2159.

[5] H. X. Liu, X. He, and W. Recker, "Estimation of the time-dependency of values of travel time and its reliability from loop detector data," *Transp. Res. B, Methodol.*, vol. 41, no. 4, pp. 448–461, 2007.

[6] X. Zhan, Y. Zheng, X. Yi, and S. V. Ukkusuri, "Citywide traffic volume estimation using trajectory data," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 2, pp. 272–285, Feb. 2017.

[7] S. Çolak, A. Lima, and M. C. González, "Understanding congested travel in urban areas," *Nature Commun.*, vol. 7, p. 10793, Mar. 2016.

[8] G. E. P. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *J. Amer. Statist. Assoc.*, vol. 65, no. 332, pp. 1509–1526, Apr. 1970.

[9] C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004.

[10] K. Kumar, M. Parida, and V. Katiyar, "Short term traffic flow prediction for a non urban highway using artificial neural network," *Proc. Social Behav. Sci.*, vol. 104, pp. 755–764, Dec. 2013.

[11] J. Tang, F. Liu, Y. Zou, W. Zhang, and Y. Wang, "An improved fuzzy neural network for traffic speed prediction considering periodic characteristic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2340–2350, Sep. 2017.

[12] Z. Hou and X. Li, "Repeatability and similarity of freeway traffic flow and long-term prediction under big data," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1786–1796, Jun. 2016.

[13] W. Zeng, C.-W. Fu, S. M. Arisona, S. Schubiger, R. Burkhard, and K.-L. Ma, "Visualizing the relationship between human mobility and points of interest," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 2271–2284, Aug. 2017.

[14] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[15] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.

[16] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[17] X. Song, H. Kanasugi, and R. Shibasaki, "DeepTransport: Prediction and simulation of human mobility and transportation mode at a citywide level," in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 2618–2624.

[18] J. Jun, "Understanding the variability of speed distributions under mixed traffic conditions caused by holiday traffic," *Transp. Res. C, Emerg. Technol.*, vol. 18, no. 4, pp. 599–610, Aug. 2010.

[19] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9508–9517, Dec. 2016.

[20] M. Ni, Q. He, and J. Gao, "Forecasting the subway passenger flow under event occurrences with social media," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1623–1632, Jun. 2017.

[21] L. Chen *et al.*, "Dynamic cluster-based over-demand prediction in bike sharing systems," in *Proc. Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 841–852.

[22] B. G. Çetiner, M. Sari, and O. Borat, "A neural network based traffic-flow prediction model," *Math. Comput. Appl.*, vol. 15, no. 2, pp. 269–278, 2010.

[23] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[24] Y. Tong *et al.*, "The simpler the better: A unified approach to predicting original taxi demands based on large-scale online platforms," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1653–1662.

[25] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[26] H. Tan, Y. Wu, B. Shen, P. J. Jin, and B. Ran, "Short-term traffic prediction based on dynamic tensor completion," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2123–2133, Aug. 2016.

[27] H. van Lint and C. van Hinsbergen, "Short-term traffic and travel time prediction models," *Artif. Intell. Appl. Crit. Transp. Issues*, vol. 22, pp. 22–41, Nov. 2012.

[28] A. G. Hobeika and C. K. Kim, "Traffic-flow-prediction systems based on upstream traffic," in *Proc. Vehicle Navigat. Inf. Syst. Conf.*, Aug./Sep. 1994, pp. 345–350.

[29] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[30] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using box-jenkins techniques," *Transp. Res. Board*, no. 722, pp. 1–9, 1979.

[31] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transp. Res. C, Emerg. Technol.*, vol. 4, no. 5, pp. 307–318, 1996.

[32] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6164–6173, 2009.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[34] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[35] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[36] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[37] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[38] B. Jiang and Y. Fei, "Vehicle speed prediction by two-level data driven models in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1793–1801, Jul. 2017.

[39] M.-C. Tan, S. C. Wong, J.-M. Xu, Z.-R. Guan, and P. Zhang, "An aggregation approach to short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 60–69, Mar. 2009.

[40] W. Zheng, D.-H. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," *J. Transp. Eng.*, vol. 132, pp. 114–121, Sep. 2006.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[42] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[43] F. Xu, Y. Li, H. Wang, P. Zhang, and D. Jin, "Understanding mobile traffic patterns of large scale cellular towers in urban environment," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 1147–1161, Apr. 2017.

[44] M. Alhazzani, F. Alhasoun, Z. Alawwad, and M. C. González, "Urban Attractors: Discovering patterns in regions of attraction in cities," 2016, *arXiv:1701.08696*. [Online]. Available: https://arxiv.org/abs/1701.08696

[45] A. Rényi, "On measures of entropy and information," in *Proc. 4th Symp. Math. Statist. Probab.*, 1961, pp. 547–561.

[46] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[47] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[48] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5534–5542.

[49] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jun. 2018.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.

[51] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980
[52] X. Zhan, R. Li, and S. V. Ukkusuri, "Lane-based real-time queue length estimation using license plate recognition data," *Transp. Res. C, Emerg. Technol.*, vol. 57, pp. 85–102, Aug. 2015.
[53] Y. Han, G. Wang, J. Yu, C. Liu, Z. Zhang, and M. Zhu, "A service-based approach to traffic sensor data integration and analysis to support community-wide green commute in China," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2648–2657, Sep. 2016.
[54] Y. Wang *et al.*, "Unlicensed taxis detection service based on large-scale vehicles mobility data," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2017, pp. 857–861.
[55] X. M. Chen, M. Zahiri, and S. Zhang, "Understanding ridesplitting behavior of on-demand ride services: An ensemble learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 76, pp. 51–70, Mar. 2017.
[56] J. Ke, H. Zheng, H. Yang, and X. M. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *J. Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 591–608, Dec. 2017.
[57] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 2588–2595.
[58] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[59] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[60] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement.*, 2016, pp. 265–283.
[61] F. Chollet. (2015). *Keras*. [Online]. Available: https://github.com/fchollet/keras
[62] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, 1989.
[63] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[64] K. Cho *et al.*, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.
[65] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015.

Chuanpan Zheng received the B.Sc. degree in applied physics from Shandong University, Jinan, China, in 2012. He is currently pursuing the Ph.D. degree in computer science and technology with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, China. His research interests include big data analytics and machine learning.

Xiaoliang Fan (M'06–SM'18) received the Ph.D. degree in computer science from University Pierre and Marie Curie, France, in 2012. He is currently a Senior Research Specialist with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, China. His research interests include big data analytics and ITS. He has published 50+ journals and conference papers in these areas. He received French Eiffel Excellent Ph.D. Fellowship and CSC-IBM China excellent teacher award. He is also a Senior Member of China Computer Federation.

Chenglu Wen (M'14–SM'17) received the Ph.D. degree in mechanical engineering from China Agricultural University, Beijing, China, in 2009. She is currently an Associate Professor with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, China. She has coauthored over 30 research papers in refereed journals and proceedings. Her research interests include machine vision, machine learning, and point cloud processing.

Longbiao Chen received the Ph.D. degree in computer science from Zhejiang University, China, in 2016, under a joint-cultivated doctoral program with the University of Paris VI, France. He is currently an Assistant Professor with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, China. He has published over 20 papers in top-tier journals and conferences. His research interests include ubiquitous computing, urban data mining, human mobility modeling, and context awareness.

Cheng Wang (M'04–SM'16) received the Ph.D. degree in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2002. He is currently a Professor, an Associate Dean of the School of Informatics, and the Executive Director of Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, China. His research interests include remote sensing image processing, mobile LiDAR data analysis, and multi-sensor fusion. He has coauthored over 150 papers in referred journals and top conferences, including IEEE-TGRS, PR, IEEE-TITS, AAAI, CVPR, IJCAI, and ISPRS-JPRS.

Jonathan Li (M'00-SM'11) received the Ph.D. degree in geomatics engineering from the University of Cape Town, South Africa. He is currently a Professor with the Department of Geography and Environmental Management, Department of Systems Design Engineering, University of Waterloo, Canada, and a Professor with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, China. His research interests include LiDAR and SAR data processing, machine learning, and remote sensing applications. He has coauthored over 350 publications, including over 150 research papers in referred journals and top conferences, such as IEEE-TGRS, IEEE-TITS, IEEE-JSTARS, IEEE-GRSL, ISPRS-JPRS, CVPR, IJCAI, and AAAI. He is currently serving as an Associate Editor for IEEE-JSTARS and IEEE-TITS.