This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TrafficGAN: Network-Scale Deep Traffic Prediction With Generative Adversarial Nets

Yuxuan Zhang, Senzhang Wang, Bing Chen, Jiannong Cao, *Fellow, IEEE*, and Zhiqiu Huang

***Abstract*—Traffic flow prediction has received rising research interest recently since it is a key step to prevent and relieve traffic congestion in urban areas. Existing methods mostly focus on road-level or region-level traffic prediction, and fail to deeply capture the high-order spatial-temporal correlations among the road links to perform a road network-level prediction. In this paper, we propose a network-scale deep traffic prediction model called TrafficGAN, in which Generative Adversarial Nets (GAN) is utilized to predict traffic flows under an adversarial learning framework. To capture the spatial-temporal correlations among the road links of a road network, both Convolutional Neural Nets (CNN) and Long-Short Term Memory (LSTM) models are embedded into TrafficGAN. In addition, we also design a deformable convolution kernel for CNN to make it better handle the input road network data. We extensively evaluate our proposal over two large GPS probe datasets in the arterial road network of downtown Chicago and Bay Area of California. The results show that TrafficGAN significantly outperforms both traditional statistical models and state-of-the-art deep learning models in network-scale short-term traffic flow prediction.**

***Index Terms*—Traffic prediction, generative adversarial nets, deep learning.**

I. Introduction

TRAFFIC congestion has become an ubiquitous problem for many big cities in both developed and developing countries, which can result in various issues including decreasing the road traffic capacity, slowing down the vehicle speed and increasing the air pollution. Currently, Intelligent Transportation System (ITS) has been widely adopted to improve traffic conditions. As one important goal in ITS, accurately predicting traffic conditions in real time can help the drivers better plan their routes, shorten travel time and improve traffic conditions, and thus effectively alleviate the issues of air pollution, energy consumption and traffic accidents. Therefore, the study of traffic flow prediction is of great importance in ITS and has attracted increasing research interest recently.

Most existing traffic prediction studies focused on road-level or region-level traffic data analysis, and the prediction models can be roughly categorized as statistic-based models [1], [2], neural network based models [3]–[6], nonlinear theory based models [7], [8] and hybrid models [9]. Statistic-based models basically use the historical data to predict the future trend with the assumption that the future projections have the same characteristics as the past data. This type of models are straightforward and their performance for the short-term traffic prediction is usually undesirable due to the randomness and the nonlinearity of the traffic flow [10]. Neural network based models can learn a much more complex and nonlinear prediction function, and thus they usually perform better for short-term traffic flow prediction [3]. The issue is that neural network based models usually need a large number of raw data for training, and the trained model for one road may not be applicable to other roads. The nonlinear theory based prediction models generally employ chaos theory and wavelet analysis [7], [8]. The problem of poor robustness hinders the engineering application of such methods. Hybrid models usually combine two or more prediction methods to make full use of the advantages of different methods and can achieve a better prediction result [11]. However, most existing prediction models mentioned above focus on the prediction of one single road segment. It is very difficult to extend them for predicting the traffic conditions of an entire road network of a city. Single road level traffic prediction only considers the time dependency of the traffic flow on the same road, but the spatial correlation among different roads is largely ignored. Considering the much higher complexity of the spatial-temporal correlations, the research on road network-scale traffic prediction is still in its exploratory stage.

To perform an accurate network-scale traffic prediction, we need to address the following two major challenges. First, the network-scale spatial correlations among the road segments are rather complex, and it is challenging to capture such correlations to facilitate traffic prediction. Besides the highly correlated traffic conditions between two road segments that are geographically close to each other, the traffic conditions of two road segments that are far away from each other can also be correlated due the highly dynamic traffic conditions. As all the road segments are connected and form a road network, the bad traffic condition in one road segment can spread

Manuscript received June 21, 2018; revised July 29, 2019 and November 5, 2019; accepted November 18, 2019. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802303, in part by the Hong Kong Innovation and Technology Fund under Grant ITP/024/18LP, in part by the National Natural Science Foundation of China under Grant 61672283 and Grant 61772270, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20171420, and in part by the China Computer Federation (CCF)-Tencent Open Research Fund. The Associate Editor for this article was P. Wang. *(Corresponding author: Senzhang Wang.)*

Y. Zhang, B. Chen, and Z. Huang are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China.

S. Wang is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China, and also with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: szwang@nuaa.edu.cn).

J. Cao is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China.

Digital Object Identifier 10.1109/TITS.2019.2955794

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

to neighbor road segments in the near future and to other road segments far away in a longer future. Considering the complex topology of the road network and the different context information of the road segments (POIs, locations, et al.), it is challenging by nature for existing shallow models to capture the network-scale spatial correlations. Second, there are different types of temporal correlations including the short-term temporal correlations and periodic correlations. The future traffic condition of a road segment is not only determined by its previous traffic conditions but also affected by the neighbor road links. It is also challenging to integrate the different types of and spatially coupled temporal correlations to make a very accurate prediction.

In this paper, we propose a TrafficGAN model that uses Generative Adversarial Nets (GAN) to perform traffic flow prediction for an arterial road network. GAN is currently the state-of-the-art deep generative model proposed recently [12]. GAN has presented a broad application potential and has been widely used in many areas including images and video generation [13], [14], natural language processing [15] and music generation [16]. Compared with other generative models, GAN can produce higher quality samples due to its adversarial training process [17]. As a general framework, GAN theoretically can train any kind of generator networks [12]. In our study, we can consider the future traffic flow data of a road network follows some distribution conditioned on the previous traffic flow data. Motivated by the success of GAN in video prediction which aims to predict the future frames conditioned on the previous image frames, we also try to use GAN for traffic flow prediction. The idea is that as GAN can learn the joint distribution of the data, we can use it to first learn the distribution of the future traffic flows conditioned on previous traffic flows, and then select the most likely sample from the distribution as the prediction result. The advantage of using the adversarial loss in GAN model is that it can more effectively address the blurry prediction issue in video prediction caused by only using the mean square error loss: averaging all possible futures into one single [18]. In addition, to make the traffic data in a road network suitable to be processed by GAN, we propose a deformable convolution kernel to fully capture the spatial correlations of the traffic flows of different road segments. To capture the temporal correlations, the long short-term memory (LSTM) [19] network is also introduced to model both the short-term temporal and long-term periodic correlations.

Our main contributions are summarized as follows.

- We propose TrafficGAN, a deep learning model that uses adversarial training to more effectively perform network-scale traffic flow prediction. To our knowledge, this is the first work that uses GAN for traffic prediction.
- To capture the spatial correlations, we propose a deformable convolution kernel to extract the spatial features of traffic flows from the neighbor road links. To capture the temporal correlations, LSTM is also integrated to the model to extract both the long-term and short-term historical information of the traffic flow.
- The model has been verified over two large traffic datasets collected from the arterial road networks of downtown Chicago and Bay Area of California. The results show the superior performance of the proposal.

The rest of this paper is organized as follows. Section II provides a systematic review on traffic prediction. In section III, we define the main concepts used in this paper, and then give a formal problem definition. Section IV presents the architecture and the details of the proposed TrafficGAN model. Section V discusses the experimental results. Finally, the paper is concluded in Section VI.

II. Related Work

Traffic prediction can be generally categorized into short-term, mid-term and long-term forecasting [20]. In general, the prediction period of short-term forecasting is from 5 minutes to 30 minutes, the prediction period of mid-term forecasting is between 30 minutes to several hours, and the prediction period of long-term forecasting is over 1 day. With the development of Advanced Transportation Information Service System (ATIS) and Advanced Transportation Management System (ATMS), short-term traffic prediction has attracted more research attention recently [21], and this paper also focuses on short-term traffic prediction.

The models for Short-term traffic prediction on one single road include multivariable linear regression models [22], time series models [10], Kalman filtering models [23] and non-parametric regression models. Multivariable linear regression has been studied for a long time and widely used in short-term traffic prediction. The advantages of this method include the simplicity and low equipment requirements. However, a multivariate linear regression model trained on one road is hard to be directly used in another road, leading to its poor adaptability [11]. Ahmed and Cook [24] for the first time proposed to apply auto-regressive integrated moving average (ARIMA) model to address the traffic flow prediction problem. Following this work, Williams and Hoel [25] used seasonal ARIMA model to predict the short-term traffic flow on the urban roads and achieved better performance. Cetin and Comert [26] put forward a ARIMA prediction model which included two traffic incident detection algorithms. Kalman filtering method is more robust in short-term traffic forecasting [27]. However, it cannot make full use of the spatial correlations of the traffic data and the trained model is usually location specified. Compared with the above statistic-based methods, the parameters of non-parametric models are not fixed, so they are more flexible and more complex. For example, Lippi *et al.* [28] compared support vector regression (SVR) model and seasonal ARIMA model, and concluded that SVR is more competitive during the most crowded period in prediction. However, the major issue of non-parametric methods is that they usually converge slowly and thus need more training time to converge.

Traffic prediction on a set of road segments or an entire road network is also extensively studied in recent years [1], [29]–[33]. Current methods can be broadly categorized into traditional machine learning methods and deep learning based methods. Whittaker *et al.* [1] first analyzed the interaction between the traffic flow and the average traffic flow among different road segments in a road network of Holland, and

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

predicted the traffic flow based on the topology structure of the network. AnStathopoulos and Karlaftis [34] carried out a spectrum analysis and a cross spectrum analysis of traffic flow at different locations, indicating that there are self-correlation and cross-correlation between traffic flows with different geographical locations. With the successful application of deep neural network models in various domains, deep learning based prediction models are also applied in road network level short-term traffic flow forecasting. Polson and Sokolov [32] proposed a deep learning architecture to predict traffic flows of a road network. To take advantage of time dependence of traffic flow, recurrent neural network (RNN) and long short-term memory(LSTM) have been applied to traffic forecasting [35]. But RNN and LSTM can only capture the temporal features of traffic flow but fails to learn the spatial features on the roads of a transportation network. Therefore, convolutional neural network (CNN) is usually combined with LSTM in network-level traffic flow prediction. Yao *et al.* [36] and Cheng *et al.* [37] put forward a short-term traffic forecasting method on an arterial road network by combining CNN and LSTM, but they only focused on making the prediction for the next time slot. Li *et al.* [33] introduced Diffusion Convolutional Recurrent Neural Network (DCRNN), which is an encoder-decoder based deep learning framework for multi-step traffic forecasting that can predict the traffic flows in the future several time slots simultaneously. Similarly, [38]–[41] also introduced Graph Convolutional Network (GCN) into the modeling of complex traffic network and achieved promising prediction performance. Compared with traditional machine learning methods, deep learning based methods can learn a nonlinear projection function from the raw feature space to the label space, and thus it is more powerful in short-term traffic flow forecasting. However, the blurry prediction issue caused by the usage of the standard mean squared error is not effectively addressed [18], [42], which reduces the prediction accuracy of deep learning models.

III. Preliminaries

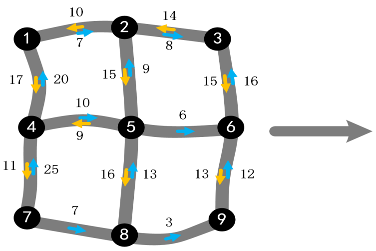
In this section, we will first give some term definitions, and then formally define the studied problem.

Definition 1: **Road Segment**. Each road in a transportation network is divided into multiple road segments based on the intersections of the road. A road segment l_{ij} between two intersections $inter_1$ and $inter_2$ is denoted as $l_{ij} = (inter_1, inter_2)$.

Definition 2: **Transportation Network**. We represent a transportation network as $G = (V, E)$, where V is the set of the intersections and the E is the set of road segments between two intersections. The intersections V can be considered as the nodes of G, and the road segments E can be considered as the links of G.

Definition 3: **Traffic flow**. The traffic flow on a road segment l is defined as the average traffic speed or the volume of the traffic per unit time of the road segment.

Definition 4: **Traffic network matrix**. We use a traffic network matrix $\mathbf{S}^t$ to characterize the traffic conditions of a transportation network G in the time slot t. The rows and columns of the matrix correspond to the intersections V of G, and each element s_{ij}^t of the matrix is the traffic flow on the road segment e_{ij}. When there is no road segment between two intersections, we set s_{ij} to -1.



	1	2	3	4	5	6	7	8	9
1	−1	7	−1	17	−1	−1	−1	−1	−1
2	10	−1	8	−1	15	−1	−1	−1	−1
3	−1	14	−1	−1	−1	15	−1	−1	−1
4	20	−1	−1	−1	10	−1	11	−1	−1
5	−1	9	−1	9	−1	6	−1	16	−1
6	−1	−1	16	−1	−1	−1	−1	−1	13
7	−1	−1	−1	25	−1	−1	−1	7	−1
8	−1	−1	−1	−1	13	−1	−1	−1	3
9	−1	−1	−1	−1	−1	12	−1	−1	−1

Fig. 1. Illustration of a transportation network and the corresponding traffic network matrix.

Definition 5: **Traffic network matrix sequence**. The traffic network matrices in a consecutive time slots $(t_1, t_2, \dots t_n)$ form a traffic network matrix sequence $\mathbf{F} = (\mathbf{S}^{t_1}, \mathbf{S}^{t_2}, \dots \mathbf{S}^{t_n})$.

The construction of a traffic network matrix is illustrated in Fig. 1. One can see that there is a road segment $l_{5,8}$ between intersections $inter_5$ and $inter_8$. The traffic flow on the road segment $l_{5,8}$ and $l_{8,5}$ are 16 mph and 13 mph, respectively. In the traffic network matrix, the corresponding entries $s_{5,8}$ and $s_{8,5}$ are 16 and 13 correspondingly. In this way, the traffic conditions on all the road segments of a transportation network in the time slot t are modeled as a traffic network matrix $\mathbf{S}_t$.

Through modeling the traffic data of an entire road network as a traffic network matrix, we can process it as a whole with deep models like an image. Meanwhile, the temporal characteristics of a traffic flow sequence is also similar to the frames in a video. Considering the significant advantages of the GAN model in image generation and video prediction, it motivates us to use it for traffic flow prediction on a transportation network.

Based on the above definitions, we formally define the studied problem as follows.

Problem Definition. Given a transportation network $G = (V, E)$ and the historical traffic network matrix sequence $\mathbf{F} = (\mathbf{S}^{t_1}, \mathbf{S}^{t_2}, \dots \mathbf{S}^{t_n})$ in the previous n time slots in G, we aim to predict the traffic conditions for all the road segments in G in the next time slot simultaneously. That is, we predict the traffic network matrix $\mathbf{S}^{t_{n+1}}$.

IV. Methodology

In this section, we will introduce the proposed TrafficGAN, which is a recurrent neural network model with adversarial training. The framework of TrafficGAN is shown in Fig. 4. As shown in this figure, the input raw historical traffic data are first processed and constructed as three traffic network matrix sequences (describe in detail later). Then these matrix sequences are fed into the TrafficGAN model for training, and the output of TrafficGAN is the traffic network matrix in the next time slot.

Previous studies show that the periodicity is a significant characteristic of the traffic flow data [34], [43]. Thus our input data include not only the traffic matrix sequence in the near past time slots, but also the traffic matrix sequences of the same time slots as the predicted time slot in the past several

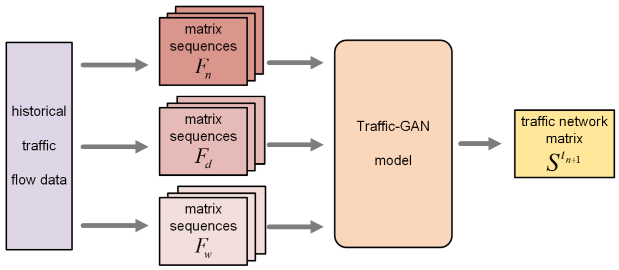


Fig. 2. The model framework.

days and weeks. Specifically, we construct the following three groups of historical traffic flow data as the model input.

$\mathbf{F}_n$: the historical traffic flow data in the previous time slots $t_1, t_2, \ldots, t_n$ of the same day. $\mathbf{F}_n = \mathbf{F} = (\mathbf{S}^{t_1}, \mathbf{S}^{t_2}, \ldots \mathbf{S}^{t_n})$.

$\mathbf{F}_d$: the historical traffic flow data in the same time slot t of the previous days capturing the daily periodicity of traffic flow. $\mathbf{F}_d = \mathbf{F}^{t^d_{1:2n}} = (\mathbf{S}^{t^d_1}, \mathbf{S}^{t^d_2}, \ldots \mathbf{S}^{t^d_{2n}})$, where t^d denotes the same time slot t in previous days.

$\mathbf{F}_w$: the historical traffic flow data in the same time slot t of the previous weeks capturing the weekly trend of traffic flow. $\mathbf{F}_w = \mathbf{F}^{t^w_{1:2n}} = (\mathbf{S}^{t^w_1}, \mathbf{S}^{t^w_2}, \ldots \mathbf{S}^{t^w_{2n}})$, where t^w denotes the same time slot t in the previous week.

The architecture of TrafficGAN is shown in Fig. 3, which contains two parts, the generator and the discriminator. The core idea of TrafficGAN is that the generator tries to generate the traffic network matrix in the next time slot t_{n+1} by using the traffic network matrix sequences $\mathbf{F} = (\mathbf{F}_n, \mathbf{F}_d, \mathbf{F}_w)$ as the input. Then both the generated and the real future traffic network matrices are input to the discriminator for training, and the discriminator tries to distinguish whether the future traffic network matrix is the generated one or the real one. The two parts are trained iteratively and simultaneously in an adversarial way, and the model converges when the generated future traffic network matrix is so similar to the real one that the discriminator cannot distinguish them.

As shown in Fig. 3, the generator contains three layers to capture the spatial-temporal features of the input traffic data. The observed traffic matrix sequence is first input a Convolution Neural Network (CNN) layer to learn the spatial features of the traffic data on the entire road network. Then a Long-Short Term Memory (LSTM) layer is used to capture the temporal correlations of the sequential traffic data. The output of the LSTM layer is next input into another CNN layer to generate the new traffic network matrices in the future time slots. The discriminator part also contains three layers, the CNN layer, the Bi-directional LSTM layer and the fully connected layer. The generated and real traffic network matrices are first input to the CNN layer to learn the latent spatial features, and then input to a Bi-directional LSTM layer to capture the latent temporal features. Next a fully connected layer is used to transform the output of the Bi-directional LSTM to a low-dimensional feature vector. Finally, a classification model is built with the feature vectors to classify whether the input future traffic network matrix is real or fake. Next, we will introduce the proposed TrafficGAN model in detail.

A. Generative Adversarial Nets

Generative Adversarial Nets (GAN) is a powerful deep generative model proposed recently [12]. Its core idea is derived from game theory, that is, a zero-sum game between a generator and a discriminator. Generator can learn the probability distribution of the real traffic flow data through a large number of historical data, and then use the learned probability distribution to generate/predict the traffic flows in the future.

The structure of GAN is shown in Fig. 4. For brevity, we use $\mathcal{D}$ and $\mathcal{G}$ to represent the discriminator and the generator respectively, and the inputs are the real traffic network matrix $\mathbf{S}^{t_{n+1}}$ and the traffic network matrix sequences $\mathbf{F}$. $\mathcal{G}(\mathbf{F})$ is the sample generated from the sample distribution learned by $\mathcal{G}$. If the input of the discriminator comes from the real data distribution, the output of the discriminator $\mathcal{D}$ is 1 (real); otherwise the output is 0 (fake).

The goal of $\mathcal{D}$ is to discriminate between the samples from the true data distribution and the ones produced by the generator; and the goal of $\mathcal{G}$ is to increase the error rate of the discriminative network $\mathcal{D}$, i.e., "fool" the discriminator by producing novel synthesized samples that appear to come from the true data distribution. These two mutually adversarial and iterative optimization processes enhance the performance of both $\mathcal{D}$ and $\mathcal{G}$. When $\mathcal{D}$ cannot correctly identify the fake samples generated by $\mathcal{G}$ and the real ones, it is considered that $\mathcal{G}$ has learned the distribution of the real data.

The training mechanism of GAN is as follows. Given an initial generator $\mathcal{G}$, the discriminator $\mathcal{D}$ is optimized firstly. As the classification model of $\mathcal{D}$ uses the Sigmoid function, the training of discriminator $\mathcal{D}$ is a process of minimizing its cross entropy, and its loss function is as follows

$$Obj^{\mathcal{D}}(\theta_{\mathcal{D}}, \theta_{\mathcal{G}}) = -\frac{1}{2}E_{\mathbf{S}^{t_{n+1}} \sim P_{data}(\mathbf{S}^{t_{n+1}})}[log\mathcal{D}(\mathbf{S}^{t_{n+1}})]$$
$$-\frac{1}{2}E_{\mathbf{F} \sim P_{\mathcal{F}}(\mathbf{F})}[log(1-\mathcal{D}(g(\mathbf{F})))], \quad (1)$$

where $P_{data}(\mathbf{S}^{t_{n+1}})$ is the real data distribution and $P_{\mathcal{F}}(\mathbf{F})$ is the prior distribution.

Different from the conventional two valued classification model, the training dataset of the discriminator $\mathcal{D}$ comes from two parts as shown in Fig. 3: the real data of the traffic network matrix and the fake data generated by $\mathcal{G}$. Given the generator $\mathcal{G}$, formula (1) is minimized to get the optimal solution. In a continuous space, formula (1) can be written as follows:

$$\begin{aligned}
&Obj^{\mathcal{D}}(\theta_{\mathcal{D}}, \theta_{\mathcal{G}}) \\
&\quad = -\frac{1}{2}\int_{\mathbf{S}^{t_{n+1}}} p_{data}(\mathbf{S}^{t_{n+1}}) log(\mathcal{D}(\mathbf{S}^{t_{n+1}})) d\mathbf{S}^{t_{n+1}} \\
&\qquad -\frac{1}{2}\int_{z} p_{\mathcal{F}}(\mathbf{F}) log(1-\mathcal{D}(g(\mathbf{F}))) d\mathbf{F} \\
&\quad = -\frac{1}{2}\int_{\mathbf{S}^{t_{n+1}}} [p_{data}(\mathbf{S}^{t_{n+1}}) log(\mathcal{D}(\mathbf{S}^{t_{n+1}})) \\
&\qquad + p_g(\mathbf{S}^{t_{n+1}}) log(1-\mathcal{D}(\mathbf{S}^{t_{n+1}}))] d\mathbf{S}^{t_{n+1}}. \quad (2)
\end{aligned}$$

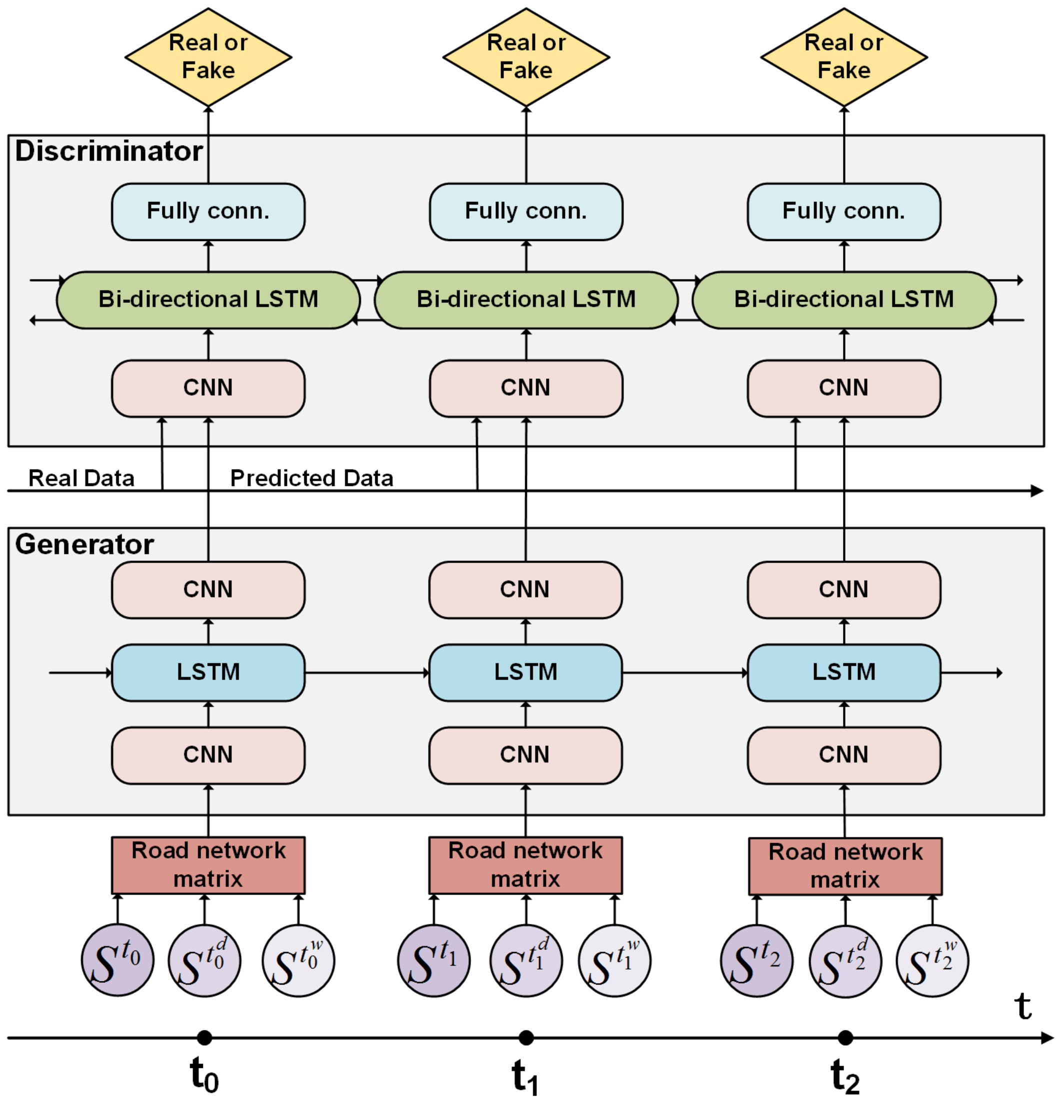


Fig. 3. The architecture of the proposed TrafficGAN model.

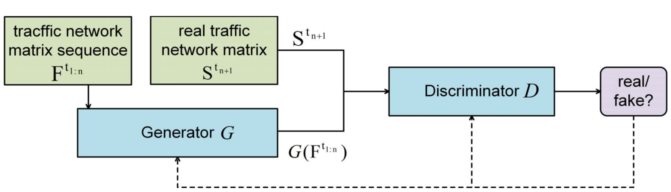


Fig. 4. Structure of GAN.

Given that the generator $\mathcal{G}$ is fixed, the optimal solution of the objective function (2) is achieved as follows

$$\mathcal{D}^*_{\mathcal{G}}(\mathbf{S}^{t_{n+1}}) = \frac{P_{data}(\mathbf{S}^{t_{n+1}})}{P_{data}(\mathbf{S}^{t_{n+1}}) + P_g(\mathbf{S}^{t_{n+1}})}.$$

$\mathcal{D}\left(\mathbf{S}^{t_{n+1}}\right)$ represents the probability that $\mathbf{S}^{t_{n+1}}$ comes from the real data. When the input data $\mathbf{S}^{t_{n+1}}$ is sampled from the real data distribution, the goal of $\mathcal{D}$ is to make the output probability $\mathcal{D}\left(\mathbf{S}^{t_{n+1}}\right)$ approaching 1. When the inputs come from the generated data $\mathcal{G}(\mathbf{F})$, $\mathcal{G}$ tries to correctly judge the source of data, making $\mathcal{D}(\mathcal{G}(\mathbf{F}))$ close to 0, while the goal of $\mathcal{G}$ is to make it close to 1. This is actually a zero sum game between $\mathcal{G}$ and $\mathcal{D}$, so the loss function of the generator $\mathcal{G}$ is $Obj^{\mathcal{G}}(\theta_{\mathcal{G}}) = -Obj^{\mathcal{D}}(\theta_{\mathcal{D}}, \theta_{\mathcal{G}})$. The objective function of GAN is as follows

$$\min_{\mathcal{G}} \max_{\mathcal{D}} f(\mathcal{D}, \mathcal{G}) = E_{\mathbf{S}^{t_{n+1}} \sim P_{data}(\mathbf{S}^{t_{n+1}})}[log\mathcal{D}(\mathbf{S}^{t_{n+1}})] + E_{z \sim P_{\mathcal{F}}(\mathbf{F})}[log(1 - \mathcal{D}(\mathcal{G}(\mathbf{F})))]. \tag{3}$$

To sum up, in the learning process of GAN, the discriminator $\mathcal{D}$ is trained to maximize the accuracy of discriminating the data generated from the real data distribution or the learned data distribution. Meanwhile, the generator $\mathcal{G}$ is trained to minimize the accuracy of discriminator. The whole model is optimized through alternatively optimizing the discriminator $\mathcal{D}$ and generator $\mathcal{G}$. The generator $\mathcal{G}$ is fixed first and the discriminator $\mathcal{D}$ is optimized, which maximizes the accuracy of $\mathcal{D}$. Then the discriminator $\mathcal{D}$ is fixed and the generator $\mathcal{G}$ is optimized, which minimizes the accuracy of $\mathcal{D}$. Finally the globally optimal solution is achieved when and only when $P_{data} = P_g$.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

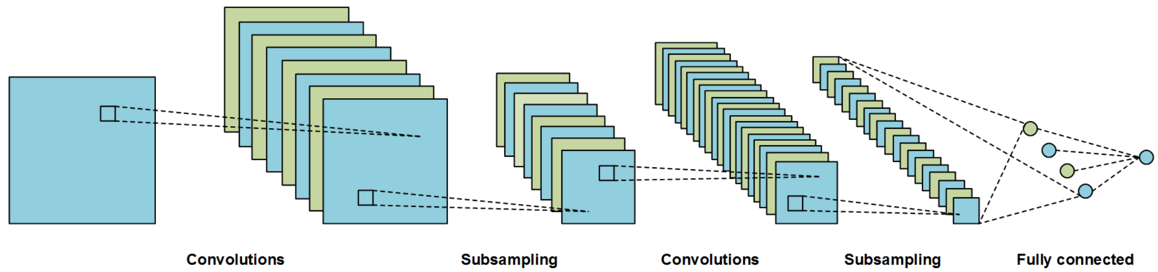


Fig. 5. The architecture of a convolution neural network.

	1	2	3	4	5	6	7	8	9
1		7		17					
2	10		8		15				
3		14				15			
4	20				10		11		
5		9		9		6		16	
6			16						13
7				25				7	
8					13				3
9						12			

(a) A normal convolution kernel

	1	2	3	4	5	6	7	8	9
1		7		17					
2	10		8		15				
3		14				15			
4	20				10		11		
5		9		9		6		16	
6			16						13
7				25				7	
8					13				3
9						12			

(b) A deformable convolution kernel

Fig. 6. Comparison between the proposed deformable convolution kernel and the normal convolution kernel.

B. The CNN Module of TrafficGAN With a Deformable Convolution Kernel

As shown in Fig. 3, both the generator and discriminator first use a CNN layer to learn the spatial features of the traffic network matrix, and the learned low dimensional features are next input to the LSTM layer. Fig. 5 shows the structure of a convolution neural network. As the input of our model is a graph rather than an image, the traditional convolution kernel cannot be applied directly and a new one that can effectively handle the input graph data is required.

The traditional convolution kernel in CNN model cannot be applied in our model due to the difference between the road traffic network matrix and the gray value matrix of an image. Given an image, usually a $n \times n$ convolution kernel is used as shown in Fig. 6(a) to extract the higher lever spatial features such as the edges. This is because the gray value matrix of an image shown in Fig. 6(a) directly reflects the locations of the pixels. That is, two neighbor pixels in the image corresponds to two neighbor entries in the matrix. However, this is not the case for a road network. The road network matrix represents the traffic flows on the road segments rather than on the nodes of a transportation network. Thus two neighbor entries in the road network matrix as shown in 6(b) actually represent two neighbor road segments (edges) rather than intersections (nodes) in the road network. We still take the road segment $(inter_5, inter_8)$ in Fig. 1 as an example. One can see that the road segment $(inter_7, inter_4)$ is the neighbor of the road segment $(inter_5, inter_8)$ in the road network matrix, but they actually are not adjacent to each other. As shown in Fig. 1, the neighbor road segments of the road segment $(inter_5, inter_8)$ are $(inter_8, inter_5)$, $(inter_5, inter_4)$, $(inter_4, inter_5)$, $(inter_5, inter_6)$, $(inter_6, inter_5)$, $(inter_7, inter_8)$, $(inter_8, inter_7)$, $(inter_9, inter_8)$, and $(inter_8, inter_9)$. Corresponding to the traffic network matrix **S**, given an entry $s_{5,8}$ as shown in Fig. 6(b), its neighbors are the entries in the same row and same volume (highlighted by green color) rather than the entries around it. Therefore, we design a deformable convolution kernel as shown in Fig. 6(b).

Our design of the convolution kernel is motivated by the idea of the deformable convolution kernel proposed in [44]. [44] lets the network automatically adjust the shape of convolution kernel according to the back propagation, so as to adapt to the areas of interest that the network focuses on. In our case, we make the convolution kernel directly scan the points which have strong spatial correlations with the center point of the kernel according to its row ID and column ID. We take the road segment $(inter_5, inter_8)$ as an example again to concretely explain the searching process. The coordinates of $(inter_5, inter_8)$ are $(5, 8)$. From Fig. 6(a), we can see that the traditional square convolution kernel (the blocks marked with red color) cannot effectively extract the spatial features of the traffic flow data in a transportation network. In Fig. 6(b), the entries covered in the convolution kernel are not around the center entry $s_{5,8}$. Instead, the proposed deformable convolution kernel scan the entries (the blocks marked with red color) in row 5, row 8, column 5 and column 8 (marked with green color) whose values are not -1 to extract the spatially correlated features.

C. The LSTM Module of TrafficGAN

The output of the CNN module is fed into the LSTM model to learn the temporal features. As a recurrent neural network, LSTM is suitable for dealing with time series data with long interval and delay [19]. The core of LSTM is the cell state. LSTM controls the cell state through the structure called “gate” and cuts or adds information to it. The gate is composed of a sigmoid network layer and a bitwise multiplying operation. It is a way of controlling information passing through the network. The output value of the sigmoid layer is between 0 and 1, indicating the proportion of information passed by each part. LSTM has three gates that control the cell state, which are input gate, forget gate and output gate. The structure of the LSTM network is illustrated in Fig. 7.

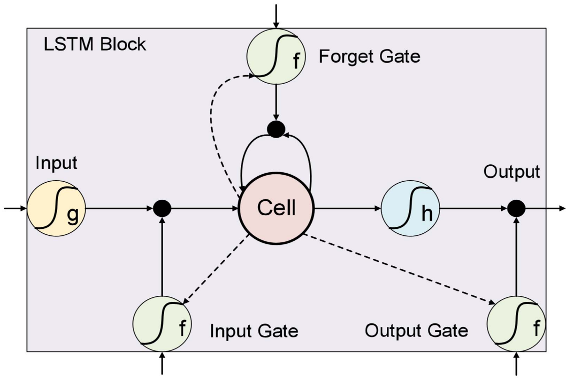


Fig. 7. The structure of a LSTM network.

Given the input data of $[x_t, h_{t-1}]$, the compact form of the equations for the forward pass of a LSTM unit with a forget gate is shown below:

Forget Gate:

$$f_t = \sigma_g \left(W_f x_t + U_f h_{t-1} + b_f\right), \tag{4}$$

Input Gate:

$$i_t = \sigma_g \left(W_i x_t + U_i h_{t-1} + b_i\right), \tag{5}$$

Output Gate:

$$o_t = \sigma_g \left(W_o x_t + U_o h_{t-1} + b_o\right), \tag{6}$$

Cell State:

$$c_t = f_t * c_{t-1} + i_t * \sigma_c \left(W_c x_t + U_c h_{t-1} + b_c\right), \tag{7}$$

Output:

$$h_t = o_t * \sigma_h \left(c_t\right), \tag{8}$$

where W, U, b are the parameters in the gates and the cell states, σ are activation functions.

Different from the regular LSTM used in the generator $\mathcal{G}$, the discriminator $\mathcal{D}$ uses a bidirectional LSTM. Bidirectional recurrent neural network (Bi-RNN) was first proposed by Schuster and Paliwal [45]. Its main goal is to increase the available information to the network by taking context in both directions into account. This structure can enhance the ability of the discriminator, so as to improve the accuracy of prediction in general.

V. Experiments

A. Dataset

The dataset used for evaluation are collected from Chicago Transit Authority (CTA) buses on Chicago's arterial streets (non-freeway streets) in real-time by continuously monitoring and analyzing their GPS traces, and this data is publicly available.[1] The average speed data of the CTA buses are produced every 10 minutes on the road segments divided through the intersections. This dataset contains the estimated speed of the buses on about 1,250 road segments covering 300 miles of the arterial roads. There are usually two traffic flows with the opposite directions on one road (except for the one-way street), and one road they have different road segment IDs. The dataset includes the following information: *Time*, *segmentID* (ID number of each road segment), *BusCount* (how many buses running on this road segment currently), *ReadCount* (the number of readings sent by the buses), and *Speed* (the average speed of the buses). We also have the following information about each road segment: street name, direction, start street name, end street name, length, the start and end locations of the segment.

We select 500 road segments from CTA road network and collect 6 months of GPS probe data on these road segments for evaluation. There are 13 million GPS probe readings in total. 70% of the data are used for training, 10% are used for validation and the remaining 20% are used for testing.

B. Evaluation Metrics

In order to evaluate the effectiveness of TrafficGAN, we first define the traffic conditions of the road segments based on the average speed of traffic flow. Average speeds of 0-9, 10-14, 15-19, 20-24, and over 25 mph correspond to the traffic conditions of *heavy congestion*, *medium-heavy congestion*, *medium congestion*, *light congestion* and *free flow* for the road segments, respectively. Note that except for a very few segments, speed on the arterials of downtown Chicago is limited to 30 mph by ordinance. As we focus on forecasting the short-term traffic flows, following the previous works whose prediction periods are from 5 minutes to 30 minutes in general [21], in this paper we conduct the traffic flow prediction in the next 10 minutes, 20 minutes, and 30 minutes, respectively.

We use the mean absolute error (MAE), the mean relative error (MRE), and the RMS error (RMSE) defined as follows as the evaluation metrics.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|x_i - \hat{x}_i\right|, \tag{9}$$

$$MRE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|x_i - \hat{x}_i\right|}{x_i}, \tag{10}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \hat{x}_i\right)^2}, \tag{11}$$

where x_i is the ground truth traffic condition and $\hat{x}_i$ is the predicted traffic condition.

C. Parameter Study

To examine the influence of parameter setup on the performance of TrafficGAN, we perform parameter analysis by increasing the hidden layer size from 64 to 512 for LSTM and setting the number of CNN layers to (3, 5, 7). Table I shows the RMSE of the prediction on the traffic flows in the next 10-min, 20-min, 30-min on weekdays, respectively.

From the table, one can see that with the increase of the hidden layer size and CNN layers, the model performance first increases. But too large a hidden layer size (larger than 256)

[1]https://data.cityofchicago.org/Transportation/Chicago-Traffic-Tracker-Historical-Congestion-Esti/77hq-huss/data

TABLE I
THE RMSE ON DIFFERENT PREDICTION TIME INTERVALS

CNN layer \ HL size		64	128	256	512
10-min	3	2.26	1.95	1.76	1.78
	5	2.09	1.81	**1.63**	**1.63**
	7	2.08	1.83	**1.62**	**1.62**
20-min	3	2.50	2.28	2.04	2.03
	5	2.31	2.07	**1.88**	**1.88**
	7	2.30	2.07	**1.86**	**1.87**
30-min	3	2.67	2.41	2.15	2.12
	5	2.51	2.26	**1.97**	**1.99**
	7	2.53	2.25	**1.98**	**1.98**

and more CNN layers (more than 5) will hurt the performance due to the issue of overfitting. Therefore, the number of the CNN layers in both $\mathcal{G}$ and $\mathcal{D}$ is set to 5, and the kernel size is set to 3×3. The layer of the LSTM models in both $\mathcal{G}$ and $\mathcal{D}$ is set to 2, and each LSTM cell has 256 internal (hidden) units. Note that the LSTM in the discriminator $\mathcal{D}$ is bidirectional to incorporate the available information by considering the contexts in both directions. The output from each LSTM cell in $\mathcal{D}$ is fed into a fully connected layer with shared weights across the time steps, and then a sigmoid function is used for the final prediction.

Mini-batch stochastic gradient descent was used to optimize the model. Learning rate was set to 0.1, and L2 regularization was applied to the weights in both generator $\mathcal{G}$ and discriminator $\mathcal{D}$. 6 epochs with a squared error loss were pre-trained in advance to predict the traffic in the next time interval in the prediction sequence. We started from the short sequence, sampled randomly from the training data, and gradually used longer sequences to train models during the pre-training period. Sometimes the discriminator becomes too strong to improve the generator which is a common problem in a adversarial training, especially when the network is initialized without pre-training. To avoid this issue, we make the updates of the discriminator $\mathcal{D}$ stop when its training loss is less than 65% of the generator $\mathcal{G}$.

D. Results

Fig. 8 compares the average speed of traffic flows of all the road segments at weekdays and weekends. One can see that the speed of traffic flows changes more smoothly at weekends, while the speed of the traffic flows in rush hours of weekdays is significantly lower than that in the other hours of a day. Considering the different traffic patterns, we perform traffic prediction in weekdays and weekends separately.

We use the TrafficGAN model to first predict the average speed of the traffic flows on the arterial transportation network in the next 10-min, 20-min, 30-min, respectively. The predicted average speed is then transformed to the traffic congestion level. The average accuracy of the future 10-min, 20-min, 30-min traffic flow prediction results are 87.5%, 84.3% and 80.9%, respectively. It shows that the model can effectively predict the congestion status of the road segments in most cases and provide a real time guidance for traffic management.

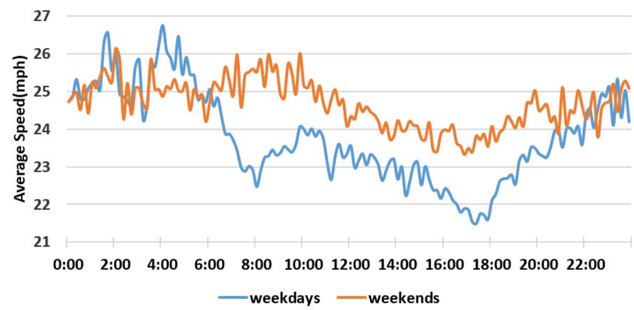
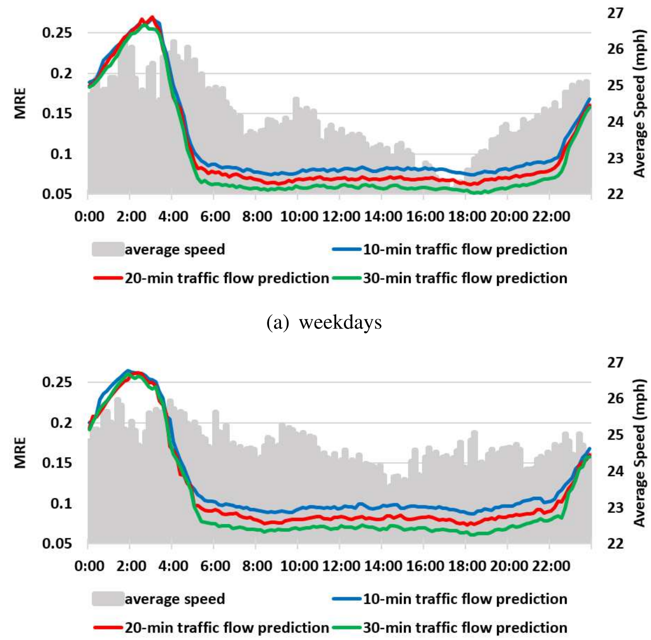


Fig. 8. Average speed of traffic flows at weekdays and weekends.



(a) weekdays

(b) weekends

Fig. 9. MRE on 10-min, 20-min, and 30-min prediction.

Evaluation on different hours of a day. Fig. 9 shows the MRE of the 10-min, 20-min, 30-min traffic flow predictions at different hours of weekdays and weekends, respectively. One can clearly see that TrafficGAN presents much better prediction result on daytime than on nights. Comparing the prediction results at different time points in this figure, one can see that the model has better prediction results when the average traffic flow speeds are relatively slow and the roads are in congestion status. This may be due to the smaller number of vehicles at night and the larger randomness of vehicle motion. Considering that the main goal of short-term traffic flow prediction is to alleviate the congestion of urban road traffic, the prediction performance during the daytime is much more important than that at nights, which also reflects the advantage of TrafficGAN.

Performance comparison with baselines. To evaluate the effectiveness of TrafficGAN, we compare its performance with the following baseline models, ARIMA [46], SVR [9], CNN-SVR [7], the scalable deep traffic flow neural networks [47] and the diffusion convolutional recurrent neural network (DCRNN) [33]. ARIMA considers the time-varying

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
THE MAE, MRE, AND RMSE ON DIFFERENT PREDICTION TIME INTERVALS (DATA OF CTA)

		TrafficGAN	DCRNN	ARIMA	SVR	CNN-SVR	Scalable Deep NN		GAN	
							CNN	LSTM	RNN	CNN
10-min (weekdays)	MAE	**1.34**	1.45	2.40	2.96	2.15	1.87	2.06	3.24	4.27
	MRE(%)	**7.71**	8.03	10.45	10.89	9.62	8.57	9.05	11.12	15.52
	RMSE	**1.63**	1.80	2.75	3.18	2.47	2.09	2.16	3.44	4.80
20-min (weekdays)	MAE	**1.55**	1.68	2.71	4.83	3.36	2.11	2.25	3.93	4.85
	MRE(%)	**8.67**	8.92	11.87	15.02	12.14	9.67	10.10	13.44	17.49
	RMSE	**1.88**	2.01	3.10	5.21	3.75	2.34	2.38	4.36	5.59
30-min (weekdays)	MAE	**1.63**	1.87	2.99	6.82	5.56	2.32	2.37	4.29	5.19
	MRE(%)	**9.02**	9.33	14.07	19.68	15.42	11.39	10.72	15.03	18.75
	RMSE	**1.97**	2.27	3.52	7.73	5.86	2.55	2.51	4.87	6.05
10-min (weekends)	MAE	**1.45**	1.53	2.82	3.53	2.56	2.21	2.45	3.82	5.13
	MRE(%)	**8.33**	8.46	12.52	13.07	11.59	10.24	10.68	13.27	18.31
	RMSE	**1.75**	1.92	3.28	3.76	2.97	2.48	2.57	4.13	5.74
20-min (weekends)	MAE	**1.67**	1.74	3.19	5.71	4.04	2.51	2.67	4.72	5.80
	MRE(%)	**9.37**	9.65	14.20	17.90	14.54	11.65	11.91	16.14	20.69
	RMSE	**2.03**	2.31	3.70	6.17	4.50	2.76	2.83	5.14	6.60
30-min (weekends)	MAE	**1.76**	1.90	3.52	8.06	6.67	2.73	2.82	5.08	6.21
	MRE(%)	**9.72**	10.84	16.71	23.22	18.22	13.44	12.86	17.90	22.25
	RMSE	**2.12**	2.49	4.18	9.17	7.02	3.05	2.98	5.80	7.14

forecast objects as a random sequence. Its basic idea is that a series of digital sequences that are time-varying but not related can be approximately described by corresponding models. SVR [9] assigns the samples with different weights and uses online learning to track the dynamic characteristic of the traffic volumes. The penalty term C is set to 0.1, and the number of historical samples is set to 5. CNN-SVR combines convolutional neural network (CNN) and support vector regression (SVR). More specifically, CNN is first applied to learn the latent feature representations from the input traffic network matrix, and then the features are input into a SVR model to predict the future traffic flows [7]. The scalable deep traffic flow neural networks include two prominent algorithms: convolutional neural network (CNN) and long short-term memory (LSTM). This model makes the traffic flow prediction for each road segment separately rather than simultaneously based on the congestion state of the neighboring road segments [47]. DCRNN is a deep learning framework for traffic forecasting with a diffusion process on a directed graph and a Seq2Seq architecture, which models the complex road networks as weighted digraphs [33]. In addition, to evaluate the effectiveness of the proposed deformable convolution kernels, we also compare TrafficGAN with the regular convolution kernels. To verify whether LSTM is better than regular RNN in traffic flow prediction, we also use GAN+RNN as a baseline. The best hyperparameters of the baselines are chosen by using the Tree-structured Parzen Estimator (TPE).

We predict the average speed of the traffic flows in the next 10 minutes, 20 minutes, 30 minutes with all these methods, and perform the prediction in the time period of 5am – 11pm. The prediction results are shown in Table II. The best results are highlighted with bold font. One can see that TrafficGAN model outperforms all the baseline models in all the cases. In order to display the results more intuitively, we present the MRE of each group of experiments in Fig. 10. One can clearly

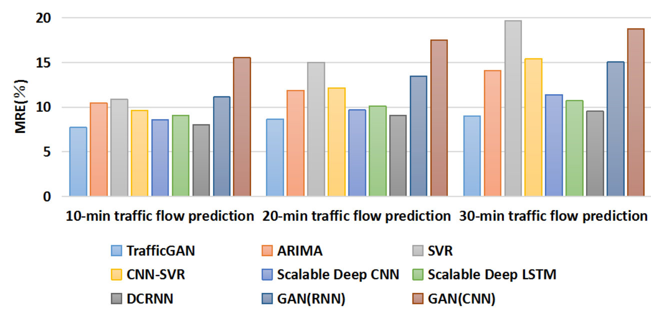


Fig. 10. MRE of the 10-min, 20-min, 30-min traffic flow predictions on various models.

see that TrafficGAN not only achieves the best prediction performance at various future time intervals, but also shows the smallest raise in error with the increase of prediction time interval. One can also see that the performance of GAN+CNN and GAN+RNN perform worse than TrafficGAN which combines LSTM and the proposed deformable convolution kernel. In order to more clearly display the performance of TrafficGAN, we compare the prediction results of several preferable models on a same segment at different times of a day. Fig 11 presents the predicted average speeds for TrafficGAN, CNN-SVR and Scalable Deep NN (CNN) as well as the real data. One can see that the prediction of TrafficGAN (red curve) better fits with the curve of the real data (blue curve) and more accurately reflects the variation trend of the average speed.

To further evaluate the effectiveness of the proposed model, we conduct experiment on a traffic flow dataset of another city, which is collected by California Transportation Agencies (CalTrans) Performance Measurement System (PeMS) in the Bay Area of California. This dataset is collected by 300 road sensors deployed on the road network of Bay Area in 6 months from Jan 1st 2017 to May 31th 2017. We also conduct the short-term traffic flow prediction in the next 10-min, 20-min,

TABLE III
MAE, MRE, AND RMSE COMPARISON OF DIFFERENT METHODS ON PEMS DATASET

		TrafficGAN	DCRNN	ARIMA	SVR	CNN-SVR	Scalable Deep NN		GAN	
							CNN	LSTM	RNN	CNN
10-min	MAE	**1.06**	1.14	1.97	2.13	1.54	1.44	1.60	2.35	3.07
	MRE(%)	**5.28**	5.36	9.86	11.02	9.77	8.06	8.41	12.65	16.52
	RMSE	**1.85**	2.04	3.56	3.80	3.01	2.69	2.79	3.94	5.18
20-min	MAE	**1.22**	1.35	2.62	2.85	2.01	2.06	2.19	2.93	3.60
	MRE(%)	**6.94**	8.71	13.54	14.44	11.73	11.10	11.35	15.06	19.31
	RMSE	**2.15**	2.92	4.30	4.59	3.35	3.21	3.29	5.18	6.65
30-min	MAE	**1.48**	1.76	3.33	3.48	2.88	2.58	2.67	3.37	4.12
	MRE(%)	**9.74**	11.73	16.21	19.55	15.34	13.04	12.48	18.56	23.07
	RMSE	**3.30**	3.97	5.76	6.18	4.73	4.20	4.10	6.07	7.47

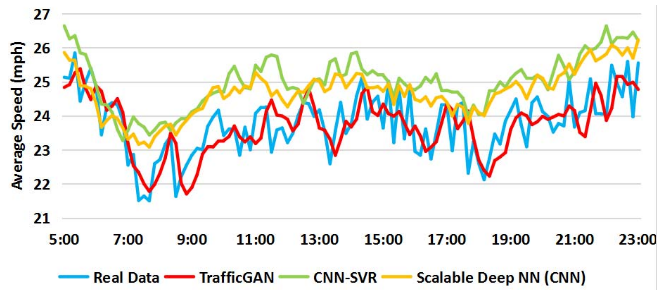


Fig. 11. Prediction results comparison on one road segment over one day.

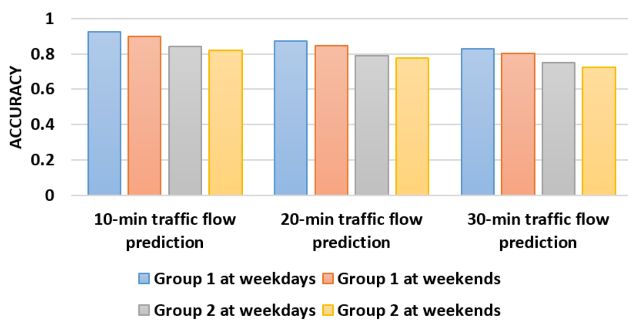


Fig. 12. Prediction accuracy of TrafficGAN under different settings.

and 30-min with TrafficGAN and baselines. The prediction results of MAE, MRE, and RMSE of different methods are shown in Table III. The best results are highlighted with bold font. From this table, one can see that TrafficGAN achieves the best performance in all the cases, which is consistent with the results with the CTA dataset. Due to space limitation, we do not put more tables and figures of the experiment results as the CTA dataset. This result further verifies the promising generalization performance of TrafficGAN.

Effect of neighbor road segments on the prediction performance. We next conduct experiment to study how the neighbor road segments affect the prediction performance on one road segment. The experimental result is shown in Fig. 12. It shows that the accuracy of the traffic prediction on one road segment is related to the number of neighbor road segments of it. We divided the road segments into two groups based on the number of its neighbor road segments. Group 1 contains the road segments with more than 3 neighbor road segments, and the other road segments form Group 2. The prediction results on the two groups of road segments in the next 10-min, 20-min, 30-min are shown in Fig. 12. One can see that the average prediction accuracy of Groups 1 is about 7% higher than that of Group 2, which demonstrates that more neighbor road segments lead to a more accurate prediction. TrafficGAN model can make full use of the spatial correlation between the neighbor road segments. In addition, the prediction accuracy on weekdays is slightly higher than that on weekends, which is also consistent with the more irregular travel patterns of people on weekends.

VI. CONCLUSION

In this paper, we proposed a deep learning model TrafficGAN for transportation network-level traffic flow prediction. TrafficGAN made full use of the spatial correlation among the road segments that were close to each other through the proposed deformable convolution kernel, and captured the temporal correlations through LSTM model. TrafficGAN also combined traffic flow data and transportation network data and trained the model in an adversarial way. We evaluated the model performance of by comparison with various baseline models on two large GPS traffic datasets, and the results showed the superiority of our proposal.

In the future, we plan to further extend TrafficGAN to multi-step traffic flow prediction scenario. TrafficGAN is designed for one-step prediction, and we plan to achieve multi-step prediction by extending current model to a Seq2Seq based encoder-decoder model. In addition, compared with images, a road network is more suitable to be modeled as a weighted digraph so that the spatial characteristics of the road network can be better captured. Such spatial features are more suitable to be learned by graph convolution network rather than convolution neural networks. Therefore, we also plan to further study how to combine GAN and GCN for road network scale traffic flow prediction in the future.

REFERENCES

[1] J. Whittaker, S. Garside, and K. Lindveld, "Tracking and predicting a network traffic process," *Int. J. Forecasting*, vol. 13, no. 1, pp. 51–61, 1997.

[2] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *J. Transp. Eng.*, vol. 121, no. 3, pp. 249–254, 1995.

[3] H. Dia, "An object-oriented neural network approach to short-term traffic forecasting," *Eur. J. Oper. Res.*, vol. 131, no. 2, pp. 253–261, Jun. 2001.

[4] M. S. Dougherty and M. R. Cobbett, "Short-term inter-urban traffic forecasts using neural networks," *Int. J. Forecasting*, vol. 13, no. 1, pp. 21–31, 1997.

[5] J. Gilmore and N. Abe, "A neural networks system for traffic congestion forecasting," in *Proc. IJCNN*, vol. 2, Oct. 1993, pp. 2025–2028.

[6] F. Jin and S. Sun, "Neural network multitask learning for traffic flow forecasting," in *Proc. IJCNN*, Jun. 2008, pp. 1897–1901.

[7] G. He, S. Ma, and Y. Li, "Study on the short-term forecasting for traffic flow based on wavelet analysis," *Syst. Eng.-Theory Pract.*, vol. 9, pp. 101–106, Sep. 2002.

[8] M. Tan, Y. Li, and J. Xu, "A hybrid ARIMA and SVM model for traffic flow prediction based on wavelet denoising," *J. Highway Transp. Res. Develop.*, vol. 7, pp. 127–132, Jul. 2009.

[9] 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.

[10] B. L. Smith and M. J. Demetsky, "Traffic flow forecasting: Comparison of modeling approaches," *J. Transp. Eng.*, vol. 123, no. 4, pp. 261–266, Jul. 1997.

[11] C. P. Van Hinsbergen, J. W. Van Lint, and F. W. Sanders, "Short term traffic prediction models," in *Proc. ITS*, 2007, pp. 22–41.

[12] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[14] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. CVPR*, Jul. 2017, pp. 4681–4690.

[15] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," 2017, *arXiv:1701.06547*. [Online]. Available: https://arxiv.org/abs/1701.06547

[16] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," 2016, *arXiv:1611.09904*. [Online]. Available: https://arxiv.org/abs/1611.09904

[17] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence generative adversarial nets with policy gradient," in *Proc. AAAI*, 2017, pp. 1–7.

[18] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," 2015, *arXiv:1511.05440*. [Online]. Available: https://arxiv.org/abs/1511.05440

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 4, pp. 303–321, Aug. 2002.

[21] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 124–132, Mar. 2006.

[22] J. Rice and E. van Zwet, "A simple and effective method for predicting travel times on freeways," in *Proc. ITS*, Aug. 2001, pp. 227–232.

[23] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transp. Res. B, Methodol.*, vol. 18, no. 1, pp. 1–11, 1984.

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

[25] 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.

[26] M. Cetin and G. Comert, "Short-term traffic flow prediction with regime switching models," *Transp. Res. Rec.*, vol. 1965, pp. 23–31, Apr. 2006.

[27] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: A general approach," *Transp. Res. B, Methodol.*, vol. 39, no. 2, pp. 141–167, 2005.

[28] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[29] S. Wang, X. Zhang, F. Li, P. S. Yu, and Z. Huang, "Efficient traffic estimation with multi-sourced data by parallel coupled hidden Markov model," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 3010–3023, Aug. 2019.

[30] S. Wang, L. He, L. Stenneth, P. S. Yu, and Z. Li, "Citywide traffic congestion estimation with social media," in *Proc. ACM SIGSPATIA GIS*, 2015, pp. 34:1–34:10.

[31] S. Wang, L. He, L. Stenneth, P. S. Yu, Z. Li, and Z. Huang, "Estimating urban traffic congestions with multi-sourced data," in *Proc. IEEE MDM*, vol. 1, Jun. 2016, pp. 82–91.

[32] N. Polson and V. Sokolov, "Deep learning predictors for traffic flows," 2016, *arXiv:1604.04527*. [Online]. Available: https://arxiv.org/abs/1604.04527v1

[33] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2018, *arXiv:1707.01926*. [Online]. Available: https://arxiv.org/abs/1707.01926

[34] A. Stathopoulos and M. Karlaftis, "Temporal and spatial variations of real-time traffic data in urban areas," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1768, no. 1, pp. 135–140, 2001.

[35] Y.-Y. Chen, Y. Lv, Z. Li, and F. Wang, "Long short-term memory model for traffic congestion prediction with online open data," in *Proc. ITSC*, Nov. 2016, pp. 132–137.

[36] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," 2018, *arXiv:1803.01254*. [Online]. Available: https://arxiv.org/abs/1803.01254

[37] X. Cheng, R. Zhang, J. Zhou, and W. Xu, "DeepTransport: Learning spatial-temporal dependency for traffic condition forecasting," 2017, *arXiv:1709.09585*. [Online]. Available: https://arxiv.org/abs/1709.09585

[38] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI*, vol. 33, 2019, pp. 922–929.

[39] Z. Cui, K. Henrickson, R. Ke, Z. Pu, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," 2018, *arXiv:1802.07007*. [Online]. Available: https://arxiv.org/abs/1802.07007

[40] L. Zhao *et al.*, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[41] Y. Zhang, S. Wang, B. Chen, and J. Cao, "GCGAN: Generative adversarial nets with graph CNN for network-scale traffic prediction," in *Proc. IJCNN*, Jul. 2019, pp. 1–8.

[42] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," 2018, *arXiv:1804.01523*. [Online]. Available: https://arxiv.org/abs/1804.01523

[43] 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.

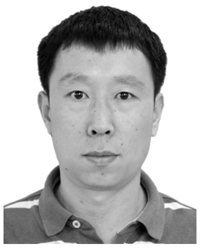
[44] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. ICCV*, Oct. 2017, pp. 764–773.

[45] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[46] S. Lee and D. B. Fambro, "Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1678, no. 1, pp. 179–188, 1999.

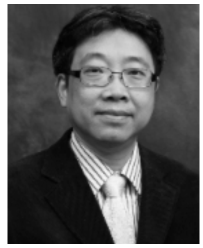
[47] M. Fouladgar, M. Parchami, R. Elmasri, and A. Ghaderi, "Scalable deep traffic flow neural networks for urban traffic congestion prediction," in *Proc. IJCNN*, May 2017, pp. 2251–2258.

Yuxuan Zhang received the B.S. degree in computer science from Wuhan University, Wuhan, China, in 2016. He is currently pursuing the master's degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His recent research interests include data mining and deep learning.



Senzhang Wang received the M.Sc. degree from Southeast University, Nanjing, China, in 2009, and the Ph.D. degree from Beihang University, Beijing, China, in 2016. He is currently an Associate Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. He has published more than 70 conference and journal articles in computer science, including ACM SIGKDD, AAAI, IJCAI, ACM SIGSPATIAL GIS, ECML-PKDD, TKDE, ACM TOIS, ACM TIST, KAIS, T-ITS, and IEEE TMM. His main research focus is on data mining, social computing, and urban computing.

Bing Chen was born in 1970. He received the B.S. and M.S. degrees from the Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1992 and 1995, respectively, and the Ph.D. degree from the College of Information Science and Technology, NUAA. He is currently a Professor with the College of Information Science and Technology, NUAA. His main research interests are computer networks and embedded systems.



Jiannong Cao (F'15) received the B.Sc. degree in computer science from Nanjing University, China, in 1982, and the M.Sc. and Ph.D. degrees in computer science from Washington State University, USA, in 1986 and 1990, respectively. He is currently a Chair Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. He has coauthored five books in mobile computing and wireless sensor networks, co-edited nine books, and published over 500 articles in major international journals and conference proceedings. His research interests include parallel and distributed computing, wireless networks and mobile computing, big data and cloud computing, pervasive computing, and fault tolerant computing. He is a Distinguished Member of ACM and a Senior Member of China Computer Federation (CCF).

Zhiqiu Huang received the B.S. and M.S. degrees from the National University of Defense Technology of China and the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics of China. He is currently a Professor and Tutor of Ph.D. candidates with the Nanjing University of Aeronautics and Astronautics of China. His major research interests include formal method, software engineering, and cloud computing.