IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.Doi Number

Spatiotemporal Scenario Generation of Traffic Flow based on LSTM-GAN

Chao Wu[1], Lei Chen[1], Guibin Wang[1], Songjian Chai[1,2], Hui Jiang[2], Jianchun Peng[1] and Zhouzhenyan Hong[3]

[1] College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China
[2] College of Physics and Optoelectronic Engineering, Shenzhen University, Shenzhen 518060, China
[3] Department of Electrical Engineering, Zhejiang University, Hangzhou 310000, China

Corresponding author: Songjian Chai (e-mail: chaisongjian@gmail.com).

This work is jointly supported by the Foundations of Shenzhen Science and Technology Committee under Grant JCYJ20170817100412438 and Grant JCYJ20190808141019317.

ABSTRACT In recent years, a surging development of vehicles and continuous enhancement of transportation infrastructures have been witnessed worldwide, leading to a remarkable growing of traffic flow data. The traffic data is highly valuable in today's society, accurate modelling of traffic flow for the concerned areas can significantly benefit the government agencies, related commercial departments and individuals. Specifically, road users are allowed to make better traveling decisions, avoid traffic congestion, reduce carbon emissions and improve traffic operation efficiency. In order to estimate the possible traffic flow scenarios within a specific area for multiple horizons, we propose a scenario generation model based on sequential generative adversarial networks (LSTM-GAN) where the long short term memory (LSTM) network is incorporated to capture the temporal dynamics involved in traffic flows. Through game training, the spatiotemporal scenarios of traffic flow in line with the characteristics of observed road network traffic flow can be well generated. These traffic scenarios can be applied in the design and planning of road traffic system, as well as in the virtual training cases of intelligent driving.

INDEX TERMS Traffic flow, Generative Adversarial Networks, Long Short term Memory Network, Scenario generation

I. INTRODUCTION

A. MOTIVATION

With the fast development of Intelligent Transportation System (ITS) technologies, people are allowed to acquire and share information that can prevent potential crashes, keep traffic moving, and decrease the negative environmental impacts of the transportation sector on society. Japan is the earliest country to carry out ITS research and the highest degree of practicality. Nowadays, it has established a relatively integrated system of traffic control and information services. Besides, it has basically realized the drawing of electronic maps and has basically completed the drawing of electronic maps covering the whole country. The research and development of ITS in the United States started a little later than Japan, but due to tremendous investment during last two decades, it has taken the leading position around the world. The Intelligent Transportation Society of America (ITS America) has spent 10 years reducing the travel time by more than 15%. in 75 metropolitan areas in the United States It has deployed intelligent operating points on the highway network in 450 administrative regions in the United States and connected to the systems of the major cities. The research of ITS in China started relatively later. Since 1995, both the Highway Research Institute of the Ministry of Communications and the Traffic Management Research Institute of the Ministry of Public Security have been engaged in ITS-related research. Afterwards, Tsinghua University, ZTE Corporation and Neusoft Group successively established ITS research and development institutions. In addition, China government has included ITS in the "Ninth Five-Year" and "Tenth Five-Year" Science and Technology Development Plan. Current ITS research mainly focuses on traffic control and management, vehicle safety and control, travel information services, human factors analysis in traffic, traffic modeling and so on. As an important studying aspect of ITS, traffic modeling aims to simulate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029230, IEEE Access

the realistic scenarios of traffic behaviors or other related parameters, which is fairly useful in densely urbanized areas.

Lately, with the development of AI technology and its extensive application in various domains, researchers began to apply this powerful tool in ITS. These applications are highly dependent on the collection of necessary dataset, wherein, such as road network data, positioning data, population, and traffic flow data. As a part of traffic flow data analytics, traffic flow scenario generation aims to provides a set of virtual data, conforming to the features of real traffic flow in practice. The simulated traffic scenarios can be used to guide and plan traffic [1]. For example, the government can set up different traffic signs or lights according to the traffic flow data of the corresponding places. Besides, the it provides valuable guidance for the road extension or infrastructure enhancement. Another popular application is the virtual training environment for intelligent driving, wherein a real virtual road environment for intelligent vehicle platform learning is restored according to the data [2]. Yet, collection of these traffic flow data is rather difficult due to the scarcity of publicly available data sources. On the other hand, the acquired data is more or less contaminated with different incompletion levels for specific locations or periods, making it unacceptable for some data-driven decision-making cases or model learning tasks. In light of this, generating a set of realistic traffic scenarios has become a prevailing practice in recent years.

B. LITERRATURE REVIEW

Early before the development of computer technology, traffic researchers can only use empirical methods and mathematical methods to conduct traffic modeling. However, the traffic system is typically a complex system, where the states and interaction laws of elements in the system are affected by multi-dimensional random factors, so it is difficult to accurately describe them with empirical model or mathematical analysis model. Owing to the development of computer engineering, a diverse number of traffic model development techniques have been proposed during the past few years. We classify them into three categories: analog simulation, data-driven models, and image-driven models.

Although analog simulation technology has been came up with as early as 50 years ago, it receives great popularity and application until recent 20 years. Throughout the whole development process, it has experienced three obvious stages. At the earliest stage, the road traffic flow simulation software TRANSYT was developed by Robertson of the Transport and Road Research Laboratory (TRRL) in 1967 to determine the optimal value of timing traffic signal parameters [3]. The data and information required by the TRANSYT system traffic model are: road network geometric characteristic, traffic volume data, economic indicators, etc. However, this simulation software is limited by its high computational burden, which is more evident when the urban network is large. To address this, an adaptive control system called Split, Cycle, and Offset Optimization (SCOOT) was developed on the basis of TRANSYT [4]. The system was tested on site in Glasgow, England in 1975, and achieved good results. SCOOT has taken the advantages of TRANSYT for all aspects, it employed real-time control, of which the performance was significantly better than that of a static system. Similar to the TRANSYT system, the SCOOT system is also composed of two key parts: traffic prediction model and timing parameter optimization. The difference is that the former is offline and the latter is online. But it also has shortcomings. The establishment of its traffic model requires a large amount of road network geometric characteristic and traffic flow data, which is time-consuming and laborious. In the medium term, with the rapid development of computer science, the accuracy of computer simulation model has been improved, and the functions became more diversified. The most typical is the NETSIM model [5] developed by the Federal Highway Administration of the United States, which is a microscopic traffic simulation model that describes the movement of a single vehicle based on scanning method. The model is highlighted with its flexibility of describing road geometric characteristic and is the most widely used simulation tool. Yet, the deficiency of NETSIM is also manifest. It requires a lot of traffic descriptions that are difficult to quantify and acquire, such as lack of full route selection, rail vehicles, etc. The latest advances (e.g., CORSIM, EACULA) have improved the conventional simulation tools in a variety of perspectives, such as universality, interactivity, maintainability and expansibility. However, the biggest issue of these simulation software is that they involve a large number of non-numerical parameterization, which requires great artificial efforts [6].

Data-driven methods aim to exploit and leverage the knowledge from data. The uncovered knowledge normally can't be interpreted explicitly and needs specific model to learn. The advantages of data-driven techniques lie in that it can make full use of the computing power of the computer and reduce labor costs. Christopher [7] provided a computer-implemented method for displaying traffic flow data on a graphical map of a road system. An animated traffic flow map of the road system is created by combining the graphical map and the status of each segment. The animated traffic flow map is created by being continuously rendered in real time. The traffic flow data is updated in real-time, and the traffic flow map immediately reflects the updated traffic data. Chao [8] proposed a novel data-driven method to populate virtual road networks with realistic traffic flows in 2018. Specifically, given a limited set of vehicle trajectories as the input samples, the approach first synthesizes a large set of vehicle trajectories. By taking the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029230, IEEE Access

spatiotemporal information of traffic flows as a 2D texture, the generation of new traffic flows can be formulated as a texture synthesis process, which is solved by minimizing a newly developed traffic texture energy. The synthesized output captures the spatiotemporal dynamics of the input traffic flows, and the vehicle interactions in it strictly follow traffic rules. Daniel [9] presented a novel approach for calculating realistic traffic flows for traffic simulators, called Flow Generator Algorithm (FGA). It requires an OpenStreetMap to cooperate with the traffic data of each site. Each site can be regarded as a node, and a node-to-node path is generated on the map. The traffic volume on these paths is in compliance with the node constraint. Varun [10] describe the process that was created to synthetically generate traffic data using a minimal dataset of GPS traces and a map of a city. A mobility model of the city is needed for this purpose, using the raw GPS traces hosted by OpenStreetMap to collected traffic demand needed to create the model. Running this newly generated traffic demand through SUMO (Simulator for Urban Mobility) gives us highly detailed data on the microscopic behavior of traffic. All the above methods require a large amount of real data, which is sometimes limited and unavailable in practice.

Traffic behavior understanding based on video image is one of the most active research topics in the field of intelligent transportation and computer vision. It intends to use computer vision technology to detect, recognize and understand traffic scenarios and traffic behaviors from traffic image sequences. A method of obtaining parameters from real traffic images through image processing technology is proposed by Aoyama [11], which attempts to forecast the traffic state that is analog to weather prediction. To this end, the author constructed the dynamic image database management system (DBMS). The system comprises a collection of dynamic images with indices, i.e. time, date, place, congestion parameters and so on. The next frame of traffic scenario will be automatically generated through index. Brickwedde et al [12] proposed a novel monocular 3D scenario flow estimation method, called Mono-SF. Mono-SF jointly estimates the 3D structure and motion of the scene by combining multi-view geometry and single-view depth information. Mono-SF considers that the scene flow should be consistent in terms of warping the reference image in the consecutive image based on the principles of multi-view geometry. For integrating single-view depth in a statistical manner, a convolutional neural network, called ProbDepthNet, is proposed. ProbDepthNet estimates pixel-wise depth distributions from a single image rather than single depth values. Additionally, as part of ProbDepthNet, a novel recalibration technique for regression problems is proposed to ensure well-calibrated distributions. The advantage of image-driven method is that a variety of features can be exploited from the same image for processing, which greatly enriches the data source. But, such methods are usually difficult to implement in practice, since they rely on advanced computing equipment for efficient image processing.

The method of generating traffic flow scenario by LSTM-GAN is a completely data-driven model, which is purely dependent on the historic samples. As a very popular deep learning network model, generative adversarial networks (GAN) has been recognized as a powerful data generation tool and has been successfully applied in a variety of real-life cases, such as image generation domain, outlier detection and so on [13]. Alec Radford and Luke Metz [14] proposed Deep Convolutional Generative Adversarial Networks (DCGAN) for image generation in 2016, it is a special network structure model specially proposed for image data processing. Dan Li et al used the GAN to solve the problem of outlier detection [15]. However, applying GAN to generate spatiotemporal traffic flow scenario has rarely been touched before. Specially, in this study, in order to better capture the temporal dependencies of traffic series [16], the LSTM network is integrated with GAN to construct a sequential generative network.

C. CONTRIBUTION

As far as the author knows, this is the first work that uses LSTM-GAN to model the traffic flow dynamics considering multiple horizons and locations. The merits of the proposed model can be summarized as follows: i), Most of conventional scenario generation models require extensive work on parameter and condition setting, whereas the whole training process of LSTM-GAN is completely data-driven, only historic samples are needed; ii), The generated scenarios in most of the existing works only consider one time slot or location, instead, the scenario data generated by the proposed LSTM-GAN can simultaneously cover consecutive time slots and multiple locations; iii), SGAN can not only learn the temporal dynamics, but also takes into account the spatial characteristics; iv), LSTM-GAN has strong generalization capability to deal with different data. Once the model is well trained, it can be applicable to any similar dataset through fine-tuning; v), LSTM-GAN has strong robustness, it is viable when the samples are permutated by a small amount of abnormalities [17].

II. MODELS: LSTM-GAN AND MAPPING NETWORKS

The GAN was first proposed by Ian Goodfellow [18] in 2014, which is composed of two deep neural networks. One is the generator network called Generator, which constantly generates new samples, and the other is the discriminator network called Discriminator, which is used to determine whether the input samples are the false samples generated by the generator or the true samples from historical data. The two networks play games back and forth during training, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029230, IEEE Access

finally the Discriminator is not able to determine whether the input samples are from Generator or real samples.

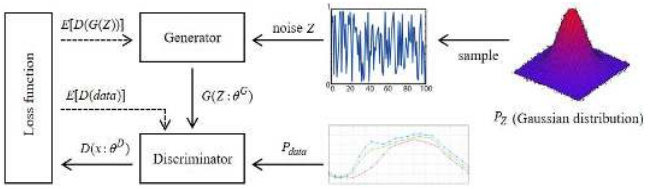


FIGURE 1. Generative adversarial networks diagram

As seen from Fig 1, the real samples distribution of the obtained data is defined as P_{data} and noise Z is assumed to be with arbitrary distribution P_Z (such as Gaussian distribution). By feeding the noise Z into the Generator, the Generator output is obtained as $G(Z,\theta^G)$. We label it as the generated samples distribution P_G, where θ^G is the weight parameter to control Generator. Assuming the input of Discriminator is x, which comes from either the real samples distribution P_{data} or the generated samples distribution P_G. The output of the Discriminator is denoted as $D(x,\theta^D)$, where θ^D is the weight parameter to control Discriminator. Since x may come from two inputs, so the output of the Discriminator is recorded as $D(data,\theta^D)$ and $D(G(Z,\theta^G),\theta^D)$. For the Discriminator, the expected output is between 0 and 1. When the Discriminator input is from real samples data, we hope that the expected output is as close to 1 as possible. When the discriminator input is from Generator, we hope that the expected output is as close to 0 as possible. Then, the Discriminator can distinguish true and false data well. So, the larger $E[D(data,\theta^D)]$ and the smaller $E[D(G(Z,\theta^G),\theta^D)]$, it means that the discriminator has a strong judgment between the real and generated samples data. For the Generator, it is hoped that after training, when the output data is fed to the discriminator, the expected output of the discriminator is as close to 1 as possible, which means that the data generated by the Generator is getting closer and closer to the real samples data. So, the larger $E[D(G(Z,\theta^G),\theta^D)]$ means that the generator has a strong generating ability, and the generated samples data is closer to the real data. Therefore, in the process of game training, the ability of the Generator and the Discriminator will be stronger and stronger, until a balance point is reached, the balance point is that the discriminator can't judge whether the generated samples is true or false, because the probability given by the discriminator is $1/2$, which means that the input of the discriminator could come from the real samples or the generated samples. According to this relationship, the loss functions of Generator and Discriminator can be defined as L_G and L_D respectively:

$$L_G = -E[D(G(Z,\theta^G),\theta^D)] \quad (1)$$

$$L_D = -E[D(data,\theta^D)] + E[D(G(Z,\theta^G),\theta^D)] \quad (2)$$

According to the training rules of GAN, the game formula can be obtained by fixing Generator and Discriminator respectively:

$$\min_G \max_D V(G,D) = E[D(data,\theta^D)] - E[D(G(Z,\theta^G),\theta^D)] \quad (3)$$

In this paper, the original GAN network is augmented by integrating LSTM network to enable spatiotemporal traffic flow generation. Besides, a multi-layer perceptron (MLP) based Inverse Mapping Network (IMN) is incorporated to carry out more specific numerical analysis. The configuration of the entire project is shown in Fig 2.

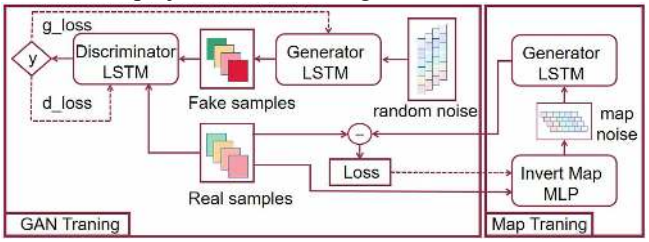


FIGURE 2. System block diagram

We divide the whole framework into two parts, one is the training part of GAN, the other is the training part of Map network which is the IMN mentioned before. In the part of GAN Training, Generator accepts multiple noise sequences as inputs, each sequence is randomly sampled from a Gaussian distribution. Then, the output sequences will be produced by Generator. Discriminator receives the input from both Generator and real samples, after calculating the loss function y, the corresponding outputs g_loss and d_loss are given to the Generator and Discriminator respectively for gradient updating. In the part of Map Training, the Generator completely duplicates the well-trained GAN model from the training part of GAN. Its parameters are fixed here, and no more training is needed in this part. The purpose of IMN is to enhance the training of generator and discriminator by updating a mapping from the real-time space to a certain latent space. The input of IMN comes from the true samples, and the output is the map noise. Then, the map noise will then be fed to the Generator, and the output of Generator will calculate the loss function with the real samples. The result will be returned to the IMN for gradient update.

we modify the network layer of the original GAN network, and use LSTM network to replace the original fully connected neural network to deal with the sequential data. LSTM is a variant of Recurrent Neural Network (RNN). It resolves the gradient vanishing issue of the traditional RNN by introducing a cell state c. The cell state acts as an accumulator of the state information, in the process of network forward propagation, the cell state is accessed, written and cleared by three sigmoid function governed gates. The configuration of LSTM network is shown in Fig 3. The left is the schematic diagram of LSTM network, concatenated by a set of LSTM units. The right figure block depicts the internal configuration of a LSTM unit, where c_t is the state value at time t, which selectively retains the state before time t, ensuring that the earliest information will not be lost in the case of long time series input. h_t is the output value of time t, which will be the input of time $t+1$. Because the output of the previous time is the input of the next time,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029230, IEEE Access

the LSTM network has a strong correlation in time, which is the reason why it can deal with the time series problem well.

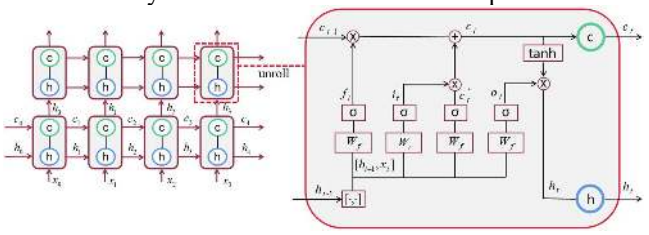


FIGURE 3. Schematic diagram of LSTM network

In the expanded view on the right, $[h_{t-1}, x_t]$ is the merged input, σ is sigmoid function, f_t is the output of the forgetting gate, which determines the amount of the state value at the previous time is retained in the next state; i_t is the input gate output, which determines how much of the current state is retained to the next state; $c_t^{'}$ is the current input status value, which will be multiplied with i_t to determine the output value of the current state; o_t is the output gate output, which determines the output, and they have the following relationship:

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_t] + b_f) \quad (4)$$

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i) \quad (5)$$

$$c_t^{'} = \tanh(W_c \bullet [h_{t-1}, x_t] + b_c) \quad (6)$$

$$o_t = \sigma(W_o \bullet [h_{t-1}, x_t] + b_o) \quad (7)$$

In the above formula, W is the weight matrix, b is the offset, $\bullet$ denotes product operation and $\times$ denotes the pointwise multiplication operation.

$$c_t = c_{t-1} + i_t \times c_t^{'} \quad (8)$$

$$h_t = \tanh(c_t) \times o_t \quad (9)$$

Since the network layer of GAN is changed to LSTM network which required sequential input, so different from the original GAN network, the random noise inputs is represented by Z_i, $Z_i \in R^{T \times M}, i = 0,1,..,K$. Here K represents the number of samples, T represents the input/output sequence length, M represents the noise dimensions. The discriminator input is represented by X_i^{dis}, $X_i^{dis} \in R^{T \times N}, i = 0,1,..,K$, which either comes from the Generator or real sample. Here N represents the number of traffic detection sites. The IMN is fed with the real sample, denoted as $X_i^{IMN} \in R^{T \times N}, i = 0,1,..,K$. The output of IMN is the input of the well-trained Generator, the loss of the IMN is jointly determined by the Generator output and real samples.

Algorithm 1 LSTM-GAN for traffic flow scenario generate. Here v represents Generator and Discriminator training interval.

Loop

for number of training iterations **do**

for v steps **do**

Sample from noise distribution P_Z

$Z_i \in R^{T \times M}, i = 0,1,..,K$

Sample from real data distribution P_{data}

$X_i^{IMN} \in R^{T \times N}, i = 0,1,..,K$

Update the Discriminator by Adam optimizer:

$$\nabla \theta_d \frac{1}{K} \sum_{i=1}^{K} \left[\log D(X_i^{IMN}, \theta^D) + \log(1 - D(G(Z_i, \theta^G), \theta^D)) \right]$$

end for

Sample from noise distribution P_Z

$Z_i \in R^{T \times M}, i = 0,1,..,K$

Update the Generator by Adam optimizer:

$$\nabla \theta_g \frac{1}{K} \sum_{i=1}^{K} \log(1 - D(G(Z_i, \theta^G), \theta^D))$$

end for

end Loop

Algorithm 2 IMN for noise mapping

Loop

for number of training iterations **do**

Sample from real data distribution P_{data}

$X_i^{IMN} \in R^{T \times N}, i = 0,1,..,K$

$M_{out} = \text{IMN}(X_i^{IMN})$

$y\_ = G(M_{out})$

$E = mse(y\_, X_i^{IMN})$

Update the mapping by minimize E

end for

end Loop

III. EXPERIMENTS

A. DATA DESCRIPTION

The experimental data were collected from Performance Measurement System (PeMS) public database provided by California department of transportation. It is an open public transport data set, which contains traffic flow, road occupancy, vehicle speed, congestion and other types of traffic-related data. We use the traffic flow data recorded in District 4, Alameda County, California across the year 2019 for numerical experiment in this study. The data of the former nine months (Jan 2019 – Sep 2019) is used for training, and the remaining (Oct 2019 – Dec 2019) is used for verification. The typical daily data profile is shown in Fig 4. The sampling rate is 1 hour. The maximum and minimum values in Fig 4 correspond to the maximum and minimum traffic flow data recorded in multiple lanes, and the average value is used in this paper.

36 neighbored detection stations of the District 4, Alameda road network are considered in this study, as depicted in Fig 5(a). The red dots on the map represent the traffic data detection points of the road network. The data for each

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029230, IEEE Access

moment can be expressed by a 2D frame, as the time goes forwards, a traffic flow spatiotemporal sequence can be built, which is shown in Fig 5(b). In this case, the frame organized with the shape of 6*6. The target is to generate the spatiotemporal scenarios every four hours (e.g., 12:00 a.m. to 15:00 p.m., 15:00pm- 18:00 pm, etc.). Thus a 3D tensor $X \in R^{K \times T \times N}$ with shape

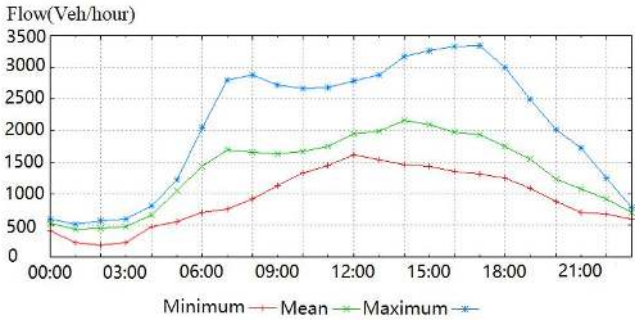
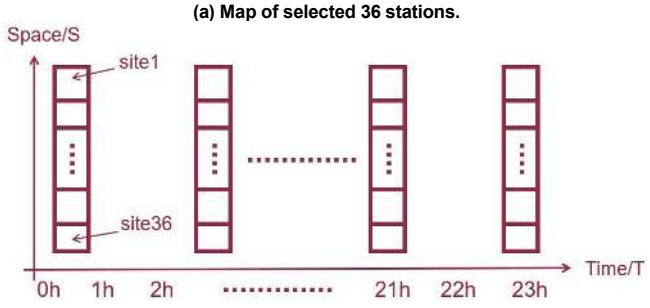


FIGURE 4. Typical traffic flow data for one day in one site

(number of samples, sequence length, number of sites) is formed as the real samples of the LSTM-GAN. All samples are normalized between -1 and 1 to shorten the model training time [19].

(a) Map of selected 36 stations.



(b) Tensor data schematic

FIGURE 5. Schematic diagram of data format conversion

B. MODEL SETUP

The comparative study is carried out against traditional GAN, which is applied to generate spatial traffic flow scenarios independently for each moment.

The collected samples are firstly preprocessed to satisfy the output requirements for LSTM-GAN as described in last section. Then we use Gaussian distribution to generate random noise data with the shape (4, 100), which will be used as the input of the Generator. The output of the Generator is a fake data sample with the same format as the real data sample. Because the data is characterized with temporal correlation, we use LSTM network layer to build Generator and Discriminator respectively. We use interval training between the Generator and the Discriminator. The network parameters of the Discriminator are fixed while training the Generator, and the network parameters of the Generator are fixed at the same time when training the Discriminator. After the training of the generation adversarial networks is completed, we will conduct a reflection mapping on the output sample of the Generator to find the optimal noise input using IMN. The IMN is composed of six layers of fully connected network layer and a reshape layer. The number of neurons from the first layer to the sixth layer of the full connection layer is *h1*, *h2*, *h3*, *h4*, *h5*, *h6* respectively, corresponding to *2000*, *1600*, *1200*, *800*, *600*, *400* in this paper. The activation function is RELU by default.

For traditional GAN, the traditional full connection layer is adopted respectively for the Generator and Discriminator. the difference lies in the setting of some specific parameters.

The specific network setups for both networks are shown in Table I and II. The value in each layer stands for the number of neurons. BatchNorm represents Batch Normalization. [-1, 1] is the normalized range. Dropout is introduced to avoid overfitting.

TABLE I
THE LSTM-GAN MODEL SETUP

	Generator	Discriminator
Input	(4, 100)	(4, 36)
BatchNorm	(-1, 1)	(-1, 1)
Layer1	LSTM,300	LSTM,300
Dropout	0.8	0.8
Activation	Tanh	Tanh
Layer2	LSTM,300	LSTM,300
Dropout	0.8	0.8
Activation	Tanh	Tanh
Layer3	MLP,36	MLP,100
Layer4	------	MLP,1

TABLE II
THE TRADITIONAL GAN MODEL SETUP

	Generator	Discriminator
Input	100	36
BatchNorm	(-1, 1)	------
Layer1	MLP,256	MLP,512
Activation	LeakyReLU(0.2)	LeakyReLU(0.2)
BatchNorm	(-1, 1)	------.
Layer2	MLP,512	MLP,256
Activation	LeakyReLU(0.2)	LeakyReLU(0.2)
BatchNorm	(-1, 1)	------
Layer3	MLP,1024	MLP,1
Activation	LeakyReLU(0.2)	Sigmoid
BatchNorm	(-1, 1)	------
Layer4	MLP,36	------
Activation	tanh	------

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3029230, IEEE Access

IV. RESULT

We show the heatmap of the generated and real spatiotemporal traffic scenarios in Fig 6. Each color block corresponds to a certain detection site. The pixel reflects the traffic flow of the corresponding site. If the pixel values of the two stations are similar, it means that the traffic volumes of these two stations are close. At each moment, the spatiotemporal sequences for the next four hours are simultaneously generated, which are labelled by t0, t1, t2, and t3. Fig 6(a) and Fig 6(b) show the one of generated spatiotemporal scenarios from 12:00 to 15:00 using the proposed conventional GAN and LSTM-GAN, respectively. One of the actual traffic measurements for this period is shown in Fig 6(c). We can see that, compared with the results of the data generated individually and separately, the spatial scenarios generated by LSTM-GAN are closer to the real samples for all time steps.

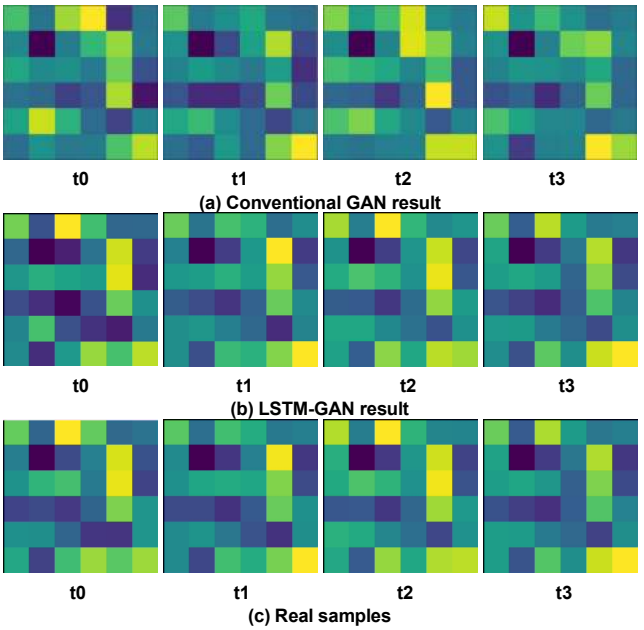


FIGURE 6. Comparison of generated scenarios for traditional GAN and LSTM-GAN

The other comparative experiment was carried out in regard of the spatial and temporal correlations exhibited by the generated traffic scenarios. Specifically, Pearson correlation coefficient is calculated for all sites and horizons. As a result, the spatial and temporal correlation matrices are formed as shown in Fig 7, where the degree of correlation is indicated by a value between -1 and +1. A value of +1 means total positive linear correlation, 0 is no linear correlation, and −1 is total negative linear correlation

The spatial correlation matrix with shape 36*36 for the traditional GAN, proposed LSTM-GAN and real samples are shown in Fig 7(a), (c) and (e), respectively. We can see from Fig 7(e) that the majority of the sites exhibits a positive correlation, i.e., if a site is detected to have large traffic volume within a time slot, the other neighbored sites are more likely to have a large traffic flow at this moment. The traditional GAN tends to overestimate this relationship between each site, giving a higher correlation than that of ground truth as seen in Fig. 7 (a). Instead, the scenarios from LSTM-GAN can well learn this spatial correlation, giving a closer correlation matrix, as shown in Fig. 7 (c). Fig 7(b), (d) and (f) depict the respective temporal correlation matrix with shape 4*4. Obviously, as the time moves forward, the temporal autocorrelation becomes lower, which can be observed in Fig 7(f). However, it still can be noticed that a relatively higher correlation exist among these adjacent moments, the lowest coefficient value can be as high as 0.79. The traditional GAN fails to learn the inherent temporal correlations, giving rather poor results, as shown in Fig 7(b). On the contrary, the LSTM-GAN renders more skillful results in terms of temporal dependencies, showing almost the same correlation matrix as that of real samples.

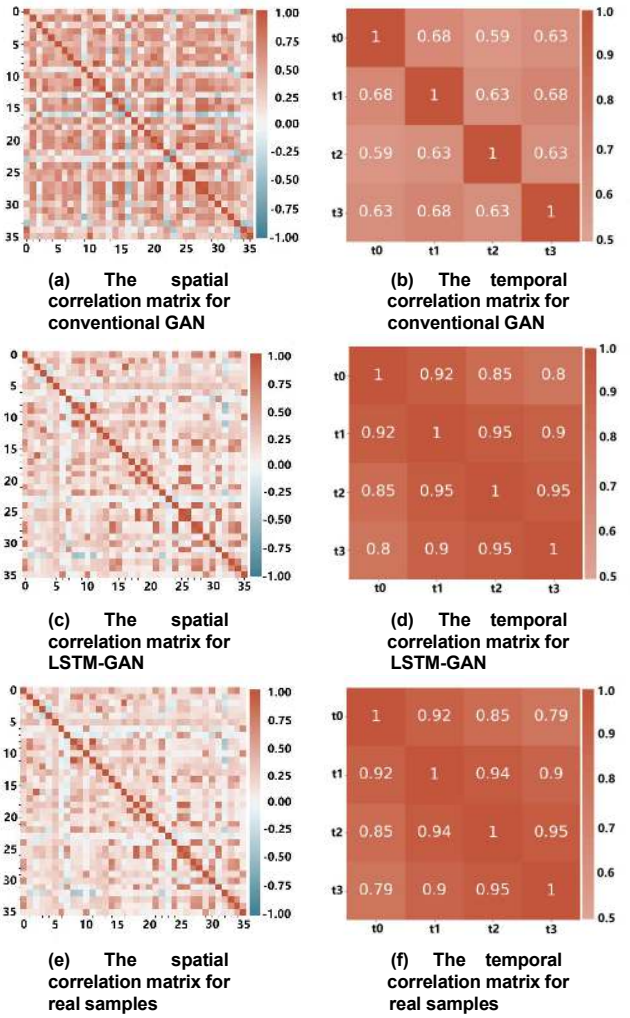


FIGURE 7. Spatiotemporal correlation matrices of comparison results

V. CONCLUSIONS

Traffic modeling is an important task in ITS research. A novel data-driven model based on LSTM-GAN is proposed

to produce the realistic traffic scenarios for multiple horizons and locations, which are highly beneficial to traffic infrastructure design and road planning. The LSTM-GAN leverages the merits of both LSTM and invert mapping networks. The former aims to capture the temporal dynamics involved in the traffic flow sequence, while the latter aims to output normal samples from a certain latent space. Case study is carried out using real traffic flow data provided by California department of transportation. Experimental results show that the proposed LSTM-GAN can well learn both temporal and spatial correlations. As compared with conventional GAN, which ignores the temporal dependencies of traffic flow, LSTM-GAN shows great superiority in model performance.

REFERENCES

[1] Igone García Pérez, Soloaga I A. Definition of traffic scennaries, application on a practise case of the criteria followed by the guide of good practises for elaboration of strategic traffic noise maps in urban routes[J]. Journal of the Acoustical Society of America, 2008, 123(5):3031.

[2] Rasslkin A, Vaimann T, Kallaste A, et al. Digital twin for propulsion drive of autonomous electric vehicle[C]// 2019 IEEE 60th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON). IEEE, 2020.

[3] Robertson D I. TRANSYT Method for area traffic control[J]. Traffic Engineering & Control, 1969, 10(6):181–182.

[4] Robertson D I, Bretherton R D. Optimizing networks of traffic signals in real time-the SCOOT method[J]. Vehicular Technology IEEE Transactions on, 1991, 40(1):11-15.

[5] Jat S, Tomar R S, Sharma M S P. Traffic Congestion and Accident Prevention Analysis for Connectivity in Vehicular Ad-hoc Network[C]// 2019 5th International Conference on Signal Processing, Computing and Control (ISPCC). IEEE, 2020.

[6] Owen L E, Zhang Y L, Rao L, et al. Street and traffic simulation: traffic flow simulation using CORSIM[C]// Simulation Conference Proceedings, 2000. Winter. 2000.

[7] Cera, Christopher D.， Soulchin, Robert M. Smith, Brian J, et al. Data-driven traffic views with continuous real-time rendering of traffic flow map, 2014.

[8] Chao Q, Deng Z, Ren J, et al. Realistic Data-Driven Traffic Flow Animation Using Texture Synthesis[J]. IEEE Transactions on Visualization & Computer Graphics, 2018:1-1.

[9] Stolfi D H, Alba E. Generating realistic urban traffic flows with evolutionary techniques[J]. Engineering Applications of Artificial Intelligence, 2018, 75(OCT.):36-47.

[10] V. Sapre, S. Kalambur, D. Sitaram and R. Bastian, "Synthetic Generation of Traffic Data for Urban Mobility," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, 2018, pp. 2151-2157, doi: 10.1109/ICACCI.2018.8554633.

[11] Aoyama H, Kato S, Tsugawa S. Indexing of traffic flow image sequences for traffic information services[C]// Intelligent Transportation Systems. IEEE, 2003.

[12] Brickwedde F, Abraham S, Mester R. Mono-SF: Multi-View Geometry Meets Single-View Depth for Monocular Scene Flow Estimation of Dynamic Traffic Scenes[J]. 2019.

[13] Peter König, Aigner S , Marco Körner. Enhancing Traffic Scene Predictions with Generative Adversarial Networks[J]. 2019.

[14] Radford A, Metz L, Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks[J]. Computer ence, 2015.

[15] Dan Li, Dacheng Chen, Jonathan Goh, See-Kiong Ng, Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series[J]. arXiv:1809.04758v3.

[16] Li H, Su L, Zhang S, et al. Spatial Temporal Convolution and LSTM Network for Predicting Traffic Flow[J]. ICCC, 2019.

[17] Chen Y , Wang Y , Kirschen D S , et al. Model-Free Renewable Scenario Generation Using Generative Adversarial Networks[J]. IEEE Transactions on Power Systems, 2017, 33(99):3265-3275.

[18] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. “Generative Adversarial Networks” [J]. advances in neural information processing systems, 2014, 3:2672-2680

[19] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[J]. 2015.